A Comparative Study of Automated Reviewer Assignment Methods

By

Joshua Peter Young

A Thesis Submitted to Saint Mary's University, Halifax, Nova Scotia,
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Applied Science

August 7, 2012, Halifax, Nova Scotia

Approved:	Dr. Stavros Konstantinidis Supervisor Department of Mathematics and Computing Science
Approved:	Dr. Vlado Keselj External Examiner Faculty of Computer Science Dalhousie University
Approved:	Dr. Pawan Lingras Supervisory Committee Member Department of Mathematics and Computing Science
Approved:	Dr. Evangelos Milios Supervisory Committee Member Faculty of Computer Science Dalhousie University
Approved:	Dr. Jeremy Lundholm Graduate Studies Representative
Date:	August 7, 2012

Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-90004-8

Our file Notre référence
ISBN: 978-0-494-90004-8

Abstract

A comparative study of automated reviewer assignment methods

by Joshua Peter Young

Abstract: The reviewer assignment problem is the problem of determining suitable reviewers for papers submitted to journals or conferences. Automated solutions to this problem have used standard information retrieval methods such as the vector space model and latent semantic indexing. In this work we introduce two new methods. One method assigns reviewers using compression approximated information distance. This method approximates the Kolmogorov complexity of papers using their size when compressed by a compression program, and then approximates the relatedness of the papers using an information distance equation. This method performs better than standard information retrieval methods. The second method assigns reviewers using Google desktop a more advanced information retrieval system. The method searches for key terms from a paper needing reviewers in a set of papers written by possible reviewers and uses the search results as votes for reviewers. This method is relatively simple and is very effective for assigning reviewers.

August 7, 2012.

Acknowledgements

I would like to express my sincere gratitude to all those who have helped make this thesis possible.

First and foremost I must thank my supervisor Dr. Stavros Konstantinidis, who has been a huge influence, introducing me to research as an undergraduate and encouraging me to continue on to graduate studies. His guidance and expertise was essential in the completion of this thesis.

I must also thank my supervisory committee members Dr. Pawan Lingras, Dr. Evangelos Milios and the external examiner, Dr. Vlado Keselj. Their insight, suggestions and corrections were invaluable in the process of shaping and completing this work.

I want to thank the Department of Mathematics and Computing Science, Faculty of Graduate Studies and Research, and Faculty of Science at Saint Mary's University for providing me the opportunity to study with them and financial support.

I would also like to thank NSERC for providing the financial support that made this possible.

Finally I would like to thank my family and specifically my parents whose support and encouragement have been an essential part of this accomplishment.

Table of contents

1. Introduction

1.1 Overview

The reviewer assignment problem is the problem of assigning qualified reviewers to review papers submitted to conferences or journals [37]. Often this task is done manually by editors or conference chairs; the process can be time consuming and is often done on tight deadlines. For this reason it is desirable to have automated systems to perform this task or at least make recommendations to help the editor or chair make final decisions. Automated reviewer assignment methods such as those in [1,14,18,24,39] use various information retrieval methods of computing document similarity as part of their algorithms.

In this research we examine two new methods of comparing document similarity for automated reviewer assignment. The first uses compression approximated information distance as an alternative to standard information retrieval methods. Information distance is a measure of the similarity or dissimilarity of the information contained in objects [29]. It can be approximated using compression programs and has been applied effectively in several practical applications such as data mining, clustering, computer learning, mapping of text data sets, program plagiarism detection, identification of websites, measuring relatedness of DNA sequences, automatic image annotation, content-based image retrieval, question answering, and music classification [26]. The second method replaces standard information retrieval methods with the more advanced Google desktop information retrieval system. Although Google desktop does not allow us to calculate document similarity directly we show how search results can be used to assign reviewers accurately.

1.2 Objectives

1. Determine the most effective information distance equation and compressor for determining document similarity in an automated reviewer assignment system. Past applications of compression approximated information distance have used various distances and compressors therefore we want to determine the best combination for reviewer assignment.
2. Compare the performance of the information distance method on different amounts of text. This is needed to determine the minimum amount of text needed to use the method and also to determine when having more text no longer improves results for reasons of efficiency.
3. Test possible improvements that could be made to the information distance method by combining it with elements of the vector space model to create a hybrid method.
4. Compare the information distance method against other standard information retrieval methods including methods that have been used for reviewer assignment in the past.
5. Determine the effectiveness of a reviewer assignment method that uses the more advanced Google Desktop API as a replacement for standard information retrieval methods.

1.3 Contributions

- The general algorithm for assigning reviewers given a similarity or distance measure.
- The comparison of the performance of the information distance method with other standard information retrieval methods used for reviewer assignment showing that information distance performs better.
- The comparison of combinations of information distance equations and compressors for the information distance method that shows what compressors and distance equations work best for reviewer assignment.
- The comparison of the information distance on different amounts of text that show the minimum amount of text required for the information distance method to be effective and the amount of text that gives the best results in terms of efficiency and accuracy.
- The hybrid information distance method that uses elements of the vector space model and its comparison to the information distance method that shows it can improve performance.
- The Google Desktop method of reviewer assignment and results showing that this method is very accurate and could be used effectively in practice.

1.4 Organization

The thesis is organized as follows; Chapter 2 contains a review of literature on both the reviewer assignment problem and the compression approximation of information distance.

Chapter 3 introduces the general reviewer assignment algorithm that was used for testing in the later Chapters. This algorithm is general in the sense that the distance measure from any standard information retrieval method can be used to assign reviewers.

Chapter 4 explains the information distance method used in this research and includes the results of a comparison of several information distance equations and several compression programs.

Chapter 5 gives a comparison of the information distance method results on datasets that have text documents of different sizes to determine how the results change based on the amount of text used.

Chapter 6 introduces some possible improvements for the information distance method using elements of the vector space model. This improved method is tested and compared to the standard information distance method.

Chapter 7 compares the information distance method against other standard information retrieval methods used for reviewer assignment. The other methods tested are the vector space model, latent semantic indexing, and a second order co-occurrence method.

Chapter 8 introduces a reviewer assignment method that replaces standard information retrieval methods with the Google desktop API. The results of this method are compared to the information distance method as well as the other methods from Chapter 7.

2. Literature Review

2.1 Reviewer assignment

The reviewer assignment problem is the problem of assigning qualified reviewers to review papers submitted to conferences or journals [37]. This problem involves several issues such as how to represent papers and reviewers, how to match papers and reviewers, and how to avoid problems such as one reviewer being assigned no papers or too many papers. This research focuses on the second issue computing the match between papers and reviewers since any automated system must have some method of comparing how similar the subject of a paper is to a reviewer's area of expertise. Past research in this area has used data mining and information retrieval methods for this purpose. One of the key papers in this area is [14] where latent semantic indexing was used in an automated reviewer assignment method. This method was further refined in [39] where the authors used the vector space model as part of a reviewer assignment system. Other papers on automated reviewer assignment that have used the vector space model as part of their system are [1,18,24].

When using these methods one of the key questions is what representation of the paper's subject and the reviewer's expertise should be used. For example the subject of the paper could be represented by a list of keywords, an abstract, or the full text version of the paper. Similarly the reviewer's expertise could be represented by a list of keywords, abstracts of their publications or research interests, or the full text of their publications. In [14] the paper was compared to abstracts written by the reviewers describing their interests. In [39] the paper was compared to papers written by the reviewer. The representations included just keywords, keywords title and abstract, full text, and full text with a weighting based on regions of the text. In [1] the authors used the paper and the reviewer's internet home page and papers linked from the home page.

2.2 Information distance

2.2.1 Kolmogorov complexity and information distance

Information distance is a measure of the similarity or dissimilarity of the information contained in objects [29]. This similarity is calculated based on the Kolmogorov complexities and relative Kolmogorov complexities of the object's binary string representation.

The Kolmogorov complexity $K(x)$ of an object x is the size of the shortest program that outputs x [29]. The conditional Kolmogorov complexity $K(x|y)$ is the length of the shortest program that outputs x when given input y. $K(x,y)$ is the length of the shortest program that outputs the string xy and a description of how to tell them apart. $K(xy)$ is the length of the shortest program that outputs the string xy.

With respect to a universal Turing machine U, the cost of conversion between two objects x and y is $E(x,y)$ [26],

$$E(x,y) = \min\{|p|: U(x,p) = y. U(y,p) = x\},$$

Where $U(x,p) = y$ means that the program p on input x gives output y.

There have been several equations proposed to measure information distance. The first two examples were given in [2]. They are the sum distance

$$Dsum(x,y) = K(x|y) + K(y|x)$$

and the max distance

$$Dmax(x,y) = \max\{K(x|y), K(y|x)\}$$

An upper bound of $E(x,y)$ is the summation information distance $Dsum(x,y)$. $E(x,y)$ is equal to the maximum information distance $Dmax(x,y)$ up to an additive $O(log(max\{K(x|y), K(y|x)\}))$ term.

$Dmax$ and $Dsum$ both satisfy the properties of a metric (symmetry, positivity, and the triangle inequality) up to an additive constant or logarithmic term but more importantly these information distances are *universal*. This means that they minorize all other computable distances. That is they account for, or measure, every effective resemblance between the two objects.

The problem with these definitions of information distance is that if objects x and y are not roughly of the same size they will be found to be dissimilar based on their size rather than the information they contain. To address this problem a new distance was proposed in [9], the shared information distance.

$$dshare(x,y) = 1 - \frac{K(x) - K(x|y)}{K(xy)}$$

where $K(x) - K(x|y)$ is defined as the mutual information between x and y and $K(xy)$ is the Kolmogorov complexity of the concatenation of x and y [29]. The shared information distance is also equivalent to the sum distance normalized.

$$\frac{K(x|y) + K(y|x)}{K(xy)} = \frac{Dsum(x,y)}{K(xy)}$$

The max distance normalized is known as the normalized information distance and is defined in [27] as

$$NID(x,y) = \frac{max\ \{K(x|y), K(y|x)\}}{max\ \{K(x), K(y)\}}$$

Both $dshare$ and NID satisfy the properties of a metric, but whether they are universal distances is not yet solved.

Another issue with these definitions of information distance is although at first it seems that any information distance should satisfy the properties of a metric, it is actually the case that what we may think of as similar concepts does not always follow the triangle inequality. To address this issue another definition of information distance was introduced in [26], the minimum information distance ($Dmin$). $Dmin$ is based on $E_{min}(x,y)$ the cost of conversion between x and y with respect to a universal Turing machine U if the information that is not relevant to the conversion is removed.

$$E_{min}(x,y) = \min\{|p|: U(x,p,r) = y, U(y,p,q) = x, |p| + |q| + |r| \leq E(x,y)\}$$

Where $U(x,p,r) = y$ means that the program p on input x and r (the information in x not relevant to y) gives output y.

In terms of Kolmogorov complexity $E_{min}(x,y)$ is equal to

$$Dmin(x,y) = \min\{K(x|y), K(y|x)\}$$

omitting $O(log(|x| + |y|))$ factors. $Dmin$ is a universal distance, and it is symmetric and positive but it does not satisfy the triangle inequality. There is also a normalized version of the minimum information distance $dmin(x,y)$

$$dmin(x,y) = \frac{\min\{K(x|y), K(y|x)\}}{\min\ \{\mathrm{K(x)}, \mathrm{K(y)}\}}$$

Again this distance is symmetric and positive but does not satisfy the triangle inequality and like the other normalized information distances its universality is unsolved.

2.2.2 Approximating information distance

Unfortunately information distance equations like those above cannot be directly calculated in practice because the Kolmogorov complexity of an object is not computable [29]. In practice it is necessary to use an approximation of Kolmogorov complexity. The most common method used is to approximate $K(x)$ with $C(x)$, where $C(x)$ is the compressed size of x using the compression program C.

When using compression to approximate Kolmogorov complexity the normalized information distance becomes the normalized compression distance [27]

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

Where $C(x)$ is the compressed size of x, $C(y)$ is the compressed size of y, and $C(xy)$ is the compressed size of the concatenation of x and y. This is because $K(x|y)$ is equal to $K(x,y)$ - $K(y)$ plus an added constant, and $K(x,y)$ is equal to $K(xy)$ plus the encoding of the separator between x and y. Hence $K(x|y)$ is approximately equal to $K(xy)$ - $K(y)$ and the NID can be written as

$$NID(x,y) = \frac{\max\{K(xy) - K(y), K(yx) - K(x)\}}{\max\{K(x), K(y)\}} = \frac{K(xy) - \min\{K(x), K(y)\}}{\max\{K(x), K(y)\}}$$

As pointed out in [6] there are some issues with the metric properties of the normalized compression distance that can lead to poor results. These problems arise from the choice of compression program C and the size of the files being compressed. In general the normalized compression distance satisfies the properties of a metric when C is a *normal* compressor [29]. A compressor is *normal* if it satisfies the following properties for the size of the files being compressed:

1. $C(xx) = C(x)$
2. $C(\lambda) = 0$
3. $C(xy) \geq C(x)$
4. $C(xy) = C(yx)$
5. $C(xy) + C(z) \leq C(xz) + C(yz)$

up to additive $O(log\ n)$ terms.

Other approximations of information distance using the compression approximation are the scaled normalized compression distance [35]:

$$NCDs(x,y) = NCD(x,y) - \frac{NCD(x,x) + NCD(y,y)}{2}$$

And the Compression based dissimilarity metric [25]

$$CDM(x,y) = \frac{C(xy)}{C(x) + C(y)}$$

These distances are based on practical experimentation rather than strictly on information distance theory.

2.2.3 Applications of information distance

Compression approximations of information distance have been shown to be effective in several practical applications of data mining and information retrieval [26]. In these applications different information distance equations and compressors have been used.

One of the important practical applications of information distance is the parameter-free data mining method introduced in [25]. The information based distance measure used in this method is the compression based dissimilarity metric (CDM).

Information distance was used for data mining again in [11]. The normalized compression distance (NCD) with compressor bzip2 was used as the distance measure for a hierarchical clustering algorithm working on a data set consisting of music files. This method was shown to be effective in distinguishing between the genres and composers of music files.

The clustering method of [11] was also applied to a much more varied group of data sets in [12]. These data sets included literature, astronomy, genomics and languages and were all shown to be clustered successfully. Various compressors were used including gzip, bzip2 and ppmz.

Information distance has also been used to support mapping and visualization of large text data sets [35]. Here the authors used the scaled normalized compression distance ($NCDs$), with the compressor bzip2. The method of mapping text documents using this distance was compared to the standard cosine metric over the vector representations of the text documents and performed well. Other benefits of the method are the following: it does not require the processing of the vector representation of the text files, and documents in different languages can all be handled by the same method since it does not require the elimination of stop words or the stemming of words.

In [15] the authors used information distance for content based image retrieval. The distance used was an approximation of Normalized information distance (NID) using the compressor gzip.

A software integrity diagnosis system using information distance was proposed in [8]. This system is designed to determine if computer programs (such as university assignments) have been plagiarized. In the method presented the authors use the shared information distance ($dshare$) and a compressor made specifically for this application called TokenCompress.

Compression based approximation of information distance has also been used in biology. In [9] the authors use compression to approximate $dshare$ between DNA sequences for the purpose of creating DNA trees. In this case a compressor, GenCompress, was specifically designed to compress the DNA sequences.

Information distance has also been used for classification of webpage information. In [32] web pages were classified by authorship, topic and domain using gzip to approximate NID.

3. General reviewer assignment algorithm

3.1 Algorithm

For all methods of comparison a general reviewer assignment algorithm was used for testing. This meant that each method was compared on a level basis since the algorithm requires only a distance measure to determine similarity of two paper's subject matter. The representations of papers and reviewers used are the full text of the paper to be reviewed and a set of full text papers published by the reviewer (unless otherwise noted). With the large number of author's papers available online in electronic format, this approach provides a better representation of reviewers' expertise than asking potential reviewers to provide keywords that describe their area of expertise.

```
For each paper $p_i$
    For each reviewer $r_j$
        For each reviewer paper $r_{j,k}$
            Calculate $d_k = D(p_i, r_{j,k})$
        Sort all $d_k$
        Calculate $a_j$ the average of the $t$ smallest $d_k$
    Sort all $a_j$ in ascending order
    Output a list of the $r_j$ that correspond to the sorted $a_j$
```

Figure 3.1: General reviewer assignment algorithm.

The algorithm (Figure 3.1) starts with a set of papers to be reviewed $P = \{ p_1 , p_2 , \dots \}$ and a set of reviewers $R = \{ r_1 , r_2 , \dots \}$ where each reviewer r_j is itself a set of publications by that reviewer $r_j = \{ r_{j,1} , r_{j,2} , \dots \}$. For each paper p_i the algorithm finds the reviewer (with n publications) that minimizes the equation

$$\frac{\sum_{k=1}^{n} D(r_{j,k}, p_i)}{n}$$

where D is any distance measure between two text documents. If the comparison method gives a similarity measure rather than a distance, D can be replaced with a similarity measure S and we would find the reviewer that maximizes the equation.

This algorithm outputs the reviewer with the body of work that is most similar to the subject of the paper. However there is a problem if a reviewer has written papers in multiple subject areas. When ranking the reviewer for one subject (the subject of the paper) the reviewer's score would be penalized by the algorithm for all the papers they have written on other subjects. To fix this problem only the most relevant papers by the reviewer should be considered. Therefore for each reviewer the algorithm calculates the distance from each of their papers to the paper to be reviewed and then sorts the results. Based on these sorted results the algorithm uses only the t most relevant papers to calculate the average and create a score for the reviewer. Here t represents the minimum number of papers on the subject we require a reviewer to have published to be selected. This means that any papers the reviewer has written other than the t most relevant will not count against him or her. A low value for t means we only require a reviewer to have a small amount of experience in the subject, while a high value means we require the reviewer to have published many papers on the subject to be considered qualified. In our tests we used $t = 5$ (see next section).

3.2 Testing methodology

Tests were performed on two datasets; the first was a set of papers by members of the Dalhousie University Faculty of Computer Science. Before the dataset was used authors in the dataset with less than 6 papers were removed so that each author had at least 5 papers to represent their expertise and one more to be removed and used as a paper to be reviewed. This resulted in a dataset of 593 papers by 23 authors.

The second dataset was a collection of papers taken from 11 scientific journals from different subject areas. 50 papers were taken from each journal to create a dataset of 550 papers. For this dataset the collection of papers from each journal was considered to be defining the expertise of a fictional reviewer from that subject area.

Since the papers in the first dataset were all from one subject area it should be more difficult for methods to correctly differentiate between the specific subjects, whereas the subjects in the second dataset are more clearly differentiated and it should be easier for a method to make correct assignments.

Each test was set up as follows; one paper by each reviewer was removed from the dataset to create a set of papers that needed to be reviewed, the general algorithm was then run (with the value of t set at 5) to assign the top five reviewers for each of the papers to be reviewed. This process was repeated five times, each time with a different set of papers to be reviewed selected and the results over the five tests were averaged.

The results of the tests were evaluated based on the observation that although in real life an author cannot review their own paper, theoretically they should be one of the best qualified reviewers. Therefore if the paper was assigned its author as a reviewer the assignment was considered correct.

The results from each test are presented using $A(x, p)$, the *accuracy* of the method A in assigning reviewers to paper p when considering the top x ranked reviewers. If the author of p appears in the top x reviewers then $A(x, p) = 1$; if the author is not in the top x reviewers $A(x, p) = 0$. The graphs presented in the following chapters shows each method's average $A(x, p_i)$ for each of the p_i's averaged over the five tests on the y-axis. The x values 1 to 5 are shown on the x-axis. Tables containing the full results for all chapters can be found in the appendix.

4. Comparison of information distances and compressors

To determine the best combination of information distance equation and compression program several equations and compressors were tested including those that have been used in previous applications seen in Chapter 2 and some that have not yet been used in practical applications.

The Information distances tested were the Normalized Compression Distance,

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

the Scaled Normalized compression Distance,

$$NCDs(x,y) = NCD(x,y) - \frac{NCD(x,x) + NCD(y,y)}{2}$$

the Compression Based Dissimilarity Metric,

$$CDM(x,y) = \frac{C(xy)}{C(x) + C(y)}$$

the compression approximation of the Shared Information Distance,

$$Cdshare(x,y) = \frac{2C(xy) - (C(x) + C(y))}{C(xy)}$$

and the compression approximation of the Normalized Minimum Information Distance.

$$Cdmin(x,y) = \frac{C(xy) - \max\{C(x), C(y)\}}{\min\{C(x), C(y)\}}$$

The last two distances are the compression approximations of the information distances $dshare$ and $dmin$ based on the approximation of $K(x|y)$ as $K(xy)$ - $K(y)$ used in the approximation of NID as NCD seen in Chapter 2.

The compression programs tested are shown below along with the compression algorithms they use:

gzip (version 1.2.4) - Lempel-Ziv 77 and Huffman coding

bzip2 (version 1.0.5) - Burrows–Wheeler transform, move-to-front transform, and Huffman coding

7zip (version 4.65) - Burrows–Wheeler transform, Move-to-front transform, Huffman coding and Lempel–Ziv–Markov chain

ppmz (version 9.1) - prediction by partial matching

ctw (version 0.1) - context tree weighting

zpaq (version 1.00) - context mixing algorithm

These distances and compressors were used to assign reviewers using the testing methodology presented in the previous chapter. Full text representations of the papers to be reviewed and the reviewers' papers were used.

4.1 Results

4.1.1 Dataset 1

The average performance of each compressor over all five tests and all five information distances on dataset1 is shown in Figure 4.1. zpaq performed the best at selecting the correct reviewer with its first recommendation. On the other hand ppmz did get more correct reviewers in the top 4 and 5 recommendations. The worst performing compressor was ctw.

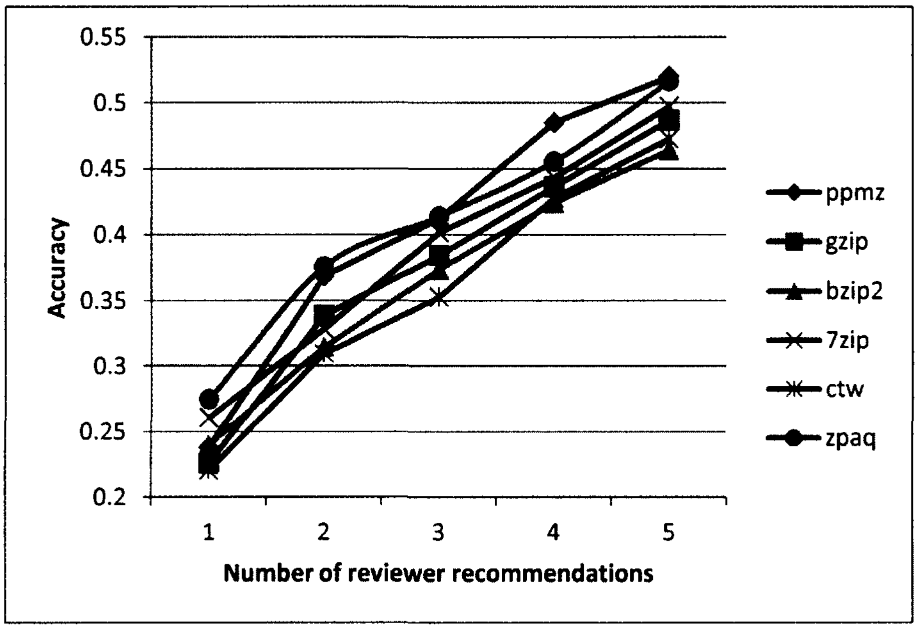


Figure 4.1: Average accuracy based on compressor for dataset1.

In Figure 4.2 the average performance of each of the distance equations over all five tests and six compressors on dataset1 is shown. The best performing distance equations were CDM and $Cdshare$. While $Cdmin$ performed much worse than the other equations.

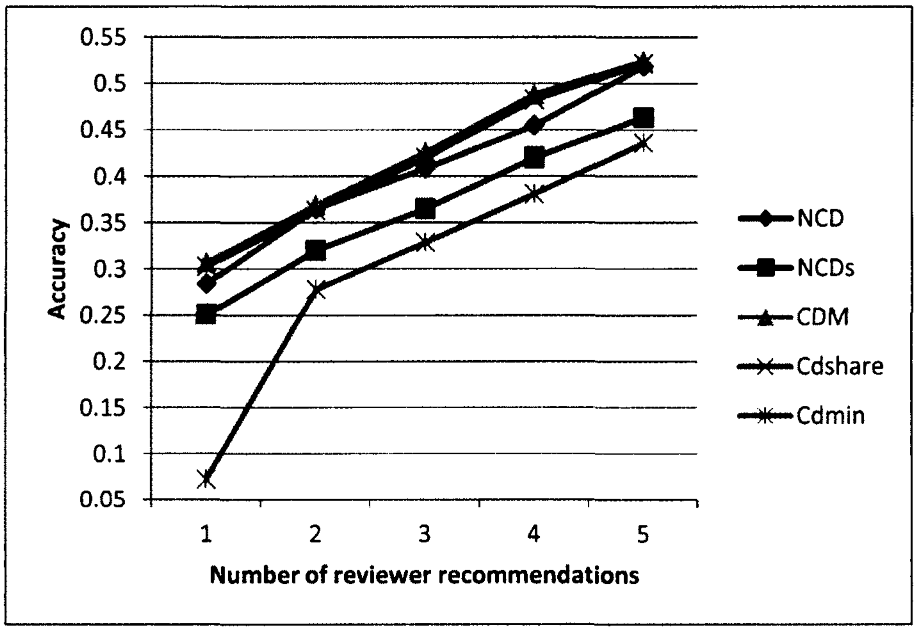


Figure 4.2: Average accuracy based on information distance for dataset1.

The best performing combination of distance equation and compressor over all five tests on dataset1 was the combination of *CDM* and bzip2. This can be seen in Figure 4.3 compared to the average of the performance of all the distance and compressor combinations (ID). The worst performing combination, *Cdmin* with the compressor bzip2, is also shown. This method performed significantly below average. Based on these result it seems that *Cdmin* performs poorly when the subject matter of the papers are very similar as is the case in dataset1.

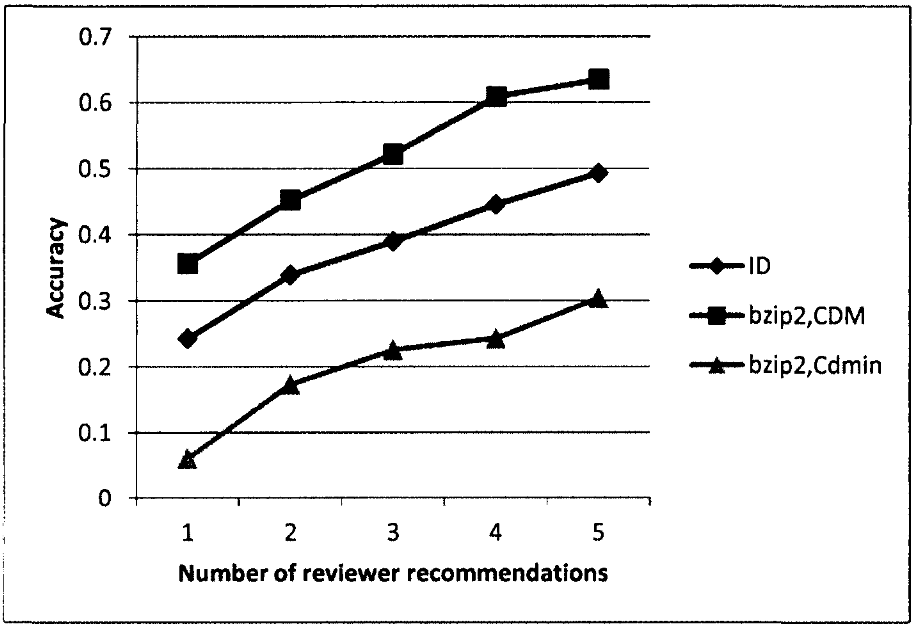


Figure 4.3: Best, worst, and average accuracy of combinations for dataset1.

4.1.2 Dataset 2

In Figure 4.4 the average performance of each compressor over all five tests and all five information distances on dataset2 are shown. 7zip performed the best when considering only the top recommendation. On the other hand bzip2 did better at getting the correct journal in the top 2 or more recommendations. Comparing these results to those from dataset1 (Figure 4.1) we see that the performance of individual compressors is not consistent. For example bzip2 was one of the worst performing compressors on dataset1 but one of the best performing on dataset2. This inconsistency is not unexpected as the varying size and content of papers in different datasets will affect how each compression algorithm will perform and therefore how well they will approximate Kolmogorov complexity.

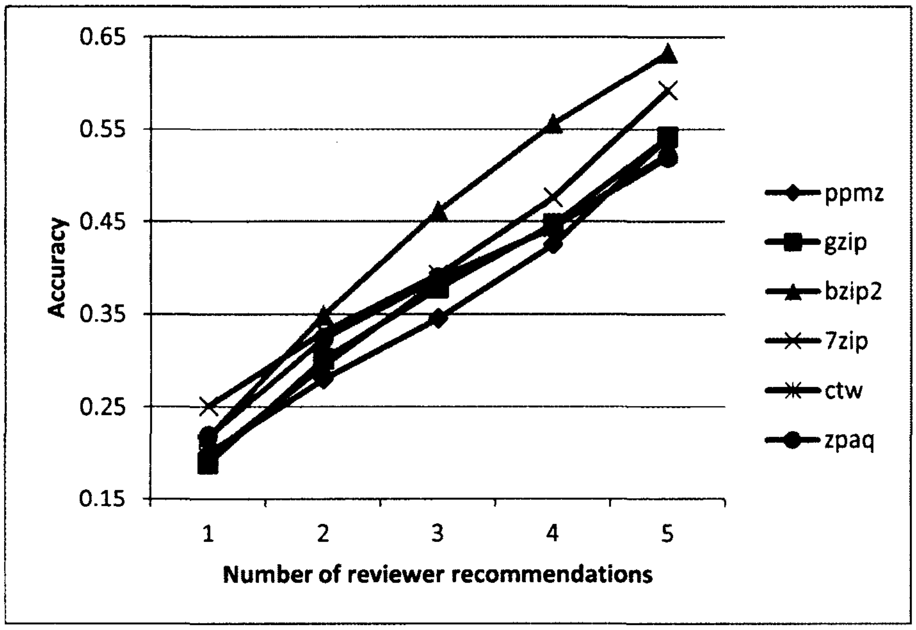


Figure 4.4: Average accuracy based on compressor for dataset2.

In Figure 4.5 the average performance of each of the distance equations over all five tests and six compressors on dataset2 is shown. The best performing distance equation was *Cdmin*. However *CDM* and *Cdshare* were close, and actually performed better when considering the top 5 recommendations. Comparing these results to those from dataset1 (Figure 4.2) we see that *Cdmin* performs much better on dataset2 were the subjects were more clearly differentiated.

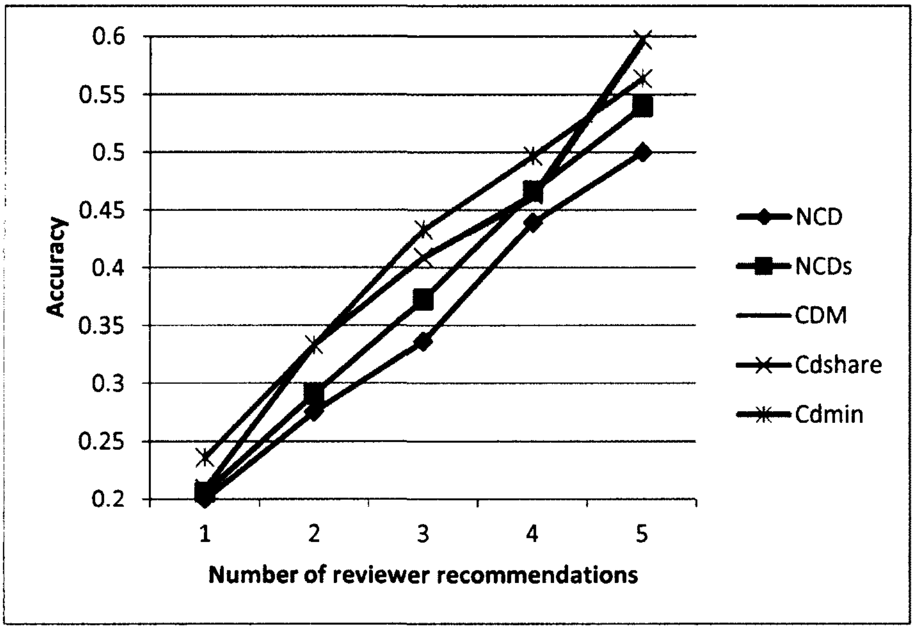


Figure 4.5: Average accuracy based on information distance for dataset2.

The best performing combination of equation and distance was not as clear cut in the five tests on the dataset2. The combination of *Cdmin* and bzip2 did the best at selecting the correct journal as its first recommendation, but after that it was outperformed by the combination of *NCDs* with the compressor ctw; both are shown along with the average performance (ID) in Figure 4.6. The worst performing combination on dataset 2 was the combination of *NCD* with the compressor gzip also seen in Figure 4.6. Compared with the results from dataset1 (Figure 4.3) we see that the worst combination for dataset1, *Cdmin* and bzip2 is actually one of the best combinations for datset2. This makes sense as we have already seen that bzip2 performed better on dataset2 as did *Cdmin*. This seems to indicate that although *Cdmin* performs very poorly when the subjects of papers are close, it is effective when subjects are more clearly differentiated.

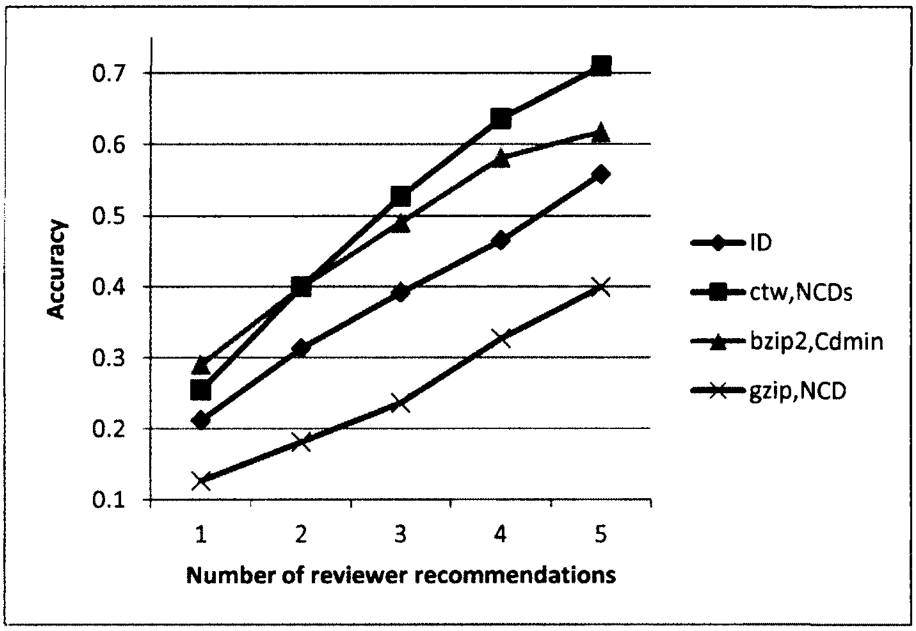


Figure 4.6: Best, worst, and average accuracy of combinations for dataset2.

4.1.3 Overall

Over the tests on both datasets the best compressors for getting a correct assignment as the top recommendation were 7zip and zpaq. However if you consider the top 3 or more recommendations then bzip2 was the best performing compressor. The results for all compressors are shown in Figure 4.7.

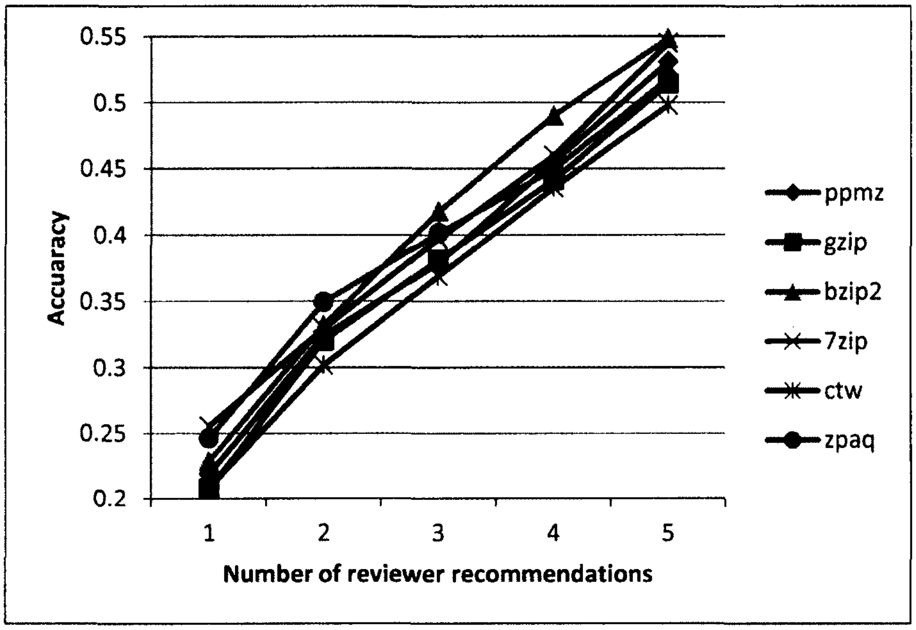


Figure 4.7: Average accuracy based on compressor for both datasets.

Over the tests on both datasets the best performing distance equations were clearly CDM and $Cdshare$ as seen in Figure 4.8.

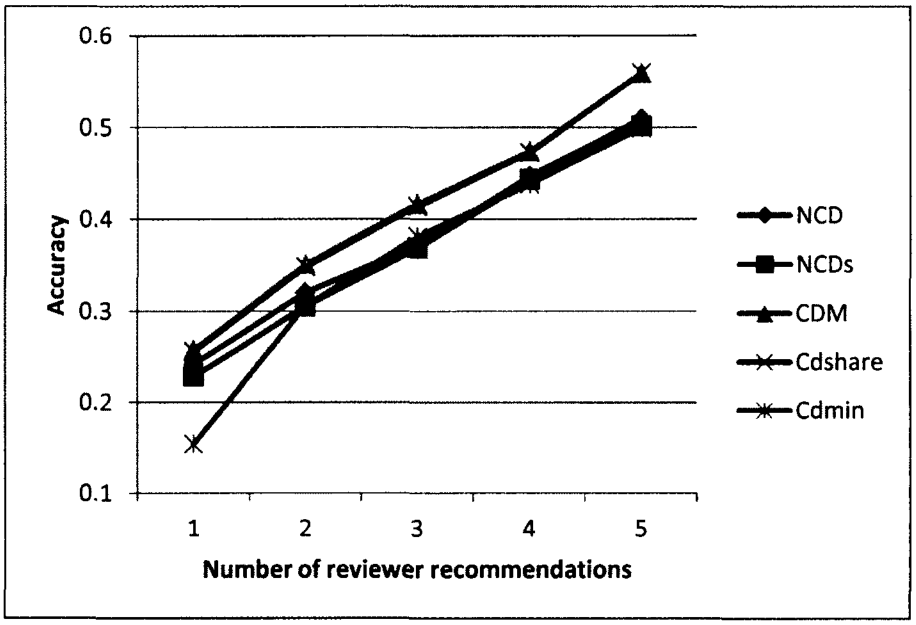


Figure 4.8: Average accuracy based on distance for both datasets.

The best performing combinations of compressors with distances (seen in Figure 4.9 with the average performance (ID)) were the combinations of CDM and $Cdshare$ with the compressor bzip2. The worst combination was NCD with the compressor gzip also seen in Figure 4.9.

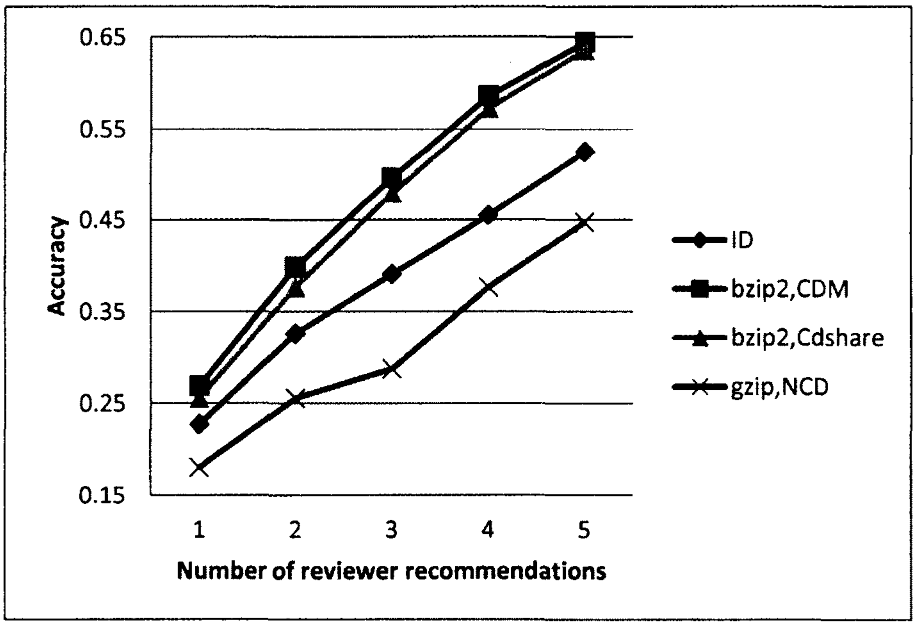


Figure 4.9: Best, worst, and average accuracy of combinations for both datasets.

4.2 Efficiency

These results consider only the accuracy of the method and not the efficiency. Using a compression approximation of an information distance equation can be expensive in terms of computing time. The amount of time needed to run the algorithm depends on the choice of compressor, the number and size of the files being compressed, and the hardware used but in general it may take hours to get the results. This may seem like it is a major problem with using these methods but for certain problems instantaneous results are not as important as accurate results. The reviewer assignment problem is an example of one of these problems since the results don't need to be immediate as long as they are accurate. However since the time needed depends on the size of the files being compressed using smaller text files can improve the efficiency of the method. The tests in the following Chapter attempt to determine the file size required to get accurate and efficient results.

5. Size comparison for information distance methods

An important question when using the information distance method is the minimum amount of text that is needed before the method is effective. As well as what amount of text gives optimal results since, as mentioned in the previous Chapter, if the amount of text processed is reduced then the efficiency of the method can be improved.

To answer these questions all the tests on dataset 1 from Chapter 4 were repeated using differing amounts of text from the papers. Tests were performed where the method only compressed the first 250, 500, 750, and 1000 words of the papers to see how the results compared to the full text results.

5.1 Results

5.1.1 250 words

Using only the first 250 words of the papers the average performance for all of the information distance combinations (ID) was in the range of 25% to just under 50% and the best performing combinations of compressors and distances were zpaq with $Cdmin$, and ctw with $NCDs$ (Figure 5.1).

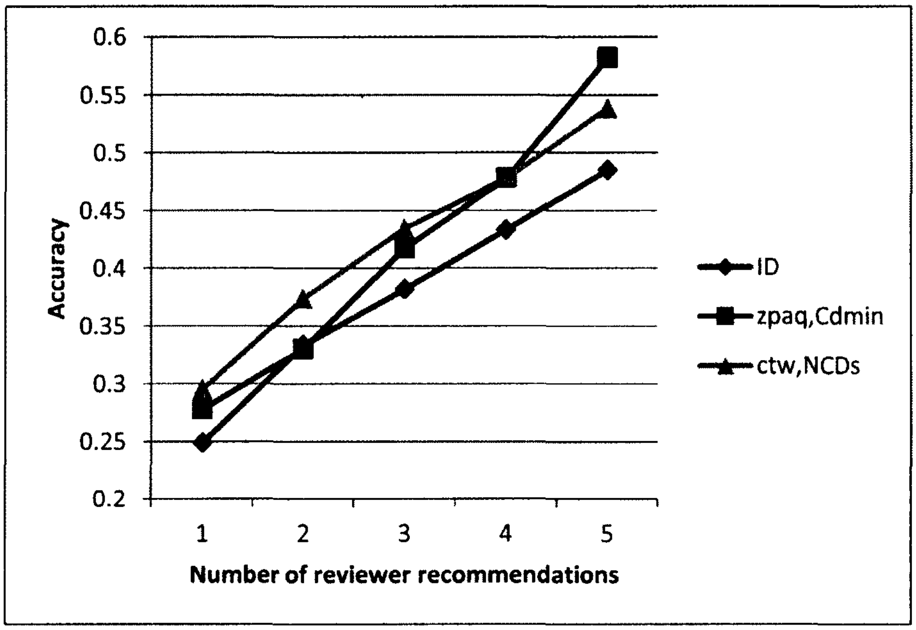


Figure 5.1: Best, worst, and average accuracy of combinations for 250 words.

5.1.2 500 words

Using the first 500 words of the papers, as would be expected, improved the results and the average performance of all combinations (ID) was in the range of just under 30% to a little more than 50%. The best performing combinations of compressors and distances were ctw with NCD and $NCDs$ as seen in Figure 5.2.

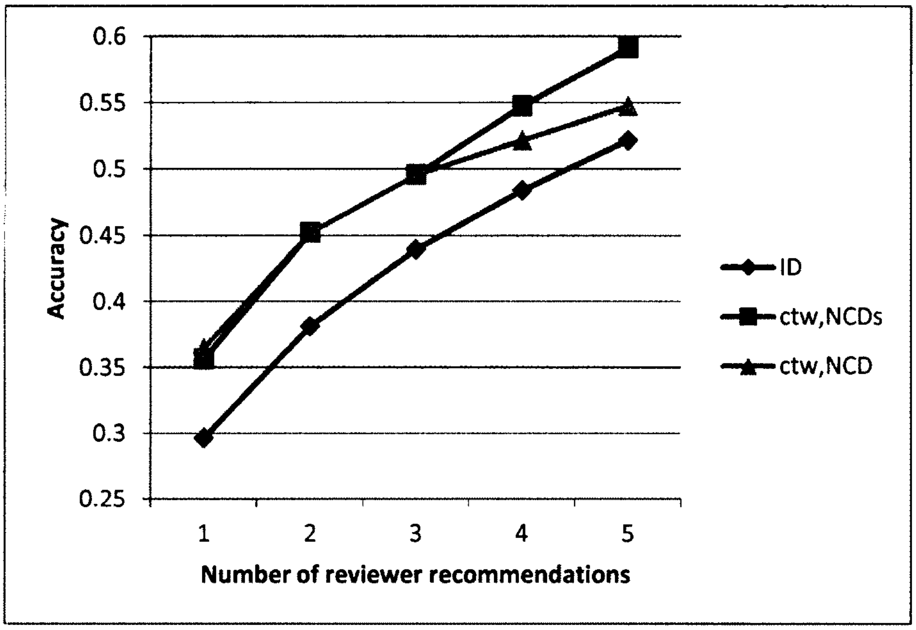


Figure 5.2: Best, worst, and average accuracy of combinations for 500 words.

5.1.3 750 words

Increasing the amount of words used to the first 750 words did not significantly improve the average performance of all combinations (ID). The performance was still in the range of just under 30% to just over 50%. The best performing combination was not clear, but the combinations of compressors 7zip, bzip2, and zpaq with *Cdshare* were the best for different numbers of results considered (See Figure 5.3).

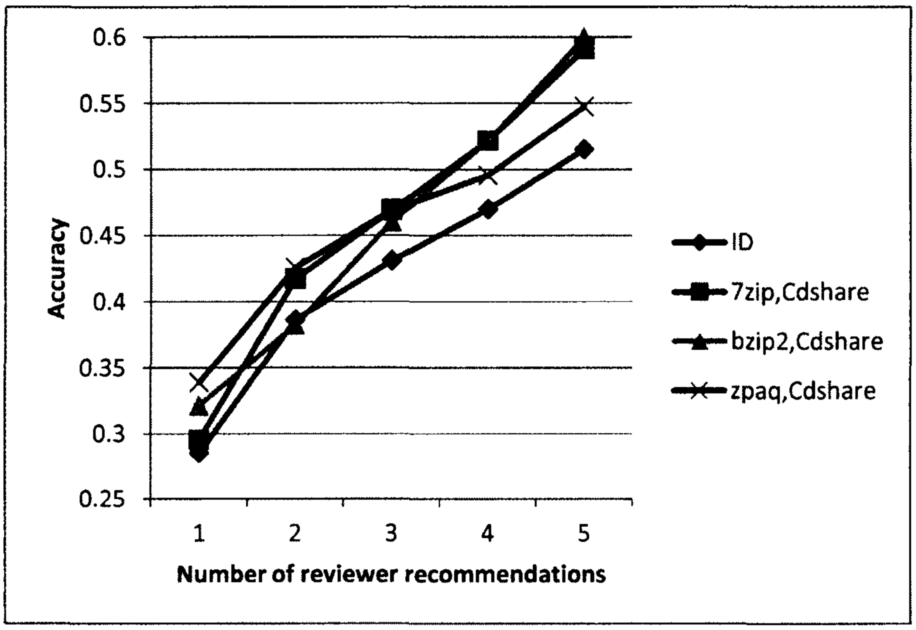


Figure 5.3: Best, worst, and average accuracy of combinations for 750 words.

5.1.4 1000 words

Further increasing the amount of words used to 1000 again only slightly increased the average performance (ID) which was from just under 30% to a little over 50%. The best performing combination was not clear, but the combination of gzip with *NCDs* did best when considering the top recommendation and the top 5 recommendations, although ctw with *Cdmin* and 7zip with *NCD* were better at points in between as seen in Figure 5.4.

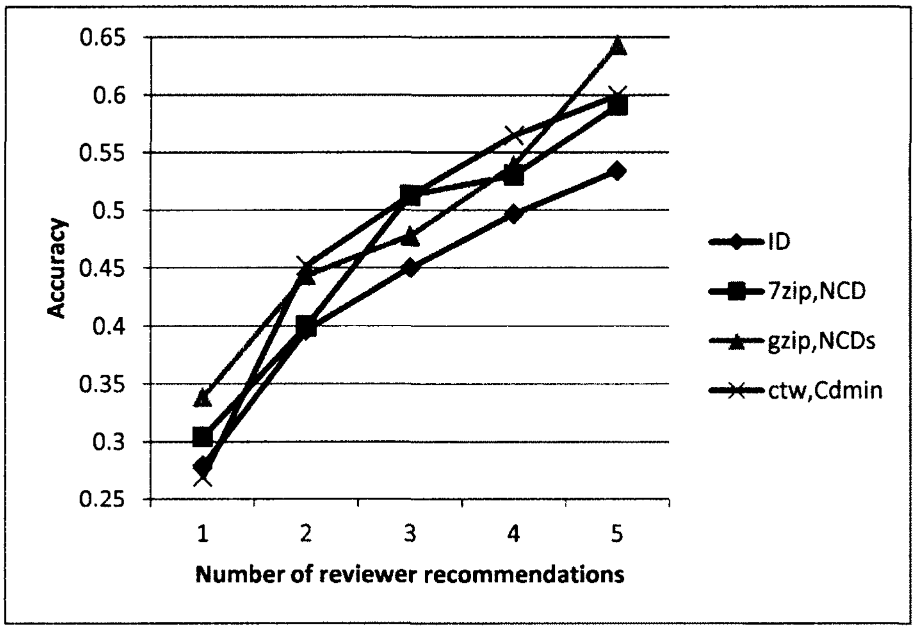


Figure 5.4: Best, worst, and average accuracy of combinations for 1000 words.

5.1.5 Comparison

In Figure 5.5 the side by side comparison of the average performance of the information distances for the different amounts of words is shown. The two amounts of text that gave the worse results were 250 words and full text. This is probably because 250 words is not enough to get the meaning of the paper, while the full text adds confusion and is not compressed as well because of its size. Using 500, 750 or 1000 words gave similar performance, therefore using 500 words would be the best choice since this would be more efficient than using 750 or 1000 words.

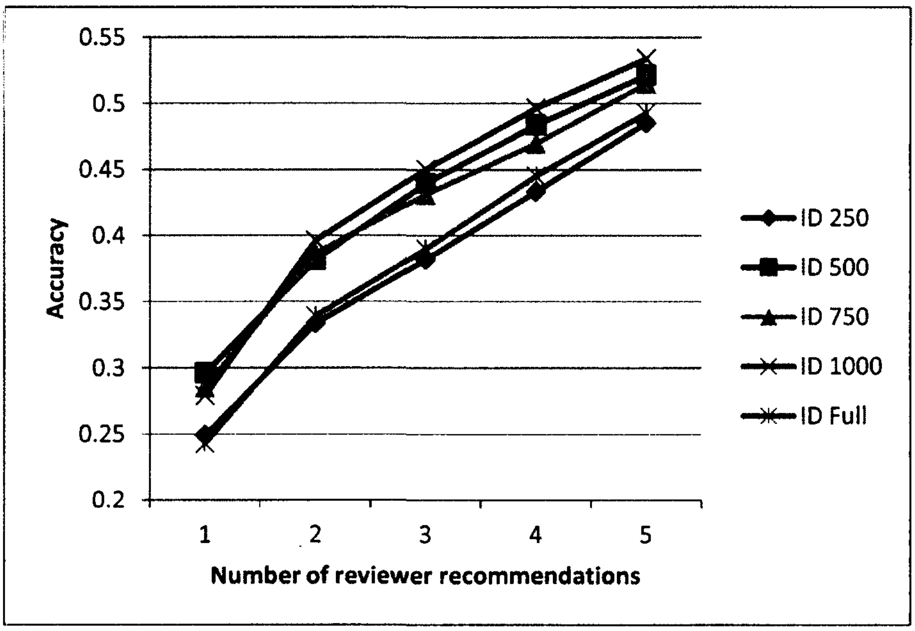


Figure 5.5: Comparison of average accuracy based on number of words.

It should also be noted that when using smaller amounts of words some compressors may be more effective than they are on larger files or vice versa. For example the files may be small enough that they fit into the window size for that compressor and, therefore, the compression approximation will be closer to the actual Kolmogorov complexity. Similarly some distance equations may be more or less effective based on the size of the files used. Based on Figure 5.2, the results for using the first 500 words (what appears to be the best choice), we see that the best results were with the compressor ctw and NCD or $NCDs$.

6. Hybrid information distance method

6.1 Algorithm

Using compression to calculate information distance does not always give good approximations since the compressed size of an object is itself an approximation of the Kolmogorov complexity. This approximation may or may not be close to the actual value in general. However when working with text documents there are established methods for finding similarity, if we could combine some of these methods with the information distance methods it might improve the approximations and thus the results. The hybrid information distance method described in this Chapter is an attempt to do just that using aspects of the vector space model to enhance the compression approximation of Kolmogorov complexity for text documents.

To improve the compression approximation of Kolmogorov complexity compression programs need to be able to identify the patterns in the text. This can be aided by adding preprocessing steps performed on the text documents before they are compressed. The first possible step is to remove stop words from the document, which leaves less confusion for the compressor to deal with and does not affect the meaning of the text. The second possible step is to stem the words in the document. This would allow the compressor to match words that have the same stem that would not have been matched as exact strings. It is important to note that unlike the vector space model these steps are performed in a way that preserves the structure of the document. That is to say that any words that are not removed from the document still appear in the same relative location in the document so that the document should still be able to be read and understood by a human. This ensures the information content of the document stays the same but some unnecessary content is removed to hopefully lead to better compression and hence a better approximation of Kolmogorov complexity.

To test this hybrid method the algorithm and methodology from Chapter 3 were used again with the addition of the preprocessing steps described above.

6.2 Results

6.2.1 Dataset 1

The comparison between the average performance of the information distance methods with the average performance of information distance hybrid methods on dataset 1 is shown in Figure 6.1. The methods perform roughly the same on this dataset with the hybrid method better at getting the right reviewer in the first three recommendations and the normal method performing better when more than three results are considered.

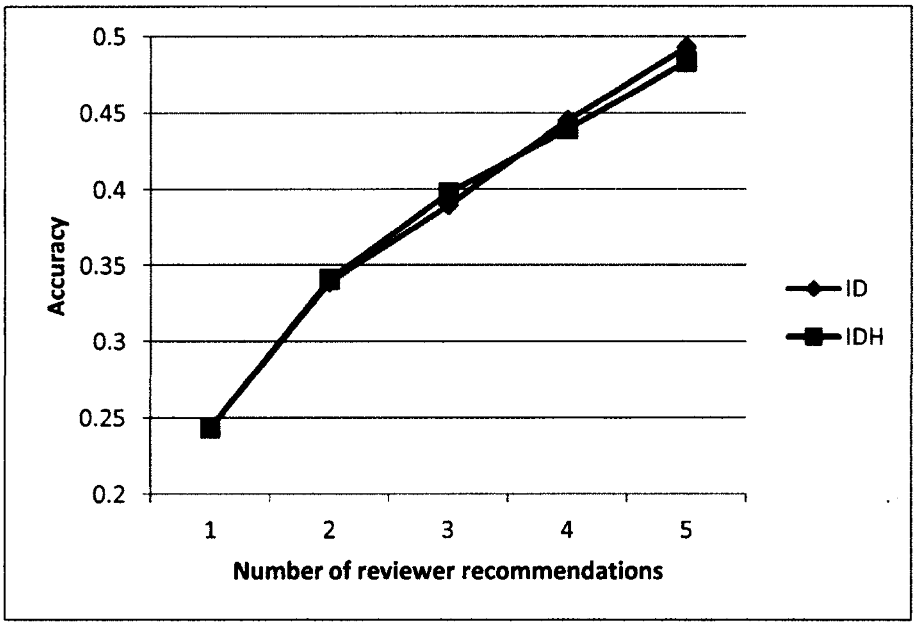


Figure 6.1: Comparison of average accuracy for dataset1.

6.2.2 Dataset 2

For dataset 2 the comparison between the hybrid and non-hybrid information distance methods can be seen in Figure 6.2. The hybrid methods clearly outperformed the non-hybrid methods on this dataset. This result is interesting as the second dataset had clearly separated subject matter compared to the first dataset which had papers from one specific subject area. This seems to indicate that the two methods perform about the same when the subjects of the papers are close together but the hybrid method performs better when the subjects are more clearly differentiated.

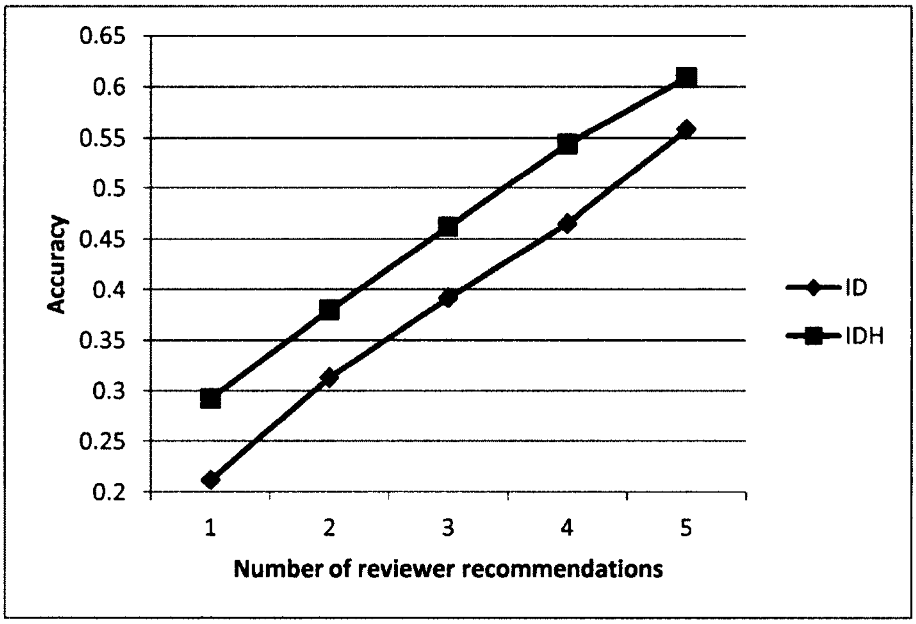


Figure 6.2: Comparison of average accuracy for dataset2.

6.2.3 Overall

The combined results over both datasets can be seen in Figure 6.3. Although the results for the hybrid and non-hybrid methods are close, the hybrid methods do perform better. This suggests that a more advanced hybrid method may be worth developing to increase the performance of information distance methods. It should also be noted that the way the compression program works has an effect on whether these preprocessing steps will improve the compression results. For some compressors the preprocessing steps will improve for other this may not be the case. A compression program specifically designed to compress text could help improve the performance of the information distance methods further, the preprocessing steps from this chapter could be used as the first steps for such a compression algorithm.

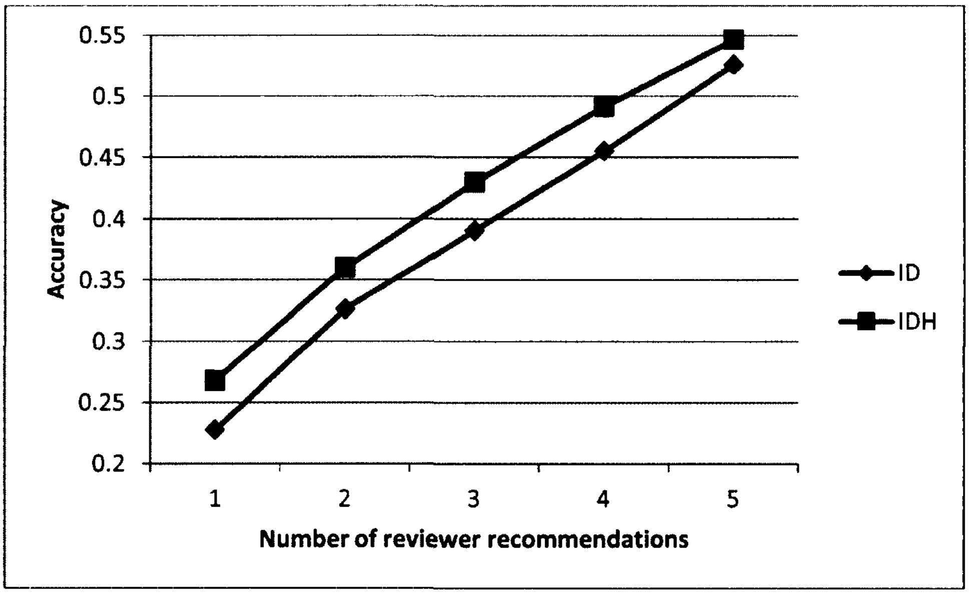


Figure 6.3: Comparison of average accuracy for both datasets.

7. Comparison with other methods of reviewer assignment

To determine how effective the information distance methods were for reviewer assignment the results were compared to several other methods, including methods that have been used for reviewer assignment in the past. The testing method used was the same as the one used in Chapter 4 where the distance calculation in the general algorithm of Chapter 3 was replaced with the distance calculation for each of the methods described below.

7.1 Vector space model

The first method the information distance methods were compared to was the vector space model (VSM). This was an important comparison since the vector space model is the standard method of comparing document similarity and has been used for reviewer assignment in past research [1,18,24,39].

Following the standard vector space model, all the documents in the datasets had stop words removed, words stemmed using Porter's stemming algorithm, and Term frequency – inverse document frequency weights calculated to create a term document matrix. More specifically each entry i, j in the term document matrix was calculated as follows,

$$(tf - idf)_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \times log \frac{|D|}{|\{d: t_i \in d\}|},$$

where D is the dataset, and $n_{i,j}$ is the number of times the term $t_i$occurs in document d_j.

The cosine metric was used to calculate the similarity of the word vectors to determine document similarity.

$$cos\theta = \frac{d_1 \cdot d_2}{||d_1||\ ||d_2||}$$

7.2 Latent semantic indexing

A slightly more advanced method based on the vector space model is the latent semantic indexing method (LSI). LSI uses the same term document matrix as the vector space model but involves performing a Singular Value Decomposition (SVD) on the matrix. The purpose of this is to determine relationships between terms and concepts rather than just matching exact terms. LSI has been used as a method for reviewer assignment before in [14]. The method works as follows:

Let A be the term document matrix for the dataset. We perform an SVD on the matrix A resulting in the matrices T, S and D such that $A=TSD^T$.

In this decomposition S is the singular value matrix. The T matrix is the term-concept vector matrix which represents how related each term in the dataset is to each concept. These vectors can be compared with the cosine metric to determine how related two terms are. D is the document-concept vector matrix which represents the extent of which each concept appears in each document. These vectors can be compared with the cosine metric to determine how related two documents are.

However before any comparisons are done, the matrices are reduced by preserving only the *k* largest singular values. This is so comparisons focus only on the most important concepts and it also reduces noise that could confuse comparisons. For our tests we used a *k* value of 50.

7.3 Second order co-occurrence for document comparison

Another method that attempts to compare terms based on meaning rather than basic matching is the second order co-occurrence method (SOC) [20]. This method determines the semantic similarity of two words based on a dataset D. The method proceeds as follows: The *pointwise mutual information* between each word and the terms it occurs with is calculated based on m the total number of terms in the dataset, the term frequency f^t (the number of times a term occurs in the data set) and the bigram frequency f^b (the number of times the term occurs together with the word in a context window). For our tests we used a context window size of 10 words.

$$f^{pmi}(t_i, W) = \log\frac{f^b(t_i, W) \times m}{f^t(t_i)f^t(W)}$$

For each word the co-occurring terms are sorted by their PMI values which are then used to calculate the ß-PMI summation function.

$$f^b(W_1) = \sum_{i=1}^{\beta_1}(f^{pmi}(t_i, W_2))^\gamma$$

This sums the PMI values of all terms semantically close to the second word that are also semantically close to the first word. β determines how many common words to consider. The value of β is defined as,

$$\beta_i = (\log(f^t(W_i)))^2\frac{(\log_2(n))}{\delta}$$

where n is the number of unique terms in the dataset and δ depends on the size of the dataset (for all our tests we used a δ value of 6.5). The choice of γ determines the emphasis put on high PMI values. For all our tests we used a γ value of 3.

Finally the semantic PMI similarity function is calculated.

$$Sim(W_1, W_2) = \frac{f^{\beta}(W_1)}{\beta_1} + \frac{f^{\beta}(W_2)}{\beta_2}$$

To compare two documents we extended the method presented in [21] for determining the similarity of sentence meaning to full documents.

```
Find $W_1 = \{ w_{1,1}, w_{1,2}, \dots, w_{1,k}\}$ the $k$ keywords with highest tf-idf values in document $D_1$
Find $W_2 = \{ w_{2,1}, w_{2,2}, \dots, w_{2,k}\}$ the $k$ keywords with highest tf-idf values in document $D_2$
$\delta = 0$
$\rho = 0$
For each keyword $w_{1,i}$ in $W_1$
    If $w_{1,i} \varepsilon W_2$
        Remove $w_{1,i}$from $W_1$ and $W_2$
        $\delta = \delta + 1$
Create $k - \delta$ by $k - \delta$ matrix $M$ as follows
For each remaining keyword $w_{1,i}$ in $W_1$
    For each remaining keyword $w_{2,j}$ in $W_2$
        $M_{i,j} = Norm( Sim(w_{1,i}, w_{2,j}) )$
While M is not empty
    Find the max value $M_{i,j}$ in $M$
    $\rho = \rho + M_{i,j}$
    Remove row $i$ and column $j$ from $M$
Return $(\delta + \rho)/k$
```

Figure 7.1: Second order co-occurrence document similarity algorithm.

The algorithm (Figure 7.1) proceeds as follows: *k* keywords are chosen from each document (using a keyword extraction method such as term frequency – inverse document frequency). Any exact matches are counted (δ) and removed from the keywords. Then a matrix of PMI similarity values (normalized to ensure they are in the range [0,1]) for the remaining terms is created. The maximum value in the matrix is added to a summation (ρ) and the matrix is reduced by removing the corresponding terms.

This process is repeated until the matrix is empty. The similarity of the two documents is then calculated using the equation

$$S(d_1, d_2) = \frac{(\delta + \rho)}{k}$$

7.4 Results

7.4.1 Dataset 1

The performance of these methods on dataset 1 compared to the average information distance performance and the best information distance performance can be seen in Figure 7.2. For this dataset the average of the information distance methods clearly outperforms the other methods. In fact on this dataset the LSI method actually performs worse than the less advanced VSM method. This result was confusing at first however it is mentioned in [14] that on some datasets the performance of LSI is significantly worse than usual, so this is the most likely reason for this result. The SOC method also does not perform well on this dataset. It does outperform VSM at some points but we would expect it to do much better since it is a more advanced method. This poor performance is probably related to the problem the LSI method encountered with this dataset. It should be noted that if we consider the best performing information distance method it outperforms the other methods by an even larger margin.

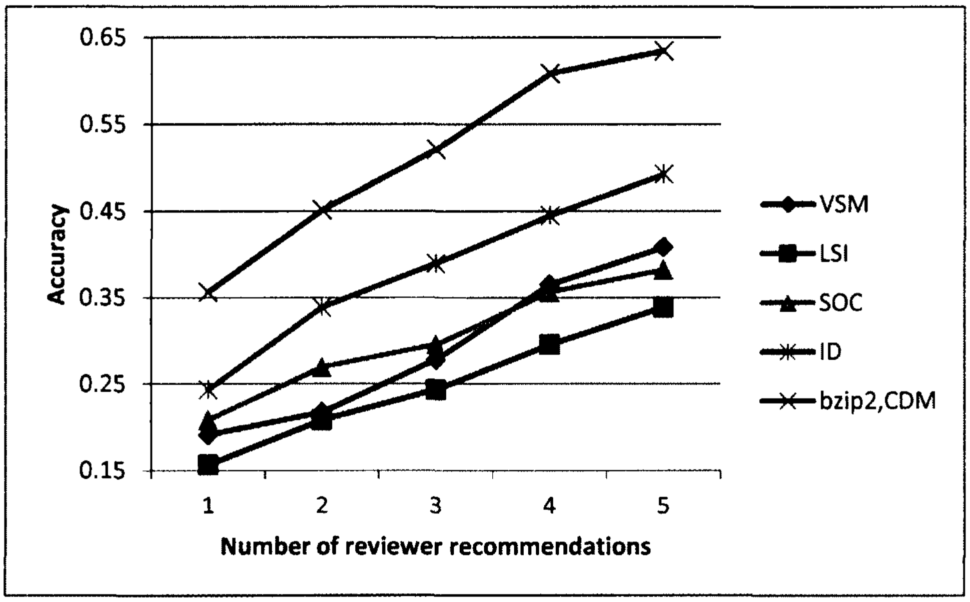


Figure 7.2: Comparison of accuracy for dataset1.

7.4.2 Dataset 2

The performance of these methods on dataset2 (Figure 7.3) is more like what we would expect. The VSM method does the worst as it is the least advanced. The SOC method outperforms VSM but it is outperformed by the average performance of the information distance methods when the top 1 to 4 results are considered. When the top 5 results are considered SOC and the average performance of the information distance methods perform about the same. The LSI method also performed much better on this dataset but it is outperformed by the average of the information distance methods when the top 3 or fewer results are considered. However the LSI method does outperform the average of the information distance methods when the top 4 or 5 results are considered. As with the first dataset if we consider the best performing information distance combination it significantly outperforms the other methods.

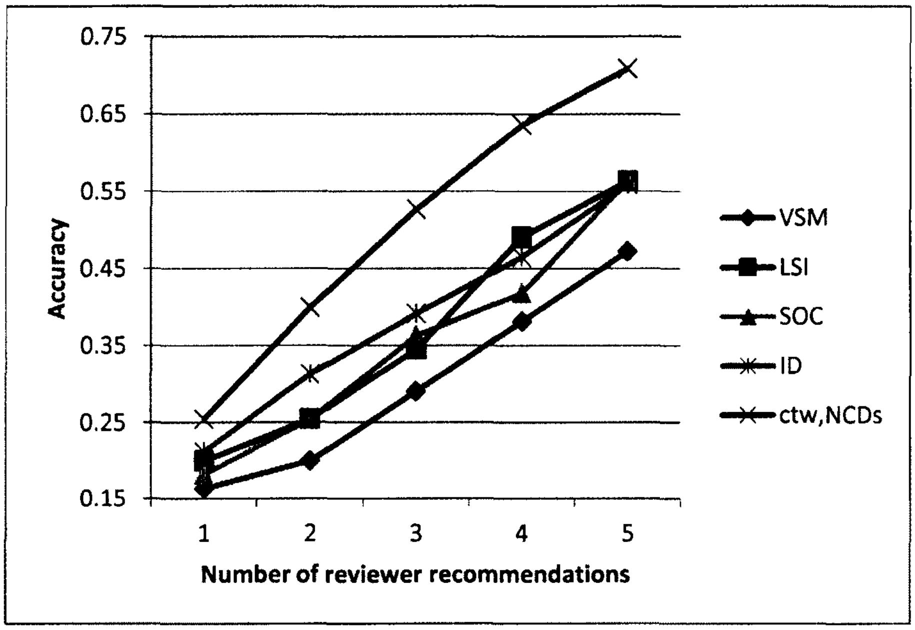


Figure 7.3: Comparison of accuracy for dataset2.

7.4.3 Overall

Considering the average performance of the methods over both datasets (Figure 7.4) it can be seen that the average information distance outperforms the other methods. This is due to the very poor performance of the LSI and SOC methods on the first dataset. This poor performance, although possible, is not typical of these methods. Therefore the results of the tests on dataset 2 are a better representation of the comparison with the information distance methods. However even though the average information distance performance does not always outperform the other methods in those results, the best information distance combination does in fact outperform the other methods.

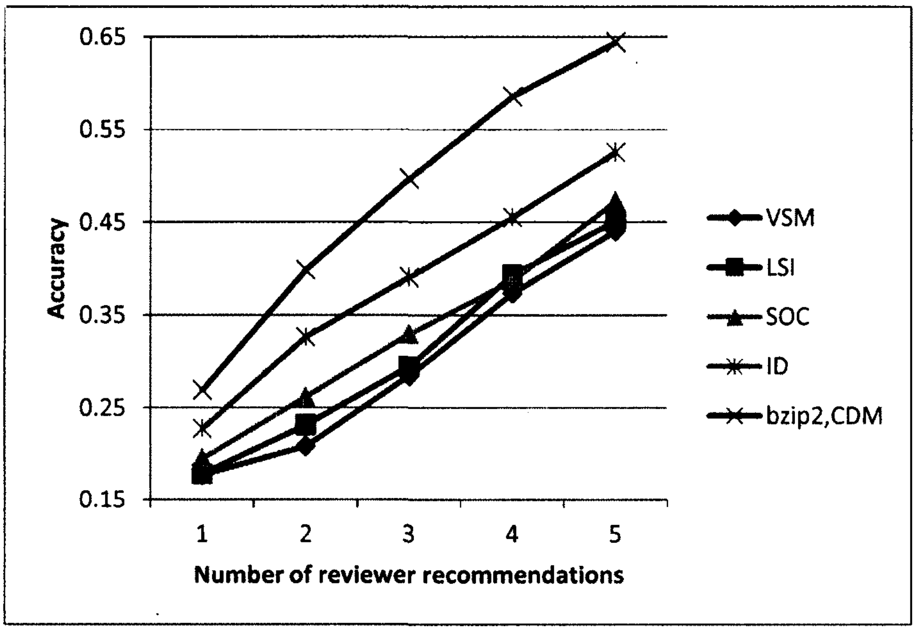


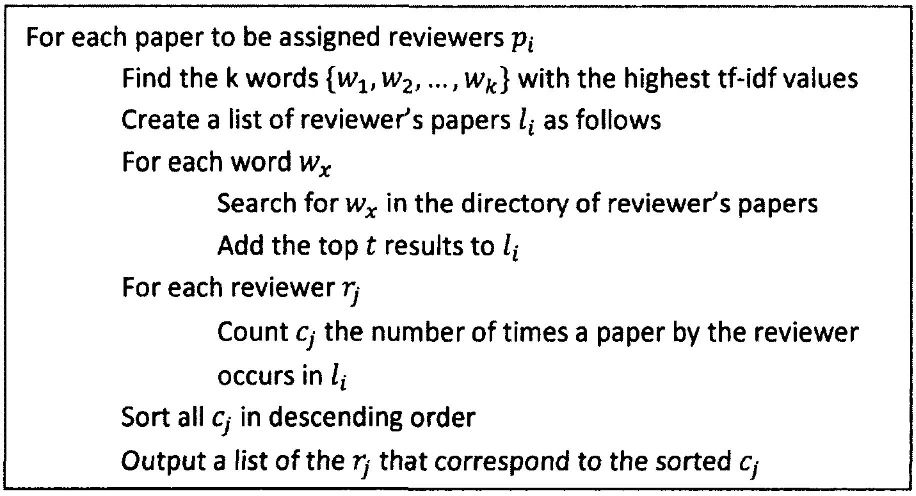
Figure 7.4: Comparison of accuracy for both datasets.

It should also be mentioned that the performance of the methods such as LSI and SOC could benefit from learning relations between words from a large external dataset and applying these relations to the smaller dataset of reviewer papers being used. This could increase the performance of these methods. However, one advantage of the information distance methods is that no external dataset is needed. For this reason the methods were compared on an even basis using only the dataset of reviewers' papers to learn from.

8. Google desktop method

8.1 Algorithm

The methods examined in the previous chapters have all used a standard information retrieval method to calculate the similarity between a paper to be reviewed and papers written by possible reviewers. In this chapter we replace these standard methods with Google, a much more advanced information retrieval system. Unfortunately it is not possible to use the actual similarity measure that Google uses, and therefore it is not possible to use the general algorithm from Chapter 3 to assign reviewers using Google. However Google does make available the API for their Google desktop search tool (available at *code.google.com/apis/desktop/)* which allows one to search files on a computer for a query string. Furthermore using the "under" keyword this search can be limited to a specific directory. Using this tool we were able to create a reviewer assignment algorithm that uses search results as a voting system to determine reviewers.



```
For each paper to be assigned reviewers $p_i$
    Find the k words $\{w_1, w_2, \dots, w_k\}$ with the highest tf-idf values
    Create a list of reviewer's papers $l_i$ as follows
    For each word $w_x$
        Search for $w_x$ in the directory of reviewer's papers
        Add the top $t$ results to $l_i$
    For each reviewer $r_j$
        Count $c_j$ the number of times a paper by the reviewer
        occurs in $l_i$
    Sort all $c_j$ in descending order
    Output a list of the $r_j$ that correspond to the sorted $c_j$
```

Figure 8.1: Algorithm for the Google Desktop method.

The algorithm (Figure 8.1) proceeds as follows: Using a keyword extraction method (we used term frequency-inverse document frequency weights), extract a number k of keywords from the paper which needs reviewers (for our tests we used $k = 10$). For each of these keywords perform a Google desktop search limited (using the "under" keyword) to a directory containing the papers representing possible reviewers. Concatenate the top t results from each search (or all the results if there are less than t) to create a single list of results for the paper needing a reviewer (for our test we used $t = 10$). Each paper by a possible reviewer in this list is then considered a vote for that person to review the paper; the reviewers are then ranked by the number of votes they have.

8.2 Testing methodology

For this method it was not possible to maintain the same testing process used for the other methods. However using the algorithm described above, a list of the best reviewers can be created, just as a list of the best reviewers was created using the algorithm from Chapter 3. Based on this ranking of reviewers the same system of evaluation described in Chapter 3 was used to determine the accuracy of the method.

8.3 Results

8.3.1 Dataset 1

The performance of the Google desktop method compared to the other reviewer assignment methods on dataset 1 can be seen in Figure 8.2. The Google desktop method outperformed all the other methods by a significant amount, achieving close to 80% correct assignment when considering the top 5 results.

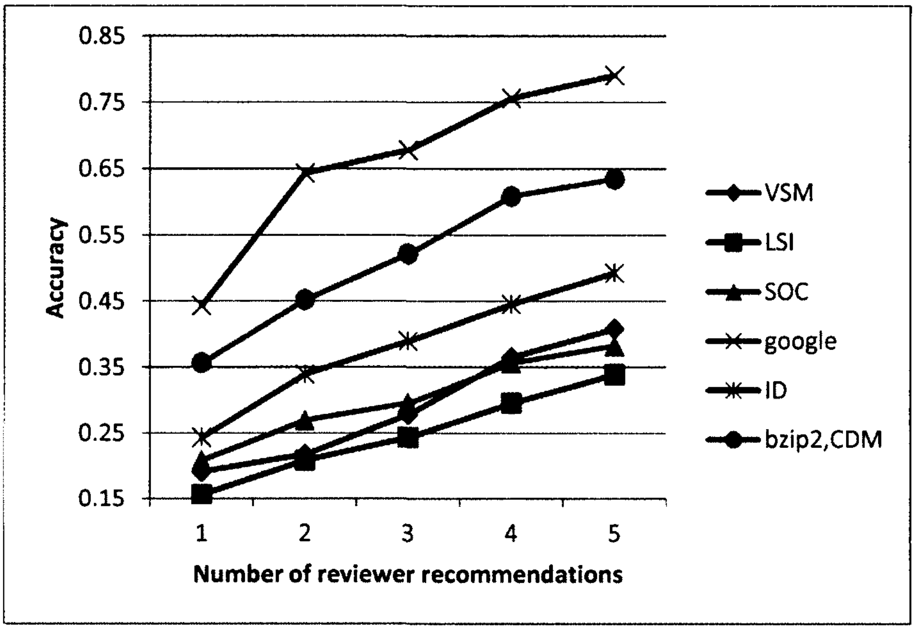


Figure 8.2: Comparison of accuracy for dataset1.

8.3.2 Dataset 2

The performance of the Google desktop method on dataset2 can be seen in Figure 8.3. Again the Google desktop method outperformed the other methods and by an even larger margin than on the first dataset. The method managed to achieve 85% correct assignment with the first result and 100% correct assignment when considering the top 3 results.

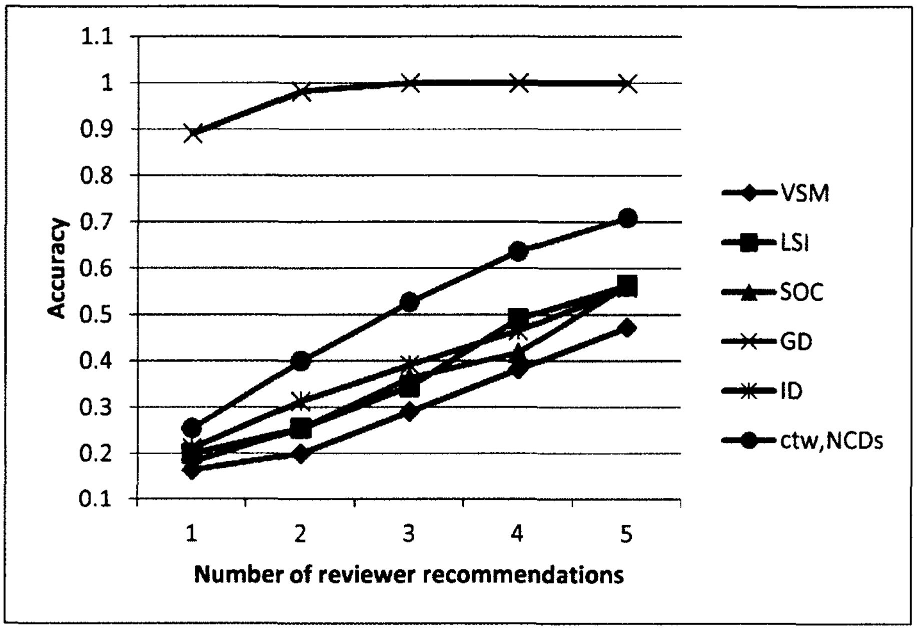


Figure 8.3: Comparison of accuracy for dataset2.

8.3.3 Overall

The average result over both datasets (Figure 8.4) clearly shows that the Google desktop method outperformed all of the other methods tested. The Google desktop method is at times 40 percent more accurate than the best alternative method. The high accuracy of this method shows that it could be used as an effective tool to help journal editors and conference chairs assign reviewers.

It should be pointed out that Google desktop is a highly refined information retrieval system. It may be based on standard methods such as VSM and LSI but it most likely includes many corporate secrets that improve performance. These improvements might include things such as separating documents into multiple sections which are given different weights, using word relations learned from analyzing a much larger dataset, or expanding search queries using known synonyms. Therefore it should not come as a surprise that Google performs better than the standard methods. However we have shown in the previous chapter that information distance performs better than these standard methods so it is possible that the performance of Google desktop method could be improved if it made use of information distance.

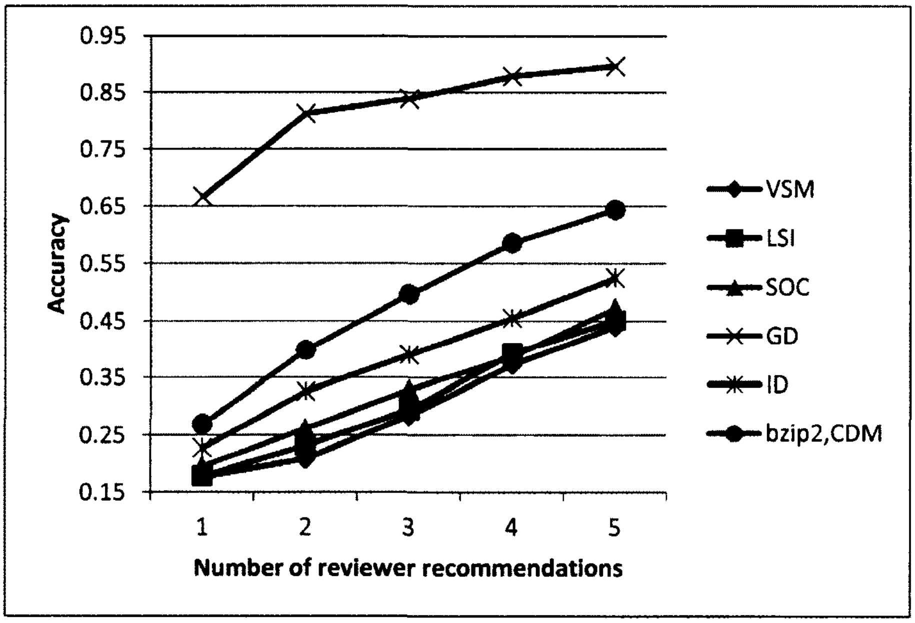


Figure 8.4: Comparison of accuracy over both datasets.

9 Conclusions and future work

9.1 Conclusions

Information distance is an effective method for determining document similarity in an automated reviewer assignment system. The results from Chapter 7 show that, using the best combination of compression program and distance equation, the information distance method outperforms the VSM and LSI methods that have been previously used in automated reviewer assignment systems.

The best choice of compressor and information distance equation for assigning reviewers using the full text of papers based on the results of Chapter 4 is the combination of the compressor bzip2 with either CDM or $Cdshare$.

Based on the results from Chapter 5 the information distance method needs to have at least 500 words of a paper to give good results. Also using the full text of a paper can actually hurt the results which can be attributed to the compression program doing a worse job of approximating the Kolmogorov complexity of larger files. Taking efficiency into account the results suggest it would be best to use only the first 500 words of a paper since this is the minimum number that gives good results. The best performing combination of distance equation and compression program when using the first 500 words was ctw with either NCD or $NCDs$. Further research could be done to determine whether taking text from other locations in a paper such as the end of the paper or beginning and end of sections would improve results further.

The results from Chapters 6 show it is possible to improve the performance of the information distance method for comparing text documents by using tools from the vector space model to help compressors create a better approximation of Kolmogorov complexity.

The Google desktop method introduced in Chapter 8 allows us to use an advanced information retrieval system like Google in automated reviewer assignment. The performance of this method is very good, and could be used in practice to help editors and conference chairs assign reviewers.

The methods presented in this research focus on calculating a good match between papers and potential reviewers' research interests. For this reason the results have wider implications in other problems that involve determining the similarity of text documents. This also means that the methods used in this research would need to be used as part of a larger reviewer assignment system. This larger system would need to deal with a variety of other considerations such as conflicts of interest, whether the paper that is being reviewed cites papers by potential reviewers or similar to those written by potential reviewers, determining weighting for more recent papers by potential reviewers, as well as optimizing matches so that the reviewing workload is distributed relatively evenly between reviewers.

9.2 Future work

Future research on the information distance method could involve finding ways to improve the approximation of the Kolmogorov complexity of text documents. The hybrid method in Chapter 6 could be further extended perhaps as part of a text specific compressor that would approximate the Kolmogorov complexity of text documents better than general compression programs.

Further research on the Google desktop method could experiment with the algorithm by using search results from information retrieval systems other than Google desktop. Comparisons between using the search results of Google desktop and using those of open source systems such as Lucene would be of particular interest. These comparisons could help in determining if the high performance of this method is directly related to the Google desktop search algorithm and how the performance of this method is affected by the search algorithm used.

References

[1] C. Basu, W. Cohen, H. Hirsh, C. Nevill-Manning, “Technical paper recommendation a study in combining multiple information sources”, *Journal of Artificial Intelligence Research* 1, 2011, pp. 231-252.

[2] C. H. Bennett, P. Gacs, M. Li, P. M. B. Vitanyi, W. H. Zurek “Information distance”, *IEEE Trans. on Inform. Theory,* vol. 44, no. 4, July 1998, pp. 1407-1423.

[3] H. Buhrman, T. Jaing, M. Li, P Vitanyi, “New Applications of the Incompressibility Method: Part II”, *Theoretical Comp. Sci.*, vol. 235, no. 1, Mar. 2000, pp. 59-70.

[4] H. Buhrman, M. Li, J. Tromp, P. Vitányi, “Kolmogorov random graphs and the incompressibility method” *SIAM J. Comput.,* vol. 29, no. 2, Oct. 1999, pp. 590-599.

[5] C.S. Calude, K. Salomaa, S. Yu, “Additive distances and quasi-distances between words”, *J. of Universal Comput. Sci.*, vol. 8, no. 2, 2002, pp. 141–152.

[6] M. Cebrian, M. Alfonseca, A. Ortega, “Common pitfalls using the normalized compression distance: what to watch out for in a compressor”, *Comm. in Inform. and Syst.,* vol. 5, no. 4, 2005, pp. 367-384.

[7] M. Charikar, E. Lehman, D. Liu, R. Panigrahy, M. Prabhakaran, A. Rasala, A. Sahai, A. Shelat, “Approximating the smallest grammar: Kolmogorov complexity in natural models” in *Proc. of the Thirty-Fourth Annual ACM Symp. on theory of Computing,* Montreal, Quebec, Canada, May 19 - 21, 2002, pp. 792-801.

[8] X. Chen, B. Francia, M. Li, B. McKinnon, A. Seker, “Shared information and program plagiarism detection”, *IEEE Trans. on Inform. Theory,* vol. 50, no. 7, July 2004, pp. 1545-1551.

[9] X. Chen, S. Kwong, M. Li, “A compression algorithm for DNA sequences and its applications in genome comparison” in *Proc. of the Fourth Annual int. Conference on Computational Molecular Biology*, Tokyo, Japan, 2000, pp. 107.

[10] S. Chen, B. Ma, K. Zhang, “On the similarity metric and the distance metric”, *Theoretical Comp. Sci.*, vol. 410, no.24-25, May 2008, pp. 2365-2376.

[11] R. Cilibrasi, P. Vitányi, R. De Wolf, “Algorithmic clustering of music based on string compression”, *Comput. Music J.,* vol. 28, no. 4, Dec. 2004, pp. 49-67.

[12] R. Cilibrasi, P. M. B. Vitanyi, “Clustering by compression”, *IEEE Trans. on Inform. Theory,* vol. 51, no. 4, April 2005, pp. 1523-1545.

[13] R. L. Cilibrasi, P. M. Vitanyi, “The Google similarity distance”, *IEEE Trans. on Knowl. and Data Eng.* Vol. 19, no. 3, Mar. 2007, pp. 370-383.

[14] S. Dumais, J. Nielsen, “Automating the assignment of submitted manuscripts to reviewers”, in *proceedings of the 15th Annual International ACM SIGIR Conference on Research and development in information retrieval,* Denmark, June 1992, pp. 233-244.

[15] D. Gondra, R. Heisterkamp, “Content-based image retrieval with the normalized information distance”, *Comput. Vision and Image Understanding,* vol. 111, no. 2, Aug. 2008, pp. 219-228.

[16] P. D. Grünwald, P. M. Vitányi, "Kolmogorov complexity and information theory. With an interpretation in terms of questions and answers", *J. of Logic, Lang. and Inf.*, vol. 12, no. 4, Sep. 2003, pp. 497-529.

[17] D. Hartvigsen, J. Wei, R. Czuchlewski, "The conference paper-reviewer assignment problem", *Decision Sciences*, Vol. 30, No. 3, 1999, pp. 865-876.

[18] S. Hettich, M. Pazzani, "Mining for proposal reviewers: lessons learned at the national science foundation", in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and data Mining,* Philadelphia, PA, USA, August 2006, pp.862-871.

[19] L. Ilie, S. Yu, K. Zhang, "Word complexity and repetitions in words", *Int. J. of Found. of Comput. Sci.*, vol. 15, no. 1, 2004, pp. 41-56.

[20] A. Islam, D. Inkpen, "Second order co-occurrence PMI for determining the semantic similarity of words", in *Proceedings of the International Conference on Language Resources and Evaluation,* Genoa, Italy, 2006, pp. 1033-1038.

[21] A. Islam, D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity", *ACM Trans. on Knowl. Discov. from Data,* vol. 2, no. 2, July 2008, Article 10.

[22] T. Jaing, M. Li, P Vitanyi, "New Applications of the Incompressibility Method", *The Comput. J.*, vol. 42, no. 4, 1999, pp. 287-293.

[23] B. Juba, "Estimating relatedness via data compression" in *Proc. of the 23rd int. Conference on Machine Learning,* Pittsburgh, Pa., USA, June 25 - 29, 2006, pp. 441-448.

[24] M. Karimzadehgan, C. Zhai, G. Belford, "Multi-aspect expertise matching for reviewer assignment", in *proceedings of CIKM,* Napa Valley, California, USA, October 2008, pp. 1113-1122.

[25] E. Keogh, S. Lonardi, C. A. Ratanamahatana, "Towards parameter-free data mining" in *Proc. of the Tenth ACM SIGKDD int. Conference on Knowledge Discovery and Data Mining*, Seattle, Wash., USA, Aug. 22 - 25, 2004, pp. 206-215.

[26] M. Li, "Information distance and its applications", *Int. J. of Found. of Comput. Sci.,* vol. 18, no. 4, 2007, pp. 669-681.

[27] M. Li, X. Chen, X. Li, B. Ma, P. M. B. Vitanyi, "The similarity metric", *IEEE Trans. on Inform. Theory,* vol. 50, no. 12, Dec. 2004, pp. 3250-3264.

[28] M. Li, P.M.B. Vitanyi, "Algorithmic Complexity", in *Int. Encyclopedia of the Social & Behavioral Sci.*, Pergamon, Oxford, 2001.

[29] M. Li, P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*, 3rd ed., New York, N.Y.: Springer, 2008.

[30] F. Li, X. Zhang, X. Zhu, "Answer validation by information distance calculation", in *Coling 2008: Proc. of the 2nd Workshop on Inform. Retrieval for Question Answering*, Manchester, UK, Aug. 24 - 24, 2008, pp. 42-49.

[31] C. Long, X. Zhu, M. Li, B. Ma, "Information shared by many objects" in *Proc. of the 17th ACM Conference on inform. and Knowledge Manage.,* Napa Valley, Calif., USA, Oct. 26 - 30, 2008, pp. 1213-1220.

[32] D. Parry, "Use of Kolmogorov distance identification of web page authorship, topic and domain" presented at *The Open Source Web Information Retrieval Workshop*, Compiegne, France, Sept. 19, 2005.

[33] C. P. Schnorr, "The process complexity and effective random tests" in *Proc. of the Fourth Annual ACM Symp. on theory of Computing*, Denver, Colo., USA, May 01 - 03, 1972, pp. 168-176.

[34] M. Sipser, "A complexity theoretic approach to randomness" in *Proc. of the Fifteenth Annual ACM Symp. on theory of Computing,* 1983, pp. 330-335.

[35] G. P. Telles, R. Minghim, F. V. Paulovich, "Normalized compression distance for visual analysis of document collections", *Comput. and Graphics*, vol. 31, no. 3, June 2007, pg 327-337.

[36] P. M. B. Vitanyi, "Universal Similarity", in *Proc. IEEE ITSOC Inform. Theory Workshop on Coding and Complexity*, Aug. 29-Sept. 1, 2005, Rotorua, New Zealand.

[37] F. Wang, B. Chen, Z. Miao, "A survey on the reviewer assignment problem", *New frontiers in applied artificial intelligence,* no. 5027, 2008, pp. 718-727.

[38] Y. Wang, S. Gong, "Refining image annotation using contextual relations between words" in *Proc. of the 6th ACM int. Conference on Image and Video Retrieval,* Amsterdam, The Netherlands, July 09 - 11, 2007, pp. 425-432.

[39] D. Yarowsky, R. Florian, “Taking the Load off the Conference Chairs: Towards a Digital Paper-routing Assistant”, in *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in NLP and Very-Large Corpora*, 1999, pp. 220–230.

[40] X. Zhang, Y. Hao, X. Zhu, M. Li, “New information distance measure and its application in question answering system”, *J. Comput. Sci. Technol.,* vol. 23, no. 4, July 2008, pp. 557-57.

Appendix

Full Results – Comparison of information distances and compressors

Dataset 1 - Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCD	0.286957	0.382609	0.434783	0.495652	0.513043
ppmz,NCDs	0.286957	0.373913	0.4	0.469565	0.530435
ppmz,CDM	0.269565	0.347826	0.408696	0.495652	0.521739
ppmz,Cdshare	0.269565	0.347826	0.408696	0.495652	0.521739
ppmz,Cdmin	0.078261	0.391304	0.408696	0.469565	0.513043
gzip,NCD	0.234783	0.330435	0.33913	0.426087	0.495652
gzip,NCDs	0.252174	0.373913	0.417391	0.452174	0.478261
gzip,CDM	0.286957	0.33913	0.382609	0.434783	0.486957
gzip,Cdshare	0.286957	0.33913	0.382609	0.434783	0.486957
gzip,Cdmin	0.069565	0.313043	0.4	0.434783	0.486957
bzip2,NCD	0.278261	0.321739	0.391304	0.417391	0.495652
bzip2,NCDs	0.173913	0.217391	0.243478	0.269565	0.269565
bzip2,CDM	0.356522	0.452174	0.521739	0.608696	0.634783
bzip2,Cdshare	0.330435	0.408696	0.486957	0.582609	0.617391
bzip2,Cdmin	0.06087	0.173913	0.226087	0.243478	0.304348
7zip,NCD	0.313043	0.373913	0.426087	0.452174	0.513043
7zip,NCDs	0.347826	0.391304	0.443478	0.495652	0.573913
7zip,CDM	0.278261	0.321739	0.434783	0.46087	0.495652
7zip,Cdshare	0.278261	0.321739	0.434783	0.469565	0.495652
7zip,Cdmin	0.086957	0.234783	0.269565	0.33913	0.408696
ctw,NCD	0.295652	0.382609	0.434783	0.478261	0.530435
ctw,NCDs	0.147826	0.182609	0.252174	0.330435	0.373913
ctw,CDM	0.295652	0.330435	0.356522	0.443478	0.495652
ctw,Cdshare	0.295652	0.33913	0.356522	0.443478	0.495652
ctw,Cdmin	0.069565	0.313043	0.365217	0.443478	0.469565
zpaq,NCD	0.295652	0.4	0.426087	0.46087	0.565217
zpaq,NCDs	0.295652	0.382609	0.434783	0.504348	0.556522
zpaq,CDM	0.356522	0.426087	0.452174	0.486957	0.513043
zpaq,Cdshare	0.356522	0.426087	0.452174	0.469565	0.513043
zpaq,Cdmin	0.069565	0.243478	0.304348	0.356522	0.434783
ID	0.243478	0.33942	0.389855	0.445507	0.493043

Dataset 1 - Average accuracy based on compressor

x	1	2	3	4	5
ppmz	0.238261	0.368696	0.412174	0.485217	0.52
gzip	0.226087	0.33913	0.384348	0.436522	0.486957
bzip2	0.24	0.314783	0.373913	0.424348	0.464348
7zip	0.26087	0.328696	0.401739	0.443478	0.497391
ctw	0.22087	0.309565	0.353043	0.427826	0.473043
zpaq	0.274783	0.375652	0.413913	0.455652	0.516522

Dataset 1 - Average accuracy based on distance

x	1	2	3	4	5
NCD	0.284058	0.365217	0.408696	0.455072	0.518841
NCDs	0.250725	0.32029	0.365217	0.42029	0.463768
CDM	0.307246	0.369565	0.426087	0.488406	0.524638
Cdshare	0.302899	0.363768	0.42029	0.482609	0.521739
Cdmin	0.072464	0.278261	0.328986	0.381159	0.436232

Dataset 2 – Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCD	0.145455	0.2	0.272727	0.4	0.454545
ppmz,NCDs	0.218182	0.272727	0.363636	0.436364	0.527273
ppmz,CDM	0.2	0.309091	0.345455	0.4	0.581818
ppmz,Cdshare	0.2	0.309091	0.345455	0.4	0.581818
ppmz,Cdmin	0.236364	0.309091	0.4	0.490909	0.563636
gzip,NCD	0.127273	0.181818	0.236364	0.327273	0.4
gzip,NCDs	0.163636	0.218182	0.254545	0.363636	0.4
gzip,CDM	0.2	0.381818	0.472727	0.509091	0.654545
gzip,Cdshare	0.2	0.381818	0.472727	0.509091	0.654545
gzip,Cdmin	0.254545	0.345455	0.454545	0.527273	0.6
bzip2,NCD	0.254545	0.345455	0.436364	0.527273	0.636364
bzip2,NCDs	0.181818	0.309091	0.436364	0.545455	0.6
bzip2,CDM	0.181818	0.345455	0.472727	0.563636	0.654545
bzip2,Cdshare	0.181818	0.345455	0.472727	0.563636	0.654545
bzip2,Cdmin	0.290909	0.4	0.490909	0.581818	0.618182
7zip,NCD	0.236364	0.345455	0.363636	0.545455	0.563636
7zip,NCDs	0.236364	0.254545	0.290909	0.363636	0.490909
7zip,CDM	0.254545	0.345455	0.436364	0.490909	0.672727
7zip,Cdshare	0.254545	0.345455	0.436364	0.509091	0.690909
7zip,Cdmin	0.272727	0.363636	0.436364	0.472727	0.545455
ctw,NCD	0.181818	0.236364	0.272727	0.381818	0.436364
ctw,NCDs	0.254545	0.4	0.527273	0.636364	0.709091
ctw,CDM	0.2	0.290909	0.363636	0.381818	0.472727
ctw,Cdshare	0.2	0.290909	0.363636	0.381818	0.472727
ctw,Cdmin	0.145455	0.254545	0.4	0.436364	0.527273
zpaq,NCD	0.254545	0.345455	0.436364	0.454545	0.509091
zpaq,NCDs	0.181818	0.290909	0.363636	0.454545	0.509091
zpaq,CDM	0.218182	0.327273	0.363636	0.418182	0.527273
zpaq,Cdshare	0.218182	0.327273	0.363636	0.418182	0.527273
zpaq,Cdmin	0.218182	0.327273	0.418182	0.472727	0.527273
ID	0.212121	0.313333	0.392121	0.465455	0.558788

Dataset 2 – Average accuracy based on compressor

x	1	2	3	4	5
ppmz	0.2	0.28	0.345455	0.425455	0.541818
gzip	0.189091	0.301818	0.378182	0.447273	0.541818
bzip2	0.218182	0.349091	0.461818	0.556364	0.632727
7zip	0.250909	0.330909	0.392727	0.476364	0.592727
ctw	0.196364	0.294545	0.385455	0.443636	0.523636
zpaq	0.218182	0.323636	0.389091	0.443636	0.52

Dataset 2 – Average accuracy based on distance

x	1	2	3	4	5
NCD	0.2	0.275758	0.336364	0.439394	0.5
NCDs	0.206061	0.290909	0.372727	0.466667	0.539394
CDM	0.209091	0.333333	0.409091	0.460606	0.593939
Cdshare	0.209091	0.333333	0.409091	0.463636	0.59697
Cdmin	0.236364	0.333333	0.433333	0.49697	0.563636

Overall – Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCD	0.216206	0.291304	0.353755	0.447826	0.483794
ppmz,NCDs	0.252569	0.32332	0.381818	0.452964	0.528854
ppmz,CDM	0.234783	0.328458	0.377075	0.447826	0.551779
ppmz,Cdshare	0.234783	0.328458	0.377075	0.447826	0.551779
ppmz,Cdmin	0.157312	0.350198	0.404348	0.480237	0.53834
gzip,NCD	0.181028	0.256126	0.287747	0.37668	0.447826
gzip,NCDs	0.207905	0.296047	0.335968	0.407905	0.43913
gzip,CDM	0.243478	0.360474	0.427668	0.471937	0.570751
gzip,Cdshare	0.243478	0.360474	0.427668	0.471937	0.570751
gzip,Cdmin	0.162055	0.329249	0.427273	0.481028	0.543478
bzip2,NCD	0.266403	0.333597	0.413834	0.472332	0.566008
bzip2,NCDs	0.177866	0.263241	0.339921	0.40751	0.434783
bzip2,CDM	0.26917	0.398814	0.497233	0.586166	0.644664
bzip2,Cdshare	0.256126	0.377075	0.479842	0.573123	0.635968
bzip2,Cdmin	0.175889	0.286957	0.358498	0.412648	0.461265
7zip,NCD	0.274704	0.359684	0.394862	0.498814	0.53834
7zip,NCDs	0.292095	0.322925	0.367194	0.429644	0.532411
7zip,CDM	0.266403	0.333597	0.435573	0.475889	0.58419
7zip,Cdshare	0.266403	0.333597	0.435573	0.489328	0.593281
7zip,Cdmin	0.179842	0.299209	0.352964	0.405929	0.477075
ctw,NCD	0.238735	0.309486	0.353755	0.43004	0.483399
ctw,NCDs	0.201186	0.291304	0.389723	0.483399	0.541502
ctw,CDM	0.247826	0.310672	0.360079	0.412648	0.48419
ctw,Cdshare	0.247826	0.31502	0.360079	0.412648	0.48419
ctw,Cdmin	0.10751	0.283794	0.382609	0.439921	0.498419
zpaq,NCD	0.275099	0.372727	0.431225	0.457708	0.537154
zpaq,NCDs	0.238735	0.336759	0.399209	0.479447	0.532806
zpaq,CDM	0.287352	0.37668	0.407905	0.452569	0.520158
zpaq,Cdshare	0.287352	0.37668	0.407905	0.443874	0.520158
zpaq,Cdmin	0.143874	0.285375	0.361265	0.414625	0.481028
ID	0.2278	0.326377	0.390988	0.455481	0.525916

Overall – Average accuracy based on compressor

x	1	2	3	4	5
ppmz	0.21913	0.324348	0.378814	0.455336	0.530909
gzip	0.207589	0.320474	0.381265	0.441897	0.514387
bzip2	0.229091	0.331937	0.417866	0.490356	0.548538
7zip	0.255889	0.329802	0.397233	0.459921	0.545059
ctw	0.208617	0.302055	0.369249	0.435731	0.49834
zpaq	0.246482	0.349644	0.401502	0.449644	0.518261

Overall – Average accuracy based on distance

x	1	2	3	4	5
NCD	0.242029	0.320487	0.37253	0.447233	0.50942
NCDs	0.228393	0.305599	0.368972	0.443478	0.501581
CDM	0.258169	0.351449	0.417589	0.474506	0.559289
Cdshare	0.255995	0.348551	0.41469	0.473123	0.559354
Cdmin	0.154414	0.305797	0.381159	0.439065	0.499934

Full Results – Size comparison for information distance methods

250 words – Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCD	0.217391	0.304348	0.365217	0.391304	0.426087
ppmz,NCDs	0.208696	0.295652	0.347826	0.391304	0.408696
ppmz,CDM	0.313043	0.382609	0.426087	0.46087	0.495652
ppmz,Cdshare	0.304348	0.373913	0.417391	0.452174	0.486957
ppmz,Cdmin	0.26087	0.347826	0.391304	0.434783	0.486957
gzip,NCD	0.217391	0.313043	0.356522	0.408696	0.46087
gzip,NCDs	0.226087	0.295652	0.347826	0.426087	0.486957
gzip,CDM	0.217391	0.313043	0.373913	0.408696	0.469565
gzip,Cdshare	0.226087	0.304348	0.365217	0.4	0.452174
gzip,Cdmin	0.243478	0.304348	0.356522	0.408696	0.452174
bzip2,NCD	0.243478	0.365217	0.4	0.452174	0.547826
bzip2,NCDs	0.243478	0.321739	0.382609	0.434783	0.513043
bzip2,CDM	0.243478	0.347826	0.4	0.434783	0.504348
bzip2,Cdshare	0.243478	0.347826	0.408696	0.443478	0.513043
bzip2,Cdmin	0.217391	0.347826	0.382609	0.452174	0.486957
7zip,NCD	0.182609	0.313043	0.373913	0.443478	0.521739
7zip,NCDs	0.226087	0.313043	0.356522	0.434783	0.46087
7zip,CDM	0.234783	0.330435	0.391304	0.478261	0.513043
7zip,Cdshare	0.243478	0.330435	0.408696	0.495652	0.530435
7zip,Cdmin	0.226087	0.347826	0.373913	0.469565	0.495652
ctw,NCD	0.269565	0.330435	0.365217	0.417391	0.434783
ctw,NCDs	0.295652	0.373913	0.434783	0.478261	0.53913
ctw,CDM	0.295652	0.330435	0.356522	0.4	0.452174
ctw,Cdshare	0.295652	0.33913	0.347826	0.391304	0.452174
ctw,Cdmin	0.243478	0.313043	0.347826	0.391304	0.452174
zpaq,NCD	0.243478	0.33913	0.391304	0.434783	0.469565
zpaq,NCDs	0.226087	0.304348	0.365217	0.417391	0.469565
zpaq,CDM	0.295652	0.373913	0.408696	0.434783	0.495652
zpaq,Cdshare	0.295652	0.373913	0.4	0.443478	0.504348
zpaq,Cdmin	0.278261	0.330435	0.417391	0.478261	0.582609
ID	0.249275	0.333623	0.382029	0.433623	0.485507

500 words – Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCD	0.304348	0.356522	0.417391	0.469565	0.486957
ppmz,NCDs	0.295652	0.356522	0.417391	0.486957	0.521739
ppmz,CDM	0.295652	0.365217	0.417391	0.434783	0.469565
ppmz,Cdshare	0.295652	0.365217	0.417391	0.443478	0.478261
ppmz,Cdmin	0.26087	0.313043	0.4	0.452174	0.486957
gzip,NCD	0.304348	0.4	0.452174	0.504348	0.565217
gzip,NCDs	0.295652	0.373913	0.408696	0.513043	0.573913
gzip,CDM	0.295652	0.373913	0.46087	0.486957	0.521739
gzip,Cdshare	0.295652	0.373913	0.469565	0.486957	0.521739
gzip,Cdmin	0.304348	0.391304	0.434783	0.46087	0.495652
bzip2,NCD	0.304348	0.4	0.46087	0.513043	0.53913
bzip2,NCDs	0.330435	0.4	0.434783	0.513043	0.53913
bzip2,CDM	0.295652	0.426087	0.469565	0.495652	0.53913
bzip2,Cdshare	0.304348	0.408696	0.452174	0.486957	0.521739
bzip2,Cdmin	0.26087	0.33913	0.373913	0.417391	0.469565
7zip,NCD	0.295652	0.408696	0.46087	0.504348	0.547826
7zip,NCDs	0.321739	0.408696	0.46087	0.504348	0.565217
7zip,CDM	0.252174	0.373913	0.426087	0.478261	0.530435
7zip,Cdshare	0.252174	0.373913	0.426087	0.486957	0.530435
7zip,Cdmin	0.208696	0.321739	0.417391	0.469565	0.513043
ctw,NCD	0.365217	0.452174	0.495652	0.521739	0.547826
ctw,NCDs	0.356522	0.452174	0.495652	0.547826	0.591304
ctw,CDM	0.313043	0.382609	0.426087	0.443478	0.469565
ctw,Cdshare	0.330435	0.391304	0.443478	0.486957	0.495652
ctw,Cdmin	0.330435	0.382609	0.434783	0.452174	0.513043
zpaq,NCD	0.295652	0.373913	0.46087	0.495652	0.530435
zpaq,NCDs	0.286957	0.356522	0.434783	0.478261	0.504348
zpaq,CDM	0.295652	0.382609	0.452174	0.513043	0.547826
zpaq,Cdshare	0.295652	0.373913	0.452174	0.504348	0.53913
zpaq,Cdmin	0.252174	0.365217	0.417391	0.469565	0.495652
ID	0.296522	0.381449	0.43971	0.484058	0.521739

750 words – Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCD	0.252174	0.33913	0.4	0.426087	0.469565
ppmz,NCDs	0.269565	0.356522	0.408696	0.452174	0.495652
ppmz,CDM	0.269565	0.373913	0.417391	0.434783	0.469565
ppmz,Cdshare	0.269565	0.391304	0.426087	0.443478	0.478261
ppmz,Cdmin	0.243478	0.330435	0.4	0.417391	0.478261
gzip,NCD	0.313043	0.4	0.443478	0.486957	0.521739
gzip,NCDs	0.321739	0.408696	0.443478	0.495652	0.521739
gzip,CDM	0.269565	0.356522	0.382609	0.417391	0.434783
gzip,Cdshare	0.269565	0.356522	0.382609	0.417391	0.443478
gzip,Cdmin	0.286957	0.417391	0.46087	0.486957	0.53913
bzip2,NCD	0.304348	0.373913	0.417391	0.452174	0.495652
bzip2,NCDs	0.278261	0.347826	0.4	0.434783	0.521739
bzip2,CDM	0.321739	0.382609	0.46087	0.513043	0.591304
bzip2,Cdshare	0.321739	0.382609	0.46087	0.521739	0.6
bzip2,Cdmin	0.304348	0.4	0.426087	0.478261	0.530435
7zip,NCD	0.269565	0.417391	0.46087	0.504348	0.547826
7zip,NCDs	0.304348	0.373913	0.408696	0.426087	0.478261
7zip,CDM	0.295652	0.4	0.452174	0.513043	0.582609
7zip,Cdshare	0.295652	0.417391	0.469565	0.521739	0.591304
7zip,Cdmin	0.130435	0.356522	0.408696	0.443478	0.495652
ctw,NCD	0.295652	0.4	0.443478	0.504348	0.53913
ctw,NCDs	0.286957	0.373913	0.391304	0.434783	0.495652
ctw,CDM	0.304348	0.408696	0.452174	0.504348	0.530435
ctw,Cdshare	0.313043	0.417391	0.46087	0.513043	0.53913
ctw,Cdmin	0.295652	0.391304	0.417391	0.443478	0.469565
zpaq,NCD	0.313043	0.4	0.434783	0.495652	0.521739
zpaq,NCDs	0.295652	0.382609	0.426087	0.452174	0.495652
zpaq,CDM	0.33913	0.426087	0.469565	0.504348	0.547826
zpaq,Cdshare	0.33913	0.426087	0.469565	0.495652	0.547826
zpaq,Cdmin	0.191304	0.391304	0.443478	0.469565	0.486957
ID	0.285507	0.386667	0.431304	0.470145	0.515362

1000 words – Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCD	0.295652	0.365217	0.452174	0.478261	0.530435
ppmz,NCDs	0.304348	0.382609	0.478261	0.513043	0.556522
ppmz,CDM	0.278261	0.373913	0.434783	0.469565	0.495652
ppmz,Cdshare	0.269565	0.373913	0.426087	0.46087	0.478261
ppmz,Cdmin	0.226087	0.382609	0.434783	0.521739	0.573913
gzip,NCD	0.295652	0.417391	0.46087	0.530435	0.565217
gzip,NCDs	0.33913	0.443478	0.478261	0.53913	0.643478
gzip,CDM	0.278261	0.33913	0.391304	0.443478	0.478261
gzip,Cdshare	0.278261	0.33913	0.4	0.443478	0.469565
gzip,Cdmin	0.156522	0.4	0.408696	0.452174	0.504348
bzip2,NCD	0.286957	0.382609	0.434783	0.504348	0.521739
bzip2,NCDs	0.295652	0.373913	0.417391	0.495652	0.521739
bzip2,CDM	0.304348	0.391304	0.46087	0.530435	0.573913
bzip2,Cdshare	0.304348	0.408696	0.469565	0.521739	0.565217
bzip2,Cdmin	0.2	0.417391	0.495652	0.53913	0.573913
7zip,NCD	0.304348	0.4	0.513043	0.530435	0.591304
7zip,NCDs	0.295652	0.373913	0.434783	0.469565	0.521739
7zip,CDM	0.330435	0.426087	0.46087	0.504348	0.53913
7zip,Cdshare	0.33913	0.434783	0.469565	0.513043	0.556522
7zip,Cdmin	0.078261	0.382609	0.417391	0.478261	0.530435
ctw,NCD	0.313043	0.382609	0.434783	0.469565	0.504348
ctw,NCDs	0.313043	0.356522	0.408696	0.46087	0.504348
ctw,CDM	0.33913	0.443478	0.495652	0.530435	0.53913
ctw,Cdshare	0.33913	0.443478	0.495652	0.521739	0.521739
ctw,Cdmin	0.269565	0.452174	0.513043	0.565217	0.6
zpaq,NCD	0.286957	0.4	0.417391	0.46087	0.46087
zpaq,NCDs	0.304348	0.373913	0.4	0.478261	0.530435
zpaq,CDM	0.321739	0.417391	0.469565	0.495652	0.521739
zpaq,Cdshare	0.321739	0.408696	0.452174	0.469565	0.495652
zpaq,Cdmin	0.113043	0.408696	0.486957	0.513043	0.556522
ID	0.27942	0.396522	0.450435	0.496812	0.534203

Comparison of average accuracy based on number of words

x	1	2	3	4	5
ID 250	0.249275	0.333623	0.382029	0.433623	0.485507
ID 500	0.296522	0.381449	0.43971	0.484058	0.521739
ID 750	0.285507	0.386667	0.431304	0.470145	0.515362
ID 1000	0.27942	0.396522	0.450435	0.496812	0.534203
ID Full	0.243478	0.33942	0.389855	0.445507	0.493043

Full Results – Hybrid information distance method

Dataset 1 – Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCDH	0.321739	0.365217	0.373913	0.417391	0.426087
ppmz,NCDsH	0.321739	0.391304	0.443478	0.46087	0.547826
ppmz,CDMH	0.330435	0.391304	0.452174	0.486957	0.53913
ppmz,CdshareH	0.33913	0.4	0.469565	0.495652	0.547826
ppmz,CdminH	0.095652	0.313043	0.417391	0.486957	0.521739
gzip,NCDH	0.33913	0.4	0.434783	0.495652	0.53913
gzip,NCDsH	0.347826	0.417391	0.486957	0.53913	0.547826
gzip,CDMH	0.286957	0.33913	0.391304	0.434783	0.486957
gzip,CdshareH	0.278261	0.330435	0.382609	0.426087	0.478261
gzip,CdminH	0.06087	0.243478	0.321739	0.408696	0.452174
bzip2,NCDH	0.278261	0.33913	0.373913	0.426087	0.469565
bzip2,NCDsH	0.147826	0.226087	0.252174	0.295652	0.365217
bzip2,CDMH	0.252174	0.321739	0.443478	0.478261	0.521739
bzip2,CdshareH	0.252174	0.321739	0.443478	0.478261	0.513043
bzip2,CdminH	0.095652	0.304348	0.347826	0.373913	0.4
7zip,NCDH	0.278261	0.330435	0.417391	0.434783	0.478261
7zip,NCDsH	0.304348	0.356522	0.365217	0.408696	0.469565
7zip,CDMH	0.278261	0.373913	0.452174	0.486957	0.530435
7zip,CdshareH	0.278261	0.373913	0.434783	0.486957	0.530435
7zip,CdminH	0.052174	0.208696	0.243478	0.286957	0.330435
ctw,NCDH	0.295652	0.373913	0.391304	0.434783	0.478261
ctw,NCDsH	0.173913	0.269565	0.330435	0.365217	0.408696
ctw,CDMH	0.26087	0.365217	0.426087	0.46087	0.504348
ctw,CdshareH	0.269565	0.391304	0.434783	0.469565	0.513043
ctw,CdminH	0.095652	0.4	0.452174	0.504348	0.530435
zpaq,NCDH	0.286957	0.356522	0.382609	0.417391	0.46087
zpaq,NCDsH	0.252174	0.330435	0.391304	0.452174	0.504348
zpaq,CDMH	0.321739	0.4	0.469565	0.495652	0.530435
zpaq,CdshareH	0.321739	0.382609	0.443478	0.469565	0.513043
zpaq,CdminH	0.086957	0.217391	0.269565	0.313043	0.365217
IDH	0.243478	0.341159	0.397971	0.43971	0.483478

Dataset 2 – Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCDH	0.381818	0.436364	0.490909	0.654545	0.709091
ppmz,NCDsH	0.327273	0.4	0.454545	0.545455	0.672727
ppmz,CDMH	0.381818	0.472727	0.563636	0.636364	0.745455
ppmz,CdshareH	0.363636	0.472727	0.563636	0.636364	0.745455
ppmz,CdminH	0.272727	0.418182	0.454545	0.490909	0.6
gzip,NCDH	0.309091	0.381818	0.527273	0.654545	0.690909
gzip,NCDsH	0.272727	0.309091	0.436364	0.545455	0.6
gzip,CDMH	0.345455	0.418182	0.490909	0.581818	0.672727
gzip,CdshareH	0.345455	0.418182	0.490909	0.581818	0.672727
gzip,CdminH	0.290909	0.363636	0.454545	0.472727	0.527273
bzip2,NCDH	0.345455	0.436364	0.527273	0.6	0.709091
bzip2,NCDsH	0.272727	0.472727	0.527273	0.618182	0.690909
bzip2,CDMH	0.272727	0.381818	0.563636	0.690909	0.727273
bzip2,CdshareH	0.272727	0.381818	0.563636	0.690909	0.727273
bzip2,CdminH	0.163636	0.272727	0.327273	0.436364	0.490909
7zip,NCDH	0.254545	0.327273	0.345455	0.490909	0.6
7zip,NCDsH	0.381818	0.454545	0.563636	0.618182	0.709091
7zip,CDMH	0.327273	0.436364	0.527273	0.563636	0.636364
7zip,CdshareH	0.309091	0.4	0.472727	0.527273	0.6
7zip,CdminH	0.127273	0.218182	0.254545	0.436364	0.527273
ctw,NCDH	0.290909	0.381818	0.454545	0.527273	0.545455
ctw,NCDsh	0.254545	0.472727	0.527273	0.581818	0.636364
ctw,CDMH	0.309091	0.327273	0.4	0.472727	0.472727
ctw,CdshareH	0.309091	0.327273	0.4	0.472727	0.472727
ctw,CdminH	0.254545	0.327273	0.418182	0.472727	0.581818
zpaq,NCDH	0.254545	0.327273	0.436364	0.545455	0.6
zpaq,NCDsH	0.309091	0.345455	0.472727	0.509091	0.563636
zpaq,CDMH	0.309091	0.4	0.436364	0.472727	0.472727
zpaq,CdshareH	0.309091	0.4	0.436364	0.472727	0.472727
zpaq,CdminH	0.163636	0.218182	0.290909	0.327273	0.4
IDH	0.292727	0.38	0.462424	0.544242	0.609091

Overall – Average accuracy of combinations

x	1	2	3	4	5
ppmz,NCDH	0.351779	0.400791	0.432411	0.535968	0.567589
ppmz,NCDsH	0.324506	0.395652	0.449012	0.503162	0.610277
ppmz,CDMH	0.356126	0.432016	0.507905	0.56166	0.642292
ppmz,CdshareH	0.351383	0.436364	0.516601	0.566008	0.64664
ppmz,CdminH	0.18419	0.365613	0.435968	0.488933	0.56087
gzip,NCDH	0.324111	0.390909	0.481028	0.575099	0.61502
gzip,NCDsH	0.310277	0.363241	0.46166	0.542292	0.573913
gzip,CDMH	0.316206	0.378656	0.441107	0.5083	0.579842
gzip,CdshareH	0.311858	0.374308	0.436759	0.503953	0.575494
gzip,CdminH	0.175889	0.303557	0.388142	0.440711	0.489723
bzip2,NCDH	0.311858	0.387747	0.450593	0.513043	0.589328
bzip2,NCDsH	0.210277	0.349407	0.389723	0.456917	0.528063
bzip2,CDMH	0.262451	0.351779	0.503557	0.584585	0.624506
bzip2,CdshareH	0.262451	0.351779	0.503557	0.584585	0.620158
bzip2,CdminH	0.129644	0.288538	0.337549	0.405138	0.445455
7zip,NCDH	0.266403	0.328854	0.381423	0.462846	0.53913
7zip,NCDsH	0.343083	0.405534	0.464427	0.513439	0.589328
7zip,CDMH	0.302767	0.405138	0.489723	0.525296	0.583399
7zip,CdshareH	0.293676	0.386957	0.453755	0.507115	0.565217
7zip,CdminH	0.089723	0.213439	0.249012	0.36166	0.428854
ctw,NCDH	0.293281	0.377866	0.422925	0.481028	0.511858
ctw,NCDsH	0.214229	0.371146	0.428854	0.473518	0.52253
ctw,CDMH	0.28498	0.346245	0.413043	0.466798	0.488538
ctw,CdshareH	0.289328	0.359289	0.417391	0.471146	0.492885
ctw,CdminH	0.175099	0.363636	0.435178	0.488538	0.556126
zpaq,NCDH	0.270751	0.341897	0.409486	0.481423	0.530435
zpaq,NCDsH	0.280632	0.337945	0.432016	0.480632	0.533992
zpaq,CDMH	0.315415	0.4	0.452964	0.48419	0.501581
zpaq,CdshareH	0.315415	0.391304	0.439921	0.471146	0.492885
zpaq,CdminH	0.125296	0.217787	0.280237	0.320158	0.382609
IDH	0.268103	0.36058	0.430198	0.491976	0.546285

Full Results – Comparison of information distance with other methods

Dataset 1 – Average accuracy of methods

x	1	2	3	4	5
VSM	0.191304	0.217391	0.278261	0.365217	0.408696
LSI	0.156522	0.208696	0.243478	0.295652	0.33913
SOC	0.208696	0.269565	0.295652	0.356522	0.382609
GD	0.443478	0.643478	0.678261	0.756522	0.791304
ID	0.243478	0.33942	0.389855	0.445507	0.493043

Dataset 2 – Average accuracy of methods

x	1	2	3	4	5
VSM	0.163636	0.2	0.290909	0.381818	0.472727
LSI	0.2	0.254545	0.345455	0.490909	0.563636
SOC	0.181818	0.254545	0.363636	0.418182	0.563636
GD	0.890909	0.981818	1	1	1
ID	0.212121	0.313333	0.392121	0.465455	0.558788

Overall – Average accuracy of methods

x	1	2	3	4	5
VSM	0.17747	0.208696	0.284585	0.373518	0.440711
LSI	0.178261	0.231621	0.294466	0.393281	0.451383
SOC	0.195257	0.262055	0.329644	0.387352	0.473123
GD	0.667193	0.812648	0.83913	0.878261	0.895652
ID	0.2278	0.326377	0.390988	0.455481	0.525916