M. Tech. (C. S.) Dissertation Report

Categorization and Automatic Linking of Web-pages

By

T.V.Sriram

Under the supervision of

Prof. Aditya Bagchi

Computer and Statistical Service Center
Indian Statistical Institute

and

Dr. Mandar Mitra

Computer Vision and Pattern Recognition Unit
Indian Statistical Institute
203, Barrackpore Trunk Road,
Calcutta-35.

Certificate of Approval

This is to certify that the thesis entitled, "**Categorization and Automatic Linking of Web-pages**", submitted by T.V.Sriram, towards partial fulfillment of the requirements for M. Tech. in Computer Science degree of the *Indian Statistical Institute, Calcutta,* is an acceptable work for the award of the degree.

Date: 27.7.2000

Bagchi

Prof. Aditya Bagchi
Computer and Statistical
Service Center,
Indian Statistical Institute.

A Bandyopadhyay
(External Examiner)

Mandar Mitra

Dr. Mandar Mitra
Lecturer,
Computer Vision and Pattern
Recognition Unit,
Indian Statistical Institute.

Abstract

The abundant availability of information in electronic form, especially in hyper-text format, has made Text Categorization and Automatic Hyper-text Linking very important.

In this dissertation an algorithm for Hierarchical Text Categorization is presented. The algorithm makes use of the similarity computations based on keyword matching. A study on the application of the algorithm to the documents collected from Google Directory-Science (http://directory.google.com) is presented.

In this dissertation an algorithm for Automatic Hyper-text Link Typing of HTML pages is also presented. The algorithm makes use of the similarity of part-pairs, obtained after dividing the documents into parts and comparing each part-pair individually. A set of measures to find the link type between any two documents is proposed.

Acknowledgement

I would like to thank Prof. Aditya Bagchi and Dr. Mandar Mitra, for without their help it would have been impossible to complete this work. As my exposure to research was meager, their guidance was invaluable. When solutions led nowhere, they helped me choose the right direction. Their encouragement at each step motivated me to learn more and work on new ideas, though they were naïve.

I also thank the Institution for giving me such an opportunity and supplementing me with the required facilities when needed.

I would like to once again thank Dr Mandar Mitra, for helping me in getting acquainted with the Smart System, which was the foundation for this work.

Table of Contents

ABSTRACT **1**

ACKNOWLEDGEMENT **2**

TABLE OF CONTENTS **3**

1. MOTIVATION **5**

1.1 HIERARCHICAL TEXT CATEGORIZATION 5
1.2 LOST IN HYPERSPACE 5
1.3 IMPORTANCE OF LINK TYPES 6

2. PROBLEM STATEMENT **7**

3. BACKGROUND **7**

3.1 THE VECTOR SPACE MODEL 7
3.2 THE MODEL 7
3.3 TERM WEIGHTING 7
3.3.1 Term Frequency Factor *8*
3.3.2 Inverse Document Frequency Factor (idf) *8*
3.3.3 Document Length Normalization *8*

4 DOCUMENT CATEGORIZATION **9**

4.1 INTRODUCTION 9
4.2 HIERARCHY DESCRIPTION 9
4.3 METHOD ADOPTED 10
4.3.1 Weighting Scheme *10*
4.3.2 Training Phase *10*
4.3.3 Testing Phase *10*
4.4 EXPERIMENTATION 11
4.4.1 Evaluation *11*
4.4.2 Evaluation Results *11*
4.5 PRELIMINARY ANALYSIS 11

5. AUTOMATIC HYPERTEXT LINKING **13**

5.1 INTRODUCTION 13
5.1.1 Link Typing *13*
5.1.2 Taxonomy of link types *13*
5.2 DOCUMENT LINK TYPING 14
5.2.1 Relationship Graph *14*
5.2.2 Graph Formation *14*
5.2.3 Graph Simplification *14*
5.3 DOCUMENT RELATIONSHIP VISUALISATION 16
5.4 GRAPH FEATURE 16
5.5 GRAPH PATTERNS 17
5.5.1 Revision pattern *17*
5.5.2 Duplicate pattern *18*
5.5.3 Part-of-a-document *18*
5.5.4 Summary and Expansion *19*
5.5.5 Specialization *19*
5.5.6 Equivalent Document *20*
SPECIAL 24

6 CONTRIBUTIONS AND FUTURE WORK **26**

7 APPENDIX **27**

7.1 DOCUMENT COLLECTION 27

8 BIBLIOGRAPHY **28**

1. Motivation

The amount of information in the World Wide Web(WWW) has increased enormously over the years. It is expected that a large portion of the world population will form an integral part of the WWW at the end of the 21st century. With more people in the web the amount of electronically available information is expected to increase much more.

The electronic information – the information stored in electronic media can be broadly classified as,

Text- the information that is available in plenty, stored in Standardized formats, such as ASCII, UNICODE etc. Information stored as text occupies very little space.

Images- The information in the form of electronic images. For example, Bio-Medical Images, Astronomical Images have gained importance in the recent past. Images take a large space for storage. With storage space becoming cheap, the information content in the form of images is increasing.

Audio- audio information is available as Lectures, Broadcasts etc.

Video- video information is mainly the combination of image and audio information.

With the increase in electronic information the problem of organizing huge information-bases has gained considerable importance in recent times. Among the information types discussed earlier, organizing text is considered relatively simple. Some of the information contained in other media can also be converted to text. Thus organization of text collections is very important.

Collections of text (e.g. web pages) are often organized into a hierarchy of categories for ease of browsing or retrieval of information. Because of the volume involved the process of manually categorizing documents is almost impossible. Thus the process of automatically categorizing text documents is a challenging problem.

The number of Web-Sites has increased in the recent past. The problem of providing links to various Web-Sites for related information, has arisen due to the lack of information transparency (because it is difficult for a person to keep track of all the related pages for a topic) across web-sites. So far the only such universal link providers are the search engines i.e., if the web page has no link to the required information, the Search engines are resorted to for providing the same.

It will be of great help if the links to a particular page are suggested automatically. Suppose after categorizing the documents, the web-sites of the respective documents are recommended links to related pages in the same category, browsing can be made effortless after incorporating those links.

1.1 Hierarchical Text Categorization

Given two sets of documents, one set categorized initially, the documents in the other set are categorized over the set of categorized documents in Hierarchical Text Categorization. This idea of categorizing documents over an existing category hierarchy is as suggested in [AISkM], with the increase in the number of categories, precision and recall reduces. This is attributed to the fact that as the scope of the collection increase the terms tend to become polysemous. Especially acronyms, which are 3 to 4 letters long, tend to get reused in completely different categories. With increase in the number of categories the size of the domain specific knowledge vocabulary increases, asking for huge system resources. In this type of categorization no such restrictions

1.2 Lost in Hyperspace

When browsing web-sites, especially large ones, there are high chances that a person can get lost in the web-site. This happens if a person wants to learn a new subject in depth from an online document. This is termed as the 'lost in hyperspace' problem. There are a few solutions for the problem, which are not effective enough. Typed links constitute one approach that is intended to ameliorate this problem.

1.3 Importance of Link Types

The link type is the semantic meaning associated with the link. The WWW does not support the link types, rather the web page designer has to provide annotations in addition to the existing text. For example,

"<A href="…..">Information Retrieval </A> technique", does not suggest what is behind the Link "Information Retrieval". So a good page designer uses

"Information Retrieval Techniques" <A href="……..">(Definition)</A>. Thus a person knows what is behind the link. The extra text is the annotation.

Annotations may not work well in all cases. Suppose a link leads to multiple destinations. Thus the link type must be a part of the link itself. The advantages gained by link types are

1. The overall complexity of the graph under one link type is less, thus can be handled easily.
2. The browsing system can be made to respond differently with different link types.
3. Enabling only those link types expected by the user at that time.

Links introduce complex inter-document graphs, thus creating the lost in hyperspace problem. Link Types add just a label to the link, and hence do not solve the problem as such. Still the advantages far outweigh the disadvantage.

2. Problem Statement

1. Given a set of documents categorized by manual means, each document in another set must be automatically placed in the correct existing category; if it does not belong to any of the category it must be rejected.
2. Given a set of HTML documents, the presence of a Link, and the semantic meaning of the link must be automatically suggested between any pair of HTML documents.

3. Background

3.1 The Vector Space Model

The Vector Space Model of Information Retrieval was developed by Salton and his students [AmitS]. This model transforms any given text (article, query, portion of an article, etc.,) into a vector in a very high-dimensional vector space. The closeness between any two documents can be visualized as that between their respective Vectors. In terms of information retrieval, when two vectors are close, then the corresponding texts are semantically related.

The Smart System, an experimental Information Retrieval System, uses the Vector space model. Many theories and techniques in the field of information retrieval, for example, automatic indexing and term weighting, evaluation of ranked systems, soft Boolean models, relevance feedback, document clustering, use of a thesaurus, were developed with the help of Smart.

3.2 The Model

In the Vector Space Model each document in a collection is viewed as an n-dimensional vector, with 'n' [1]concepts in the dictionary formed from the collection. The weighting of a vector is the procedure for assigning a weight to each dimension of the vector, in proportion to the importance of the respective concept to the document as a whole. For example if "Automatic, hypertext and construction" are the only words available in the dictionary, then the document vectors will be 3D vectors. Each dimension may be assigned a weight proportional to the number of times the concept (one of Automatic, Hypertext, or Construction) occurs in that document.

Let t1, t2, ..., tn be the concepts in the dictionary. Then the vector of a document Di is represented by

Di = <wi1, wi2, ..., win>

where, each wij is the weight assigned to the ith document vector along the jth dimension. For example, the value of wij could be proportional to the number of times the term tj occurs in document Di.

Taking the dimensions as orthogonal, the closeness of two vectors, "Vector Similarity" in the field of Information Retrieval, can be computed as their inner product. If the document vectors are P and Q, then

Similarity = P . Q

While doing this it is typically assumed that the terms in a document are independent (even though this is untrue).

3.3 Term Weighting

The process of finding the similarity between two documents boils down to computing their inner product. Then the entire effectiveness of the procedure depends on the proper assignment of weights for the terms[2] occurring in the document. Modern Information Retrieval Systems use three factors in determining the weights, the Term Frequency, the Inverse Document Frequency and the Document Length Normalization.

[1] Each Concept is a distinct word occurring in the dictionary formed from the document collection.

[2] Each instance of a Concept occurring in a document is a Term

3.3.1 Term Frequency Factor

The importance of a term increases with the frequency of occurrence of that term in the document. The number of times a term occurs in a document is called the Term Frequency (tf). The function of the Term Frequency that is used for weighting is called the Term Frequency Factor (tf factor). Some commonly used tf factors are:

Raw tf factor: the tf value is used as the tf factor.

Logarithmic tf factor: Rather than the raw tf, a logarithmic function on the raw tf works well. It was proposed on the basis that a match of one high frequency term must give a lesser contribution to the similarity than a match of two moderately weighted terms. Therefore, 1 + ln(tf) is a widely accepted tf factor.

3.3.2 Inverse Document Frequency Factor (idf)

In a document collection, a term occurring in many documents is less important than a term occurring in less number of documents in the collection. For example, 'telecommunication' occurs in a fewer number of documents in a collection than the word 'science', thus an occurrence of 'telecommunication' is much more important than the occurrence of 'science'. Thus the factor ln(N/df) is used as the Inverse Document Frequency factor, where N –number of documents in the collection and df- the number of documents in which the term occurs.

3.3.3 Document Length Normalization

Though the tf * idf gives a good estimate of the weight of a term in a document, the document length is also a significant factor. A term is likely to occur more frequently in a longer document than a short one. Also, larger the document length, more the number of distinct terms occurring. For these reasons a third factor called the Document Length Normalization factor is used. Some commonly used document length normalization factors are:

Cosine Normalization: Cosine normalization is a commonly used normalization technique in the vector space model. The vector inner product measure is used to compute the closeness (also called correlation) between two vectors. The inner product similarity measure does not compensate for document length differences. Instead, one has to introduce length normalization in the term weights when using the inner product measure.

For example, consider the Cosine correlation between a query vector Q and a document vector D

$$\cos(Q,D) = \frac{\sum_{i=1}^{T} w'_{qi} * w_{di}}{\left(\left(\sqrt{w_{q1}^2 + w_{q2}^2 + \ldots\ldots + w_{qT}^2}\right) * \left(\sqrt{w_{d1}^2 + w_{d2}^2 + \ldots\ldots + w_{dT}^2}\right)\right)}$$

Where w_{qi} is the *idf*tf* weight of term-I in the query vector, w_{di} is the *idf * tf* weight of term-I in the document vector. The cosine correlation is bound between 0 and 1 by the use of the Euclidean lengths of the individual vectors in the denominator. It implicitly normalizes for length variation of documents.

The cosine correlation can also be written as:

$$\cos(Q,D) = \sum_{i=1}^{T}\left(\left(\frac{w_{qi}}{\sqrt{w_{q1}^2 + w_{q2}^2 + \ldots\ldots + w_{qT}^2}}\right) \times \left(\frac{w_{di}}{\sqrt{w_{d1}^2 + w_{d2}^2 + \ldots\ldots + w_{dT}^2}}\right)\right)$$

which is same as the vector inner product similarity measure if individual tf * idf weights of a vector were divided by the Euclidean length of the vector. Since it is easier to implement inner product similarity in IR systems with inverted lists, as an implementation convenience, the denominator for individual vectors is pre-computed, and the tf * idf weights for terms are divided by it before being stored in the inverted lists. Doing this, one can now use vector inner product similarity to obtain cosine correlation between two vectors. The cosine normalization factor is computed as

$$\sqrt{w_1^2 + w_2^2 + \ldots\ldots + w_t^2}$$

where wi is the raw tf * idf weight for a term. Cosine normalization attacks both the reasons for normalization (higher tfs and more terms) in one step. Higher individual term frequencies increase individual wi values, increasing the penalty on the term weights. Also, if a document has more terms, the number of individual weights in the cosine factor (t in the above formula) increases, yielding a higher normalization factor.

Byte Length Normalization: More recently, a length normalization scheme based on the byte size of documents has been used in the Okapi system (and other systems). This normalization technique is based on approximations to the 2-Poisson model. This normalization formulation is used in conjunction with Okapi's tf formulation and the final normalized term weight is:

$$\frac{tf}{2\times\left(1-b+b\times\dfrac{document-length}{averaged-document-length}\right)+tf}$$

Where b is some constant, typically 0.75. Therefore, this normalization factor also attacks both the reasons for normalization in one shot.

4 Document Categorization

4.1 Introduction

Given a set of documents, the objective of Document categorization is to group the documents in a hierarchical fashion such that documents at each level discuss some particular aspect(s) of a topic discussed in a document located at a higher level in the hierarchy.

This can be achieved by starting with a flat organization of documents and then using clustering techniques to group documents. The groups can be further divided recursively to obtain the hierarchical organization of documents. There is also a bottom up approach, in which documents are clustered to form groups of small size. The groups are then combined to form larger groups recursively to get the hierarchical structuring.

Another approach is to start with an existing hierarchy and building the strength of the hierarchy, by inserting new documents in the correct categories. The documents to be categorized occur in two different sets. In one set the location of documents in the hierarchy is known, called the Training set and in the other it is not known, called the Testing set.

4.2 Hierarchy Description

In this study, we consider a set of documents that are strictly structured as a tree. This data set forms the training set of documents. The Hierarchy consists of documents, not only in the leaf level but also in the intermediate or internal non-leaf nodes. Thus each category in the hierarchy has some of its own representatives. The general property expected in a category is that it must represent the common vocabulary among the categories at the next level under it. Each category at the next level must have some distinctive vocabulary in addition to the common portion inherited from the parent category.

4.3 Method Adopted

The documents in the training set and the testing set are indexed to form their respective vectors, as mentioned in the Background section. The weighting is chosen such that document discrimination is easier.

4.3.1 Weighting Scheme

The Smart 'Ltu' weighting scheme was used to form the vector of each document. As suggested in the background section, the choice of term weighting scheme plays an important role in the performance of the vector based document similarity computation. The three factors chosen for term weighting are,

Tf factor = (1 + ln(tf))/(1+ln(average(tf)))
Idf factor = (ln(N/df))
Length Normalization = (1.0 - slope) * pivot + slope * # of unique terms
where slope and pivot are parameters obtained from experimentation.

The factors leading to the choice of the above Term Weighting scheme are discussed in detail in [AmitS].

As in the clustering problem, discrimination must be possible among similar documents.

4.3.2 Training Phase

First, a hierarchical data structure (tree) is formed from the structure of the hierarchy in the categorized data set. Each document in the training data set can be placed in a unique node of the tree (Training data is a tree not a DAG).

The nodes in the tree are of two types, the leaf nodes and the internal nodes. The leaf nodes represent the categories that have not been further categorized. The internal nodes represent those nodes that have sub-categories under them. Internal nodes may, in addition, contain some documents.

From the vectors of the documents, the vectors of the nodes must be computed. We use a Bottom Up approach. At the leaf level each node has only documents under it. The vector for a leaf node is computed from the centroid of the document vectors in that node. The vector of an internal node is computed as the centroid of the vectors corresponding to its children and the vectors for the documents contained in it. The document vectors mentioned here are document length normalized vectors [AmitS].

4.3.3 Testing Phase

The testing set vectors can be called Query vectors. Two thresholds are used, namely Upper Threshold and the lower threshold.

The following procedure is repeated for all the documents in the testing set,

Step1. The root node is put in a queue.

Step2. Quit if the queue is empty.

Step3. The first node is removed from the queue, the similarity between its vector and the Query vector, and also the similarity between the Query vector and the vectors of its children is computed. Now the thresholds are applied,

Step4. If the node's similarity is less than the lower threshold, and if the node is the root then the document is rejected. The next testing document is chosen and the control is transferred to step1.

Else if not the root then the search for categories under it is stopped. The control is transferred to step2.

Else if similarity is greater than lower threshold

The maximum similarity among the node and its children with the query vector is found. If the maximum is the node itself then print the category of the node. The control is transferred to step2.

Else a threshold similarity = maximum similarity * upper threshold is found. All child nodes with similarity greater than the threshold are inserted in the queue. The control is transferred to step2. (This ensures that the test documents can be under multiple categories.)

4.4 Experimentation

The documents used in the experimentation were obtained from the web. The documents were extracted using a spider [Appendix] from the Web-Site http://directory.google.com/Top/Science. The documents obtained from the Google categorization hierarchy, were stored in the local file system along with the information about their position in the hierarchy. While collecting it was made sure that the documents were at-least 8KB in size. The entire size of the collection was 212 MB, with 7480 documents.

The Collection included html documents from Agriculture, Astronomy, Biology, Chemistry, Earth Sciences, Environment, Maths, Physics, Social Science and Technology.

The documents collected were indexed using the SMART system (refer Background).

The 7480 documents were divided into two sets training set with 6000 documents and testing set with 1480 documents.

4.4.1 Evaluation

A program was written to implement the procedures involved in training, testing and evaluation. The program used the various utilities provided by the SMART system, for carrying out the task. The evaluation of the categorization is based on the Google categorization (stored while collecting documents). We propose the following evaluation measure for the categorization problem: a fraction given by the largest overlap between any of the categories suggested (path traced to reach the category from root)and the category under which the document was placed in Google divided by the maximum length of the two paths from the root. This parameter returns a 1 for a correctly categorized document, a value between 0 and 1 for a document categorized in the same area but not in the same category. For example, let the document be categorized in Google as /a/b/c/d/e/f, and the program returns two categories /a/b/m and /l/p/q/r then the evaluation accuracy score = 2/6.

The program was run in the training phase, then in the testing phase with a lower threshold of 5 and an upper threshold of 0.9(refer Method Adopted section).

4.4.2 Evaluation Results

The categorization is evaluated on the basis of score obtained as mentioned earlier. The results gave an overall score of 0.355. The split up of the score in each category

Agriculture – 0.4552
Astronomy – 0.49355
Biology – 0.099
Chemistry – 0.43204
Earth Sciences – 0.28917
Environment – 0.3523
Math – 0.430337
Physics – 0.4030
Social Sciences – 0.3535
Technology – 0.25178

4.5 Preliminary Analysis

The category-wise score shows that the Biology, Technology and Earth Sciences fared well below the average, Environment and Social Sciences near the average and rest above the average. Biology had a score of 0.1. The number of training documents under this category was 2559 html documents, which was far more than the documents in other categories.

The examination of the centroid vectors at various levels of the training hierarchy suggests some of the reasons for getting a low Categorization accuracy score. Firstly the centroid vector for a category is computed from the documents and its sub-categories. If the centroid is computed as suggested in the method adopted section, as the number of documents increase, the similarity between the document vectors of documents in the category and the centroid of that category increases. For example, consider a document in the Technology category, dealing with the use of Lasers for tissue treatment. The concepts concerning Laser forms a part of Physics, while the tissues are supposed to come under Biology. While categorizing such a document, discrimination between Physics, Biology and Technology is based on the weight of the concepts representing the corresponding subjects. If the number of Physics documents is small, then though Lasers form only a very meager part of physics, related concepts may get higher weight because of lesser averaging steps from the category to the centroid representing Physics.

On the other hand the Technology category has a lot of documents, thus a document with Lasers and Tissue occurs in a category at a greater depth under Technology. Also the number of horizontal categories are more. These factors result in the Physics category being chosen instead of Technology. The centroid computation does not take into account the issues considered for proper term weighting [AmitS].Thus the centroid vector of a category with lesser number of documents tend to be favoured during the categorization process.

Secondly the idf factor in term weighting, helps in determining the importance of a term based on the number of documents in which it occurs in the collection. A term occurring in more number of documents gets lesser weight and vice versa. But in categorization if a term occurs in the same category, in many documents, the *idf* factor reduces its importance even when it is important in discriminating among categories. For example, consider the terms 'mathematics' and 'prime'. Both terms occur in almost same number of documents, thus have the same *idf*. As 'mathematics' occurs in all categories of science, *idf* does not affect much. Though 'prime' occurs mostly in mathematics category, it receives the same *idf* as 'mathematics'. As 'prime' can discriminate documents of mathematics from others its weight should not be reduced.

5. Automatic Hypertext Linking

5.1 Introduction

Lost in hyperspace problem must be avoided. For achieving this, association of semantic link types with links is necessary. As mentioned in (Section 1) generation of such link types must be made automatic. The following sections discuss in detail the process of automatic generation of link types.

5.1.1 Link Typing

A link type is a description of the relationship between the source and destination of a link. The link type cannot be determined by analyzing the destination document alone, as it may share different properties with different concepts. A detailed analysis of both the documents is required to establish the relationship.

Web browsers do not use link typing, forcing the page designer to include annotations as part of displayed text. These implicit links are fine to a small extent, but for example, handling a link to multiple destinations is a problem.

5.1.2 Taxonomy of link types

The following classes of Link types are taken from [Allan95].

5.1.2.1 Pattern matching links

This first broad class of link types is those that can be found easily using simple or sometimes fairly elaborate pattern matching techniques. An obvious example of such a link type is Definition, which can be found by matching words in a document to entries in a dictionary. In almost all cases, these links are from a word or phrase to a small document, and will occur outside of any specific context i.e., the destination document may be the same for the word or phrase, no matter where the word or phrase occurs.

We also group structural links into this class. Structural links are those that represent layout or possibly logical structure of a document. For example, links between chapters or sections, links from a reference to a figure to the figure itself, and links from a bibliographic citation to the cited work, are all structural links. We include this with pattern matching links because they are typically recognised by mark-up codes that are already embedded in the text. Even when a document is not marked up, structure is usually approximated using pattern analysis.

5.1.2.2 Manual links

Pattern matching links are a class, which is easy to detect automatically. At the extreme opposite end of the spectrum are manual links, those that we are currently unable to locate without human intervention. Identifying manual links requires text analysis at a level, which the Natural Language Understanding community is trying to achieve. They have had some significant success within constrained subject areas, so some manual links could be automatically described within those limited domains. Unfortunately, the techniques cannot

yet be extended to a general setting, so this class of link types remains inaccessible to automatic approaches.

5.1.2.3 Automatic links

Between the difficulty of manual links and the ease of pattern matching links, is the location of automatic links. These are links which cannot typically be located trivially using patterns, but which the automatic techniques can identify with marked success. Only this class of link types are considered in this work.

The automatic links, which can be identified, are:

- Revision links are a very simple class of relationship between texts, including both ancestor and descendant relationships. In the context of computer edited material where successive versions of the material are archived, either intentionally or as an artefact of the editing system, information which describes relations in some revision hierarchy is crucial. Even when the revision occurs over a much greater time---e.g., a revised edition of a textbook---it is useful to know the relationship.

Some revision links are fortunately extremely simple to find and maintain automatically, since they are flagged by the editing system e.g., version numbers of a file, backup copies of a text. (Those revision links would actually be classified as ``pattern matching" links.) When that information is absent, however, a different means must be used to find the relationship.

- Summary and expansion links are inverses of one another. A summary link type is attached to a link, which starts at a discussion of a topic and has as its destination a more condensed discussion of the same topic.
- Equivalence links represent documents with discussions of the same topic.

The **additional link types added** to the class in this dissertation are

- *Specialization links.* A specialization link starts at a general discussion of a topic and has as its destination a document that discusses some aspect of the topic in depth. For example a Specialization link starts from a document on "Tree Data Structures" and points to a document on "B Trees".
- *Part-of-a-document* links is used for identifying part of a document. These links come very handy in Web-Site maintenance.
- *Duplicate-Document links* is used to identify identical documents. These are useful for removing redundant pages from Web-Sites and from the results of web searches.

5.2 Document Link Typing

The Automatic link types presented in the Taxonomy of link types section, forms the major contribution of this dissertation. The suggested links under that class can be analysed graphically.

5.2.1 Relationship Graph

The relationship between two documents can be represented by a bipartite graph, with one set of nodes coming from one document and the other set from the other document.

The document parts form the nodes of the Graph, while the similarity between document parts form the links.

The nodes are weighted by the size of the represented parts, and the links with the similarity between the nodes that they link.

5.2.2 Graph Formation

The two given documents are divided into sections.

Each section of one document is compared with every section of the other document.

The similarity results between each part pair are used to form the links of the Graph.

5.2.3 Graph Simplification

5.2.3.1 Need for Graph simplification

With increasing document size the number of sections tends to increase, resulting in a graph with many links. The analysis of such a graph is not only difficult but also expensive in both time and space.

5.2.3.2 Initial Treatment

The objective of this step is to process the graph such that it can be handled at ease in the next step.

The links are sorted based on the link weights, in order of decreasing weight.

A fraction of the sorted array of links is taken, and the others discarded. The fraction, determined by a parameter known as density, is chosen so as to simplify the graph, while ensuring little information is lost.

The resulting links are then used to reconstruct the simplified graph.

The effectiveness of this step is dependent on the density parameter.

5.2.3.3 Link Merging

A very low density would result in a simplified graph, but a great deal of information is lost while dropping a lot of links. Thus the density parameter cannot be used as the only graph simplification technique. A better procedure is to simplify the graph using Merging techniques. The process of merging links is dominated by two questions: when should links be merged, and how should the merge occur?

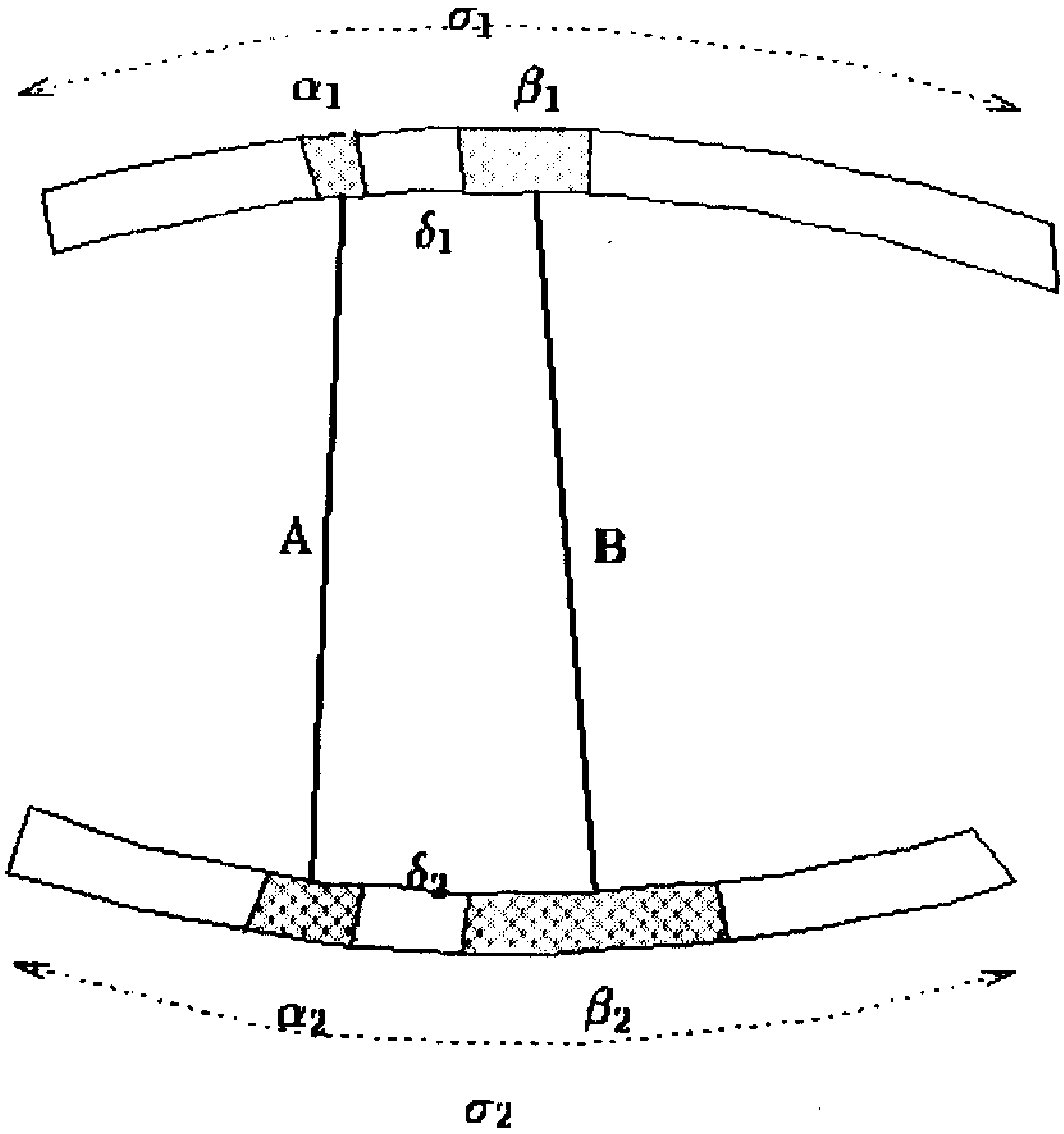
The answer for the first question is to divide the link pairs into four categories

1. Parallel Links: This is difficult to define when the documents are of different sizes. A pair of links, which do not cross each other and has almost the same proportion of unlinked text between their end-points, can be regarded as a parallel pair.
2. Askew pair: This is one which does not cross, but which is not parallel.
3. Wedge pair: This is one in which one node participates in both the links.
4. Cross pair: This is one in which the two links cross each other.

Any type of pair relationship could be useful for merging, but the distance between the nodes is of obvious importance.

Whether or not pair A and B should be merged depends upon the size of

$$Dist = \sum_{i=1}^{2} \frac{\delta_i}{\sigma_i}$$



That is, the distance between links is related to the proportion of the total size of the documents that lies between their endpoints. The link pair with the smallest distance will be merged first; considering the relationship between the links breaks ties. (A pair of links in a wedge relationship is always considered closer together than the similar askew pair with adjacent endpoints.) After a link pair has been merged, the new link will have a different distance from the remaining links. The process continues until no links are close enough for merging. When two links are merged, the resulting meta-link is between different passages of text and may even include some previously unlinked text. The similarity assigned to that edge must therefore change. It is possible to re-compute the similarity by recalculating the vectors for the merged endpoints and computing the similarity directly, but it is possible to achieve a similar effect by appropriately combining the similarities of the merged links. The new similarity must satisfy some simple properties in order to be useful. For example, if previously unlinked text is merged into the meta-link, the new similarity will necessarily be lower than the original. Appendix A of [Allan95] describes properties expected of a new similarity, presents the formula used to calculate the new value, and proves that the formula satisfies those properties.

5.2.3.4 Stopping the merge

The link merging process requires some means for knowing when merging should stop. Some possible options are:

- Merging all links between parts of the two documents.
- Merging until no remaining links are above a threshold (every merged meta-link has a similarity lower than the links used to create it); or,
- Merging until the distance between the closest link pair is larger than some threshold.

The final choice of merge termination was used in the work.

5.3 Document Relationship Visualization

The relationship between documents must be visualised before finding the features marking those relationships. There are various Visualization Techniques [Allan95]. Since this dissertation deals with documents pair-wise, the following technique suffices.

Procedure

Two parallel strips, with each representing a document represent document pairs.

The length of the strips is in proportion to the size of the documents they represent.

Only those links are taken that satisfy a threshold or density property.

The parts of documents that participate in links are shaded at their respective positions in the document, in proportion to their size.

The links are then drawn as lines between the parts of documents (shaded portions of the parallel strips).

5.4 Graph Feature

The following features were formulated for finding the relationship between documents. The features extracted from the graph can be used to match the patterns, thereby inferring the relationship.

- ***Relative Size***: The relative sizes of the two documents.
- ***Slope***- The extent to which links are parallel is measured by this feature. The formula used to calculate the slope is

$$slope_i = sim_i \times \left(\frac{part\_no\_\max_i - (part\_no\_\min_i + (\max\_part\_no - \min\_part\_no)/2)}{\max\_part\_no} \right)$$

Where *slopei* and *simi* are the slope and similarity of the ith link. *Part_no_maxi* and *part_no_mini* are the node numbers of the links on the document with larger number of parts and smaller number of parts respectively. *Max_part_no* (*min_part_no*) is the number of parts in a document with more (less) parts in the document pair.

This mainly helps in identifying the revision, part-of-a-document and duplicate document link types. It is calculated before merging the graph.

- ***Mean Slope***- mean slope of all the links.
- ***Number of Cross Links***- The number of links that cross each other. This feature helps in identifying summary and specialization kind of links.

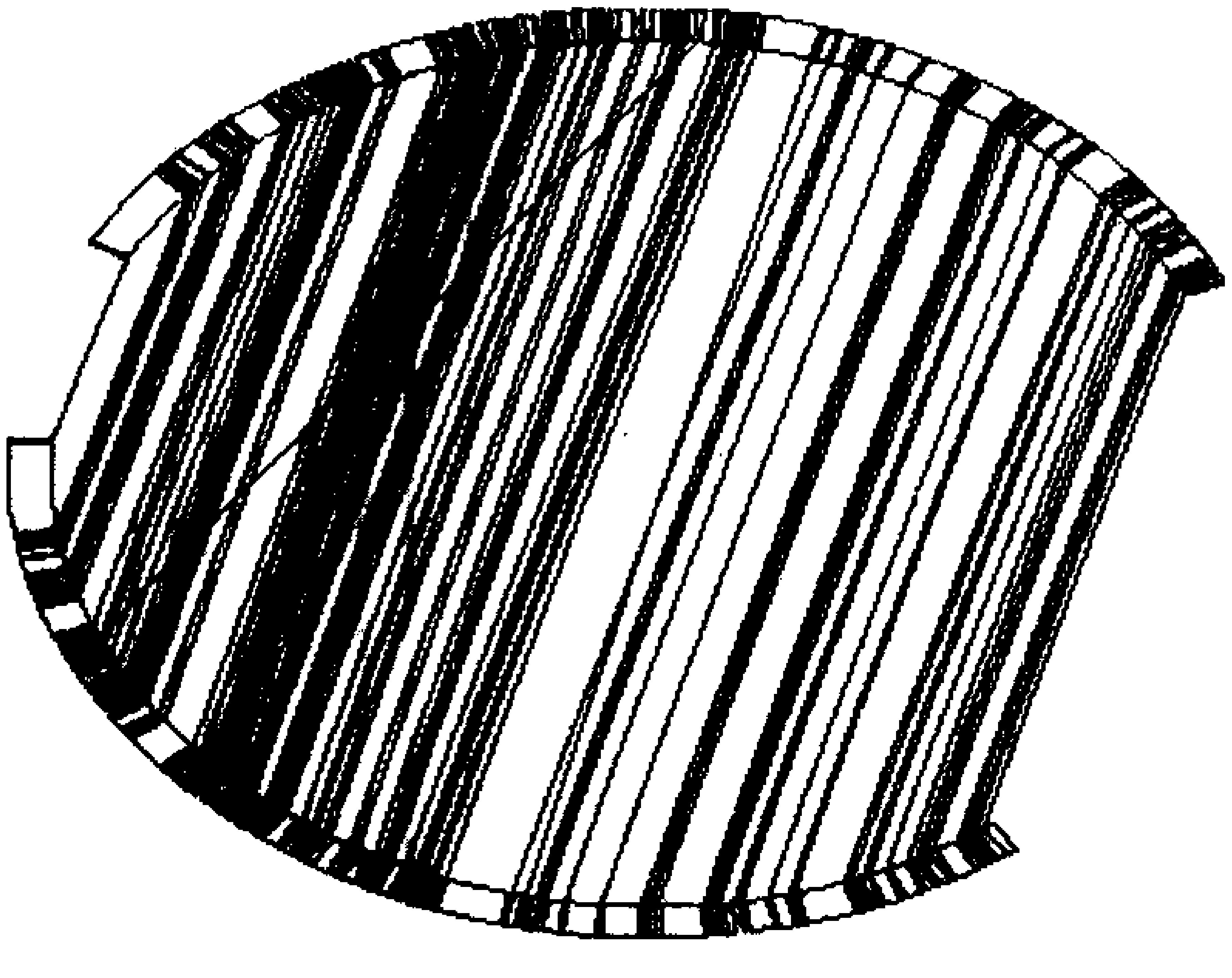
This feature is computed after the merge.

- ***Size of Unlinked text***- The bytes of unlinked text between the leftmost linked part and the rightmost. This feature helps in identifying summary and specialization. This feature is computed after merging and found for both documents.
- ***Slope-Deviation***- This feature uses the slope for measuring the standard deviation of a link's slope with the mean slope.

5.5 Graph Patterns

Some of the patterns exhibited by the relationship graph between documents under each type and the feature to be extracted from it are discussed below.

5.5.1 Revision pattern



Brief description:

The links in the document tend to remain parallel. Any cross link(a link that crosses many links) is expected to have a lower similarity.

During revision the sections remain almost the same with few additions here and there. This supports the claim that links tend to remain parallel. Since most of the sections are repeated in the revised document, all repeated sections are linked, thus resulting in very few links after merging.

Expected features:

The number of links will be very small after merging. The standard deviation of slope will be very small.

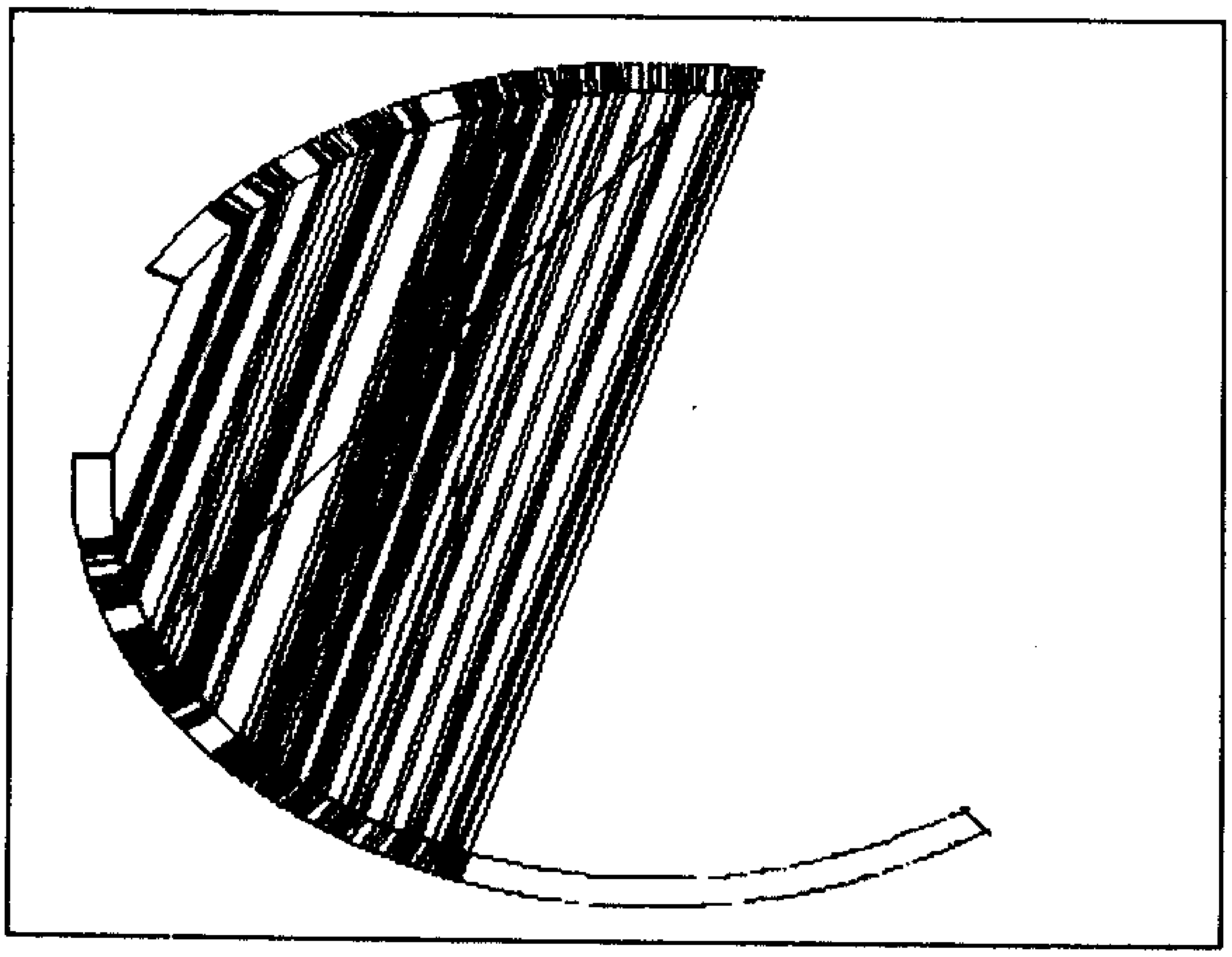
Chosen Threshold: No- of links after merging less than 3, slope deviation must be less than 0.02 and relative size greater than 70

5.5.2 Duplicate pattern

Brief description:

This is a constrained version of revision. In addition to the features of a revision, document's relative size with the duplicate will be near to 1.

5.5.3 Part-of-a-Document



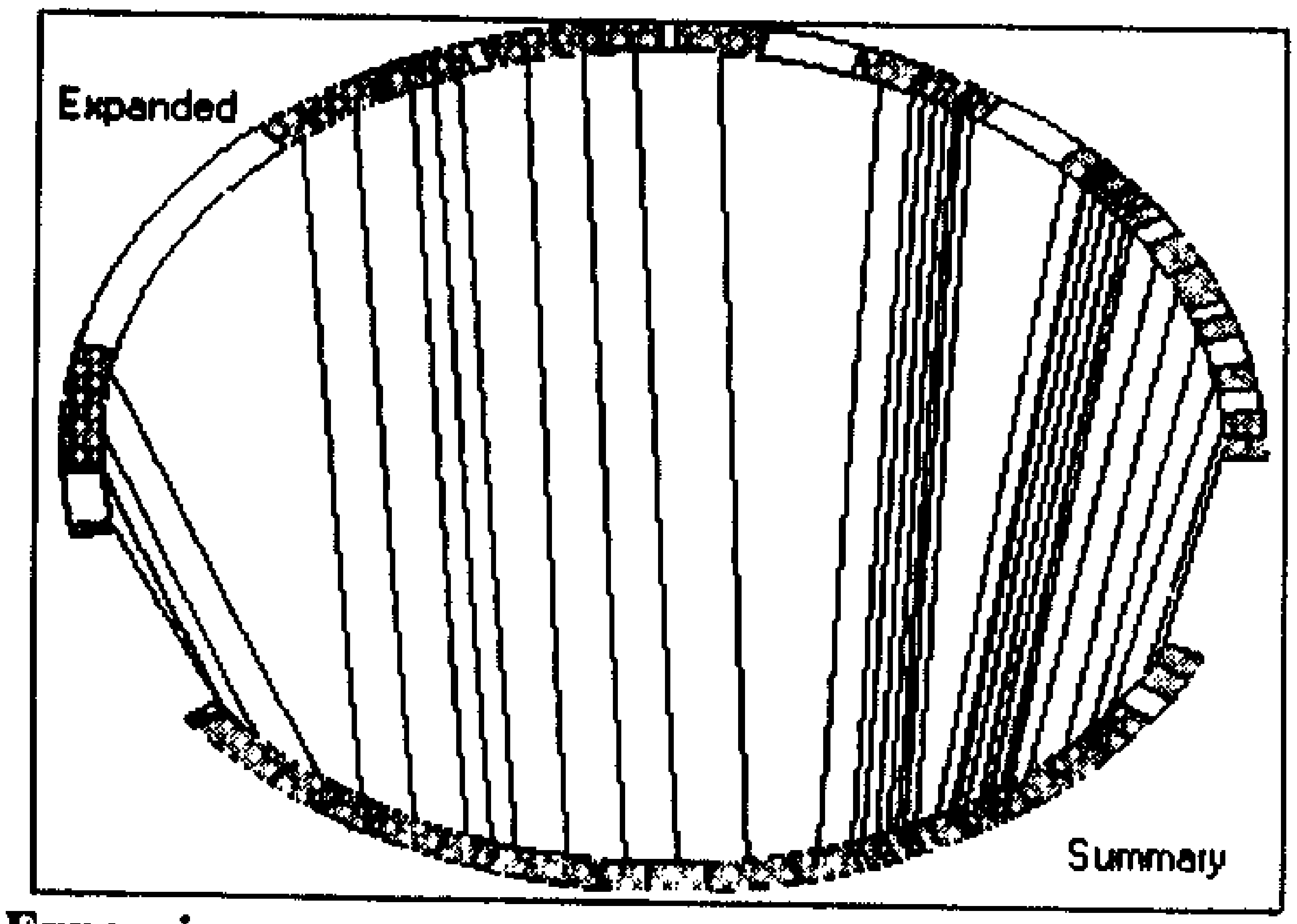
Brief Description:

A separate part of the document behaves in the same way as the duplicate link type, with the same part of the full document.

Extracted Features:

The slope standard deviation is small. The size of the linked text on both documents must be almost same after merging.

Chosen Thresholds: The size of one document must be less than 60%of the other, the slope deviation must be less than 0.02.



5.5.4 Summary and Expansion

Brief Description:

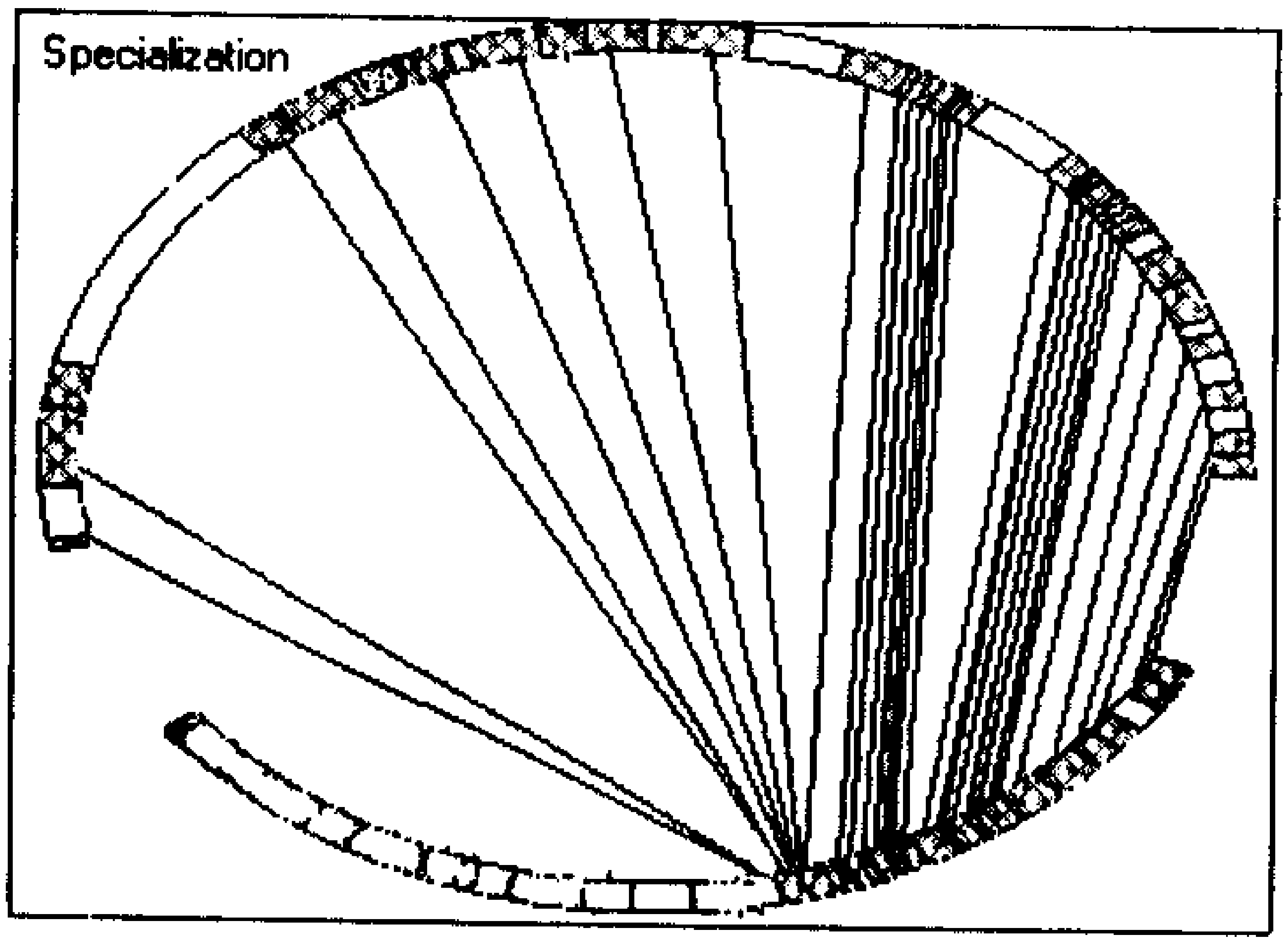
Summary deals with the subject in the same order as the detailed document. Thus the summary – expansion inverse relationship exhibits something similar to the graph presented above. The summary is expected to be much smaller than the expansion document. The amount of unlinked text if exists, is more on the detailed side than on the summary side.

Extracted Features:

The relative size of summary is much less than the detailed document. The number of cross-links in the merged graph is minimal. The linked portion of text on the summary forms a large part of its text, the portion of the document between the first linked part and the last link part forms a significant part of the text.

Chosen Thresholds: The size of one document must be less than 60% of the other. The smaller document must have less unlinked text than the larger document. The linked portion between the first and the last linked part must cover more than 90% of the smaller document and 80% of the other.

5.5.5 Specialization



Brief Description:

Specialization deals with a particular topic of a general document in greater detail. Thus the amount of linked text on the general document side is smaller than that in the specialised document.

Extracted Features:

The number of cross-links is minimal. The Linked size of both sides is calculated after merging. The linked size must be a small fraction of a general document, whereas a large fraction in the specialised one. After merging, one of the linked portions of the text on the general document must be related to most part of the other document.

Chosen Thresholds: If a part in a general document with size less than 60% of the whole document is related to a document by covering more than 80% of it after merging, then the latter is a specialization of the former.

5.5.6 Equivalent Document

Brief Description:

The equivalent document deals with the same subject. Since the order of presentation depends on the author of the document, equivalent document can be visualised as the same document with permuted sections.

Extracted Features:

Since the equivalent document deals with the same subject, most of the sections are supposed to be linked. Thus unlinked text is minimal. The standard deviation of slope can be quite high. As per [Allan95] equivalence means two strongly related documents. According to the merging algorithm, this should result in a minimal number of links (because strongly connected graphs merge with minimal links left) after merging. A document obtained by randomly permuting the sections of a given document is an equivalent document, but the graph for such a document pair was often found to contain a large number of links after merging. As the supposition that random permutation of the sections in the same document represents an equivalent document seems reasonable, the pattern for an Equivalence relation suggested in [Allan95] may not end up in finding such a relation.

5.6 Experiments

The algorithms were written as plug-ins to the Smart System. The documents having such relationships were collected and studied. Some of the results are presented here.

Revised RFC 903

A Reverse Address Resolution Protocol**
(RFC-903)

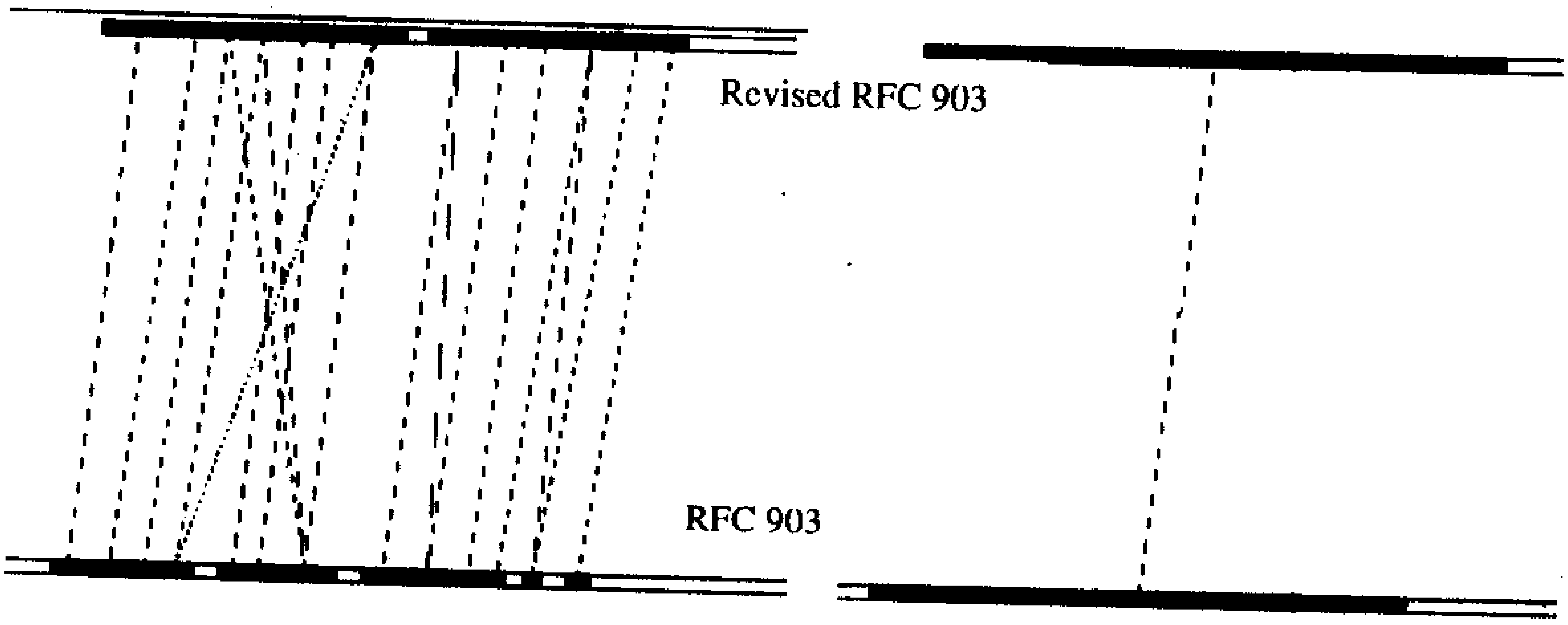
ABSTRACT

This RFC suggests a method for workstations to dynamically find their protocol address (e.g., their Internet Address), ...requests discussion and suggestions for improvements.

---

1. Introduction
2. Design Considerations
3. The Proposed Protocol

Graphs



Before merging

After merging

Original RFC 903

A Reverse Address Resolution Protocol
Ross Finlayson, Timothy Mann, Jeffrey Mogul, Marvin Theimer Computer Science Department
Stanford University June 1984
Status of this Memo

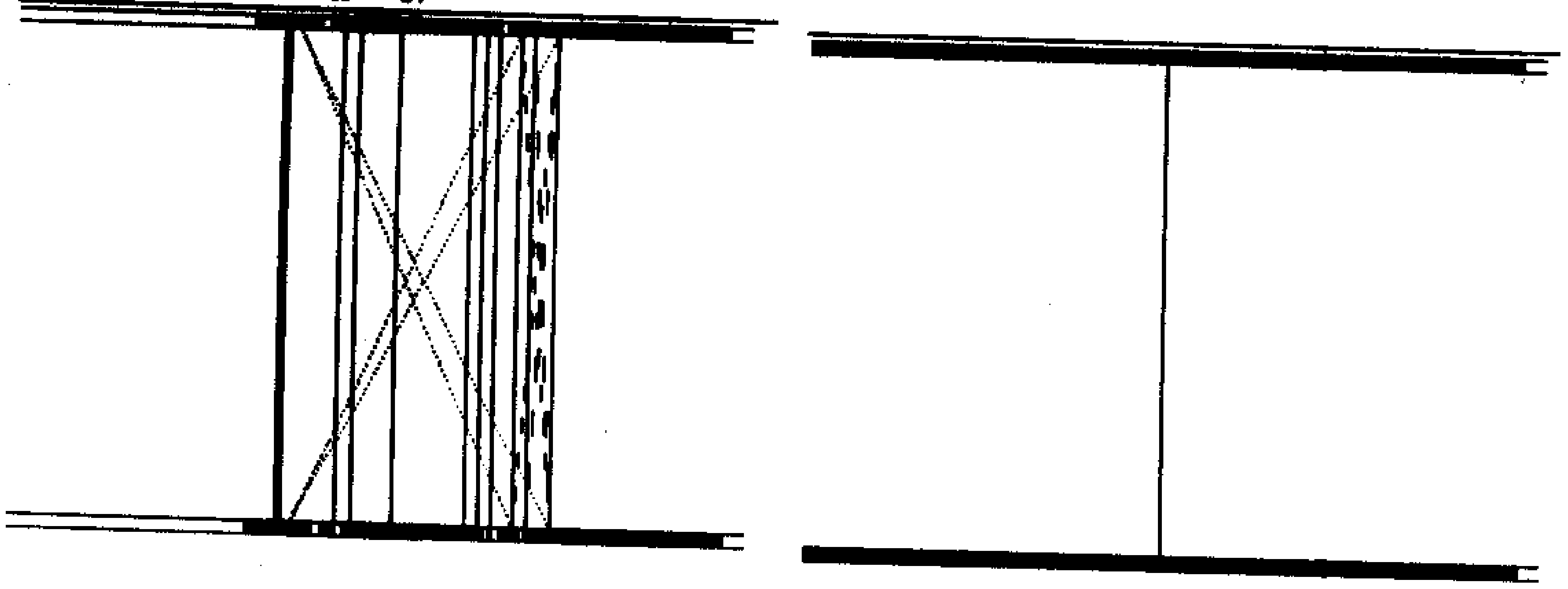
This RFC suggests a method for workstations to dynamically find their address).

Graph features,

Relative size = 78%, Slope deviation = 0.016, No of links after merging = 1.

Duplicate Document

The graph features, Relative sizes =0.0, Slope deviation = 0.028, Number of links after merging = 1, Number of crosslinks = 0.

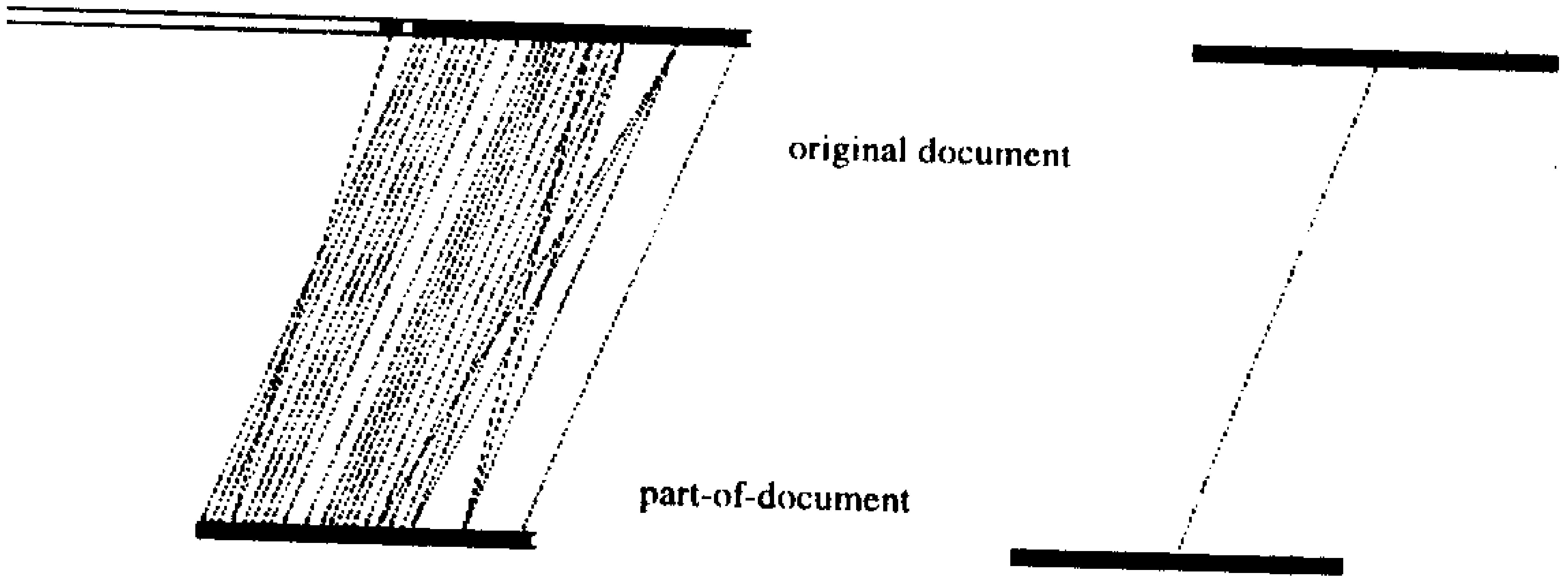


Before merge

After merge

Part-of-a-document

A document on Software architecture was broken and then tested.
Graph features:
Relative Size 46%, Slope Deviation 0.015, No.of cross links = 0.



Before merge

After merge

Summary and Expansion:

Example 1.

The document was downloaded from www.the-hindu.com. The documents have a News-in-Brief and News-in-Detail relationship, similar to a summary-expansion relationship.

Document description.

News-in-Brief

The front page of the paper had briefings on the following,

TV station attacked in Fiji capital A new wave of political unrest swept across the Fijian capital of Suva tonight, ...

Only the odd apple is rotten: Gavaskar Batting legend Sunil Gavaskar today said, ``There may be an odd rotten apple ......

I've done my job, says Prabhakar Young World Former Test cricketer, Manoj Prabhakar, today said he had done his job of exposing the match-fixing racket ......

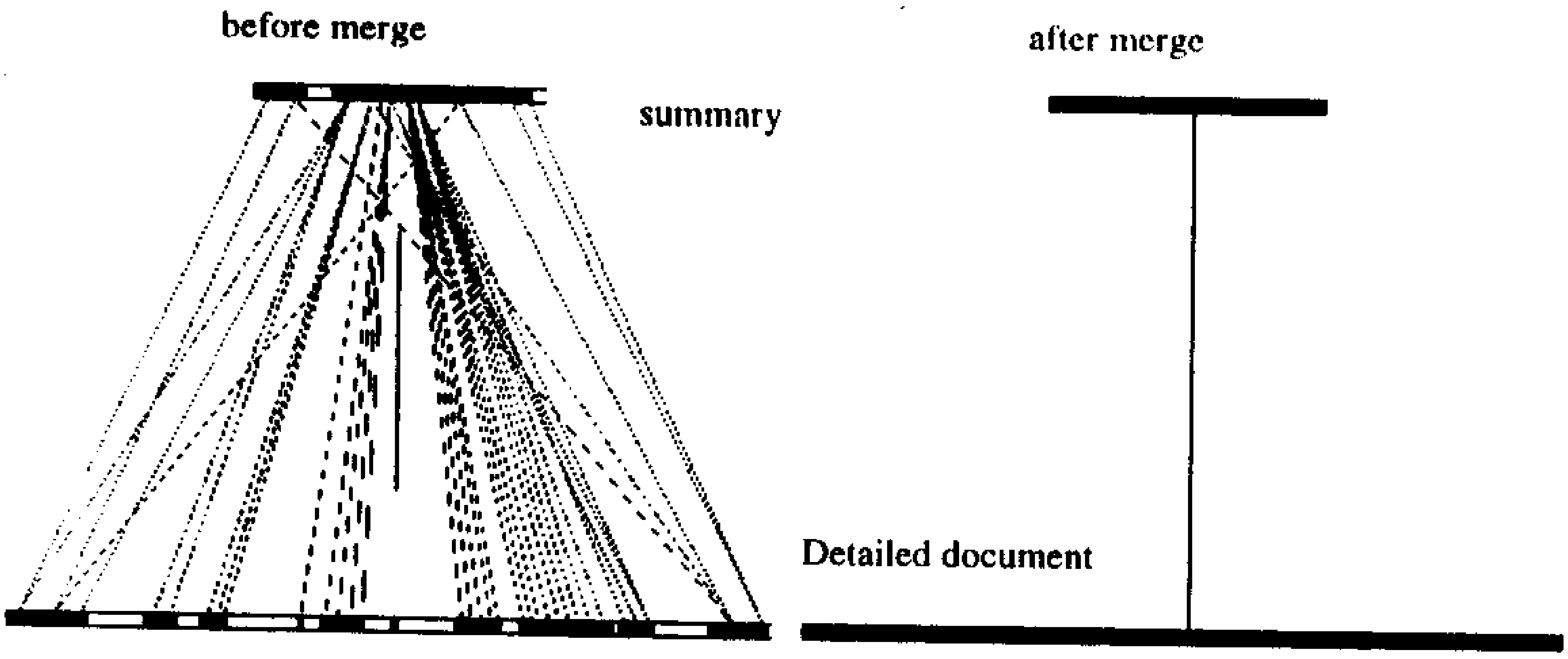
Malicious, says Sinha The Union Finance Minister, Mr. Yashwant Sinha, today warned his detractors who have been attributing personal motives to him for allowing major foreign companies to .Jaffna remains quiet A day after the unilateral 12-hour ceasefire of the Liberation Tigers of Tamil Eelam (LTTE), the Jaffna peninsula remained largely quiet, with only sporadic clashes reported in the Narayanan looks forward to talks with Jiang Hoping to impart a sense of trust and confidence to Sino-Indian relations,.....

Disinvestment panel soon The Disinvestment Minister, Mr. Arun Jaitley, said a new Disinvestment Commission would be.....

News in detail had the above mentioned news in detail.

The following results were obtained

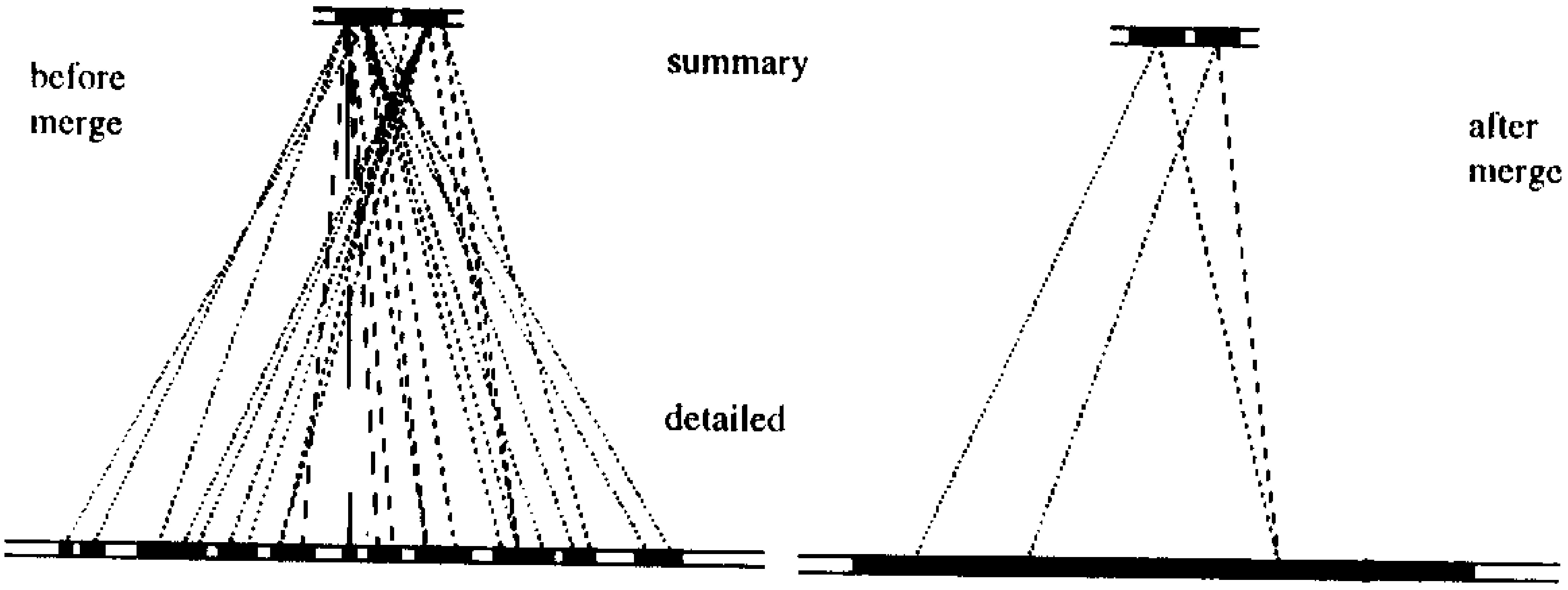
Relative size 47%. Linked text on brief document 100%, Linked text on Detail document 99%



Example 2.

This document-pair discussed regarding Decision Making.

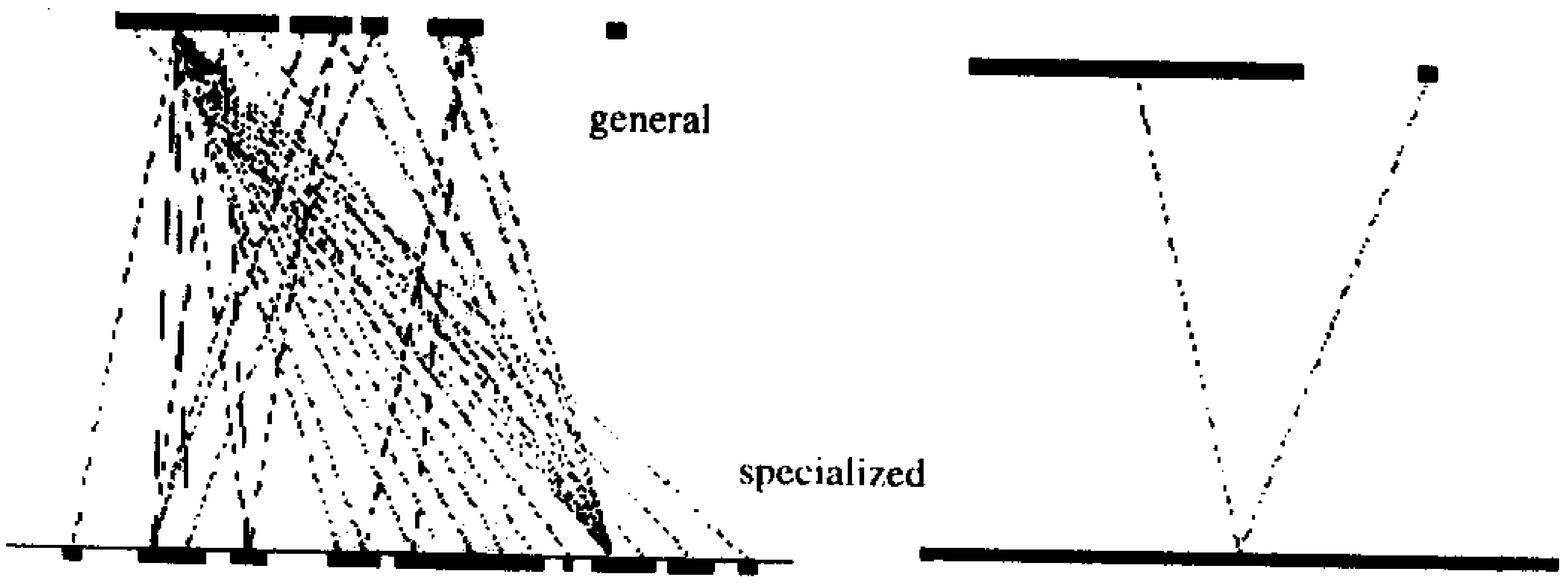
The topics of discussion were Predictive deduction, Bayesian Deliberation, Creative Decision making. In the detailed document the initial part discussed on the 'free will' and 'predictive deduction'. The second part explained in detail the Bayesian Deliberation, the third part described 'Creative Decision Making', the last part on Frankenstein's Dilemma.



Specialization:

Example 1:

The document pair discussed Pest Management. One of the documents discussed generally on the use of pesticides for pest management. The document briefly tells about the dangers associated with pest management. The other document discussed in detail the dangers associated with pesticides.



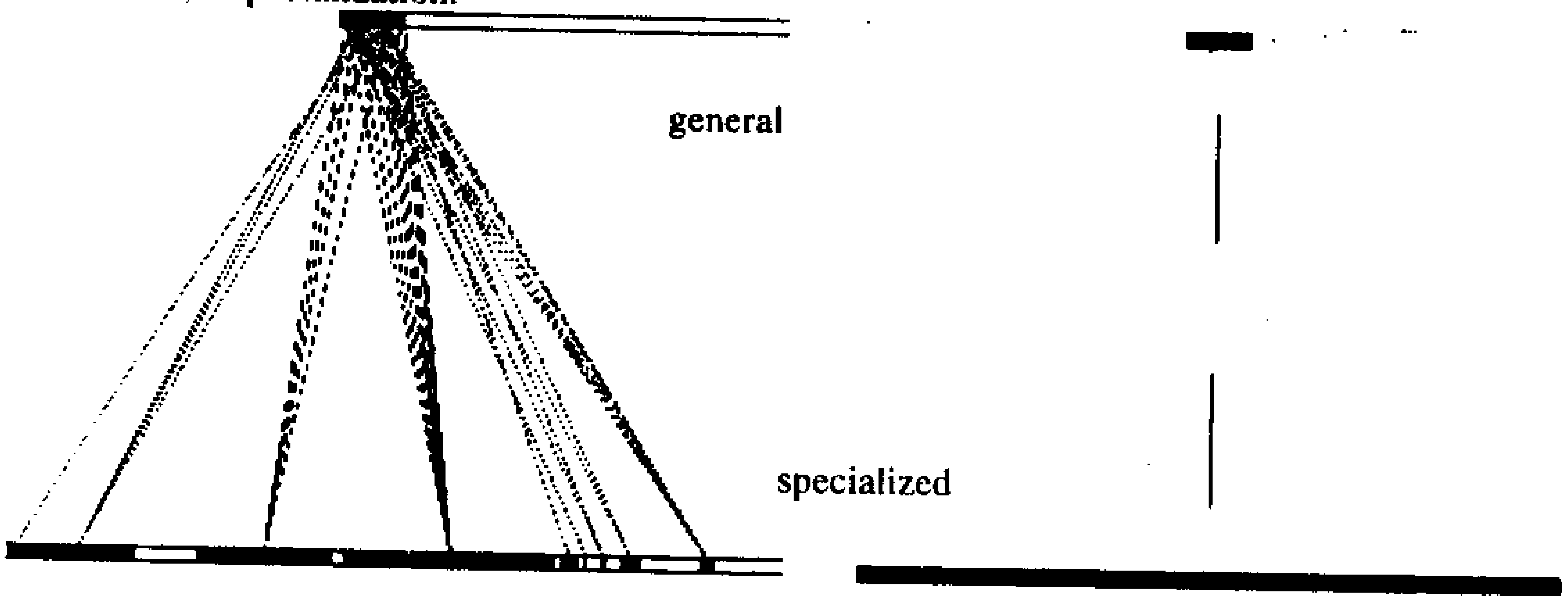
before merge

after merge

The relative size = 0.7, Had a part in document 1 which was 54%of the document size, and the part was related to 94% of the document2.

Example2.

The documents discussed on the Unmanned space missions. One of the document discussed on many space missions, mentioned briefly regarding Pioneer10. The second document gave a detailed account of Pioneer 10, a specialization.



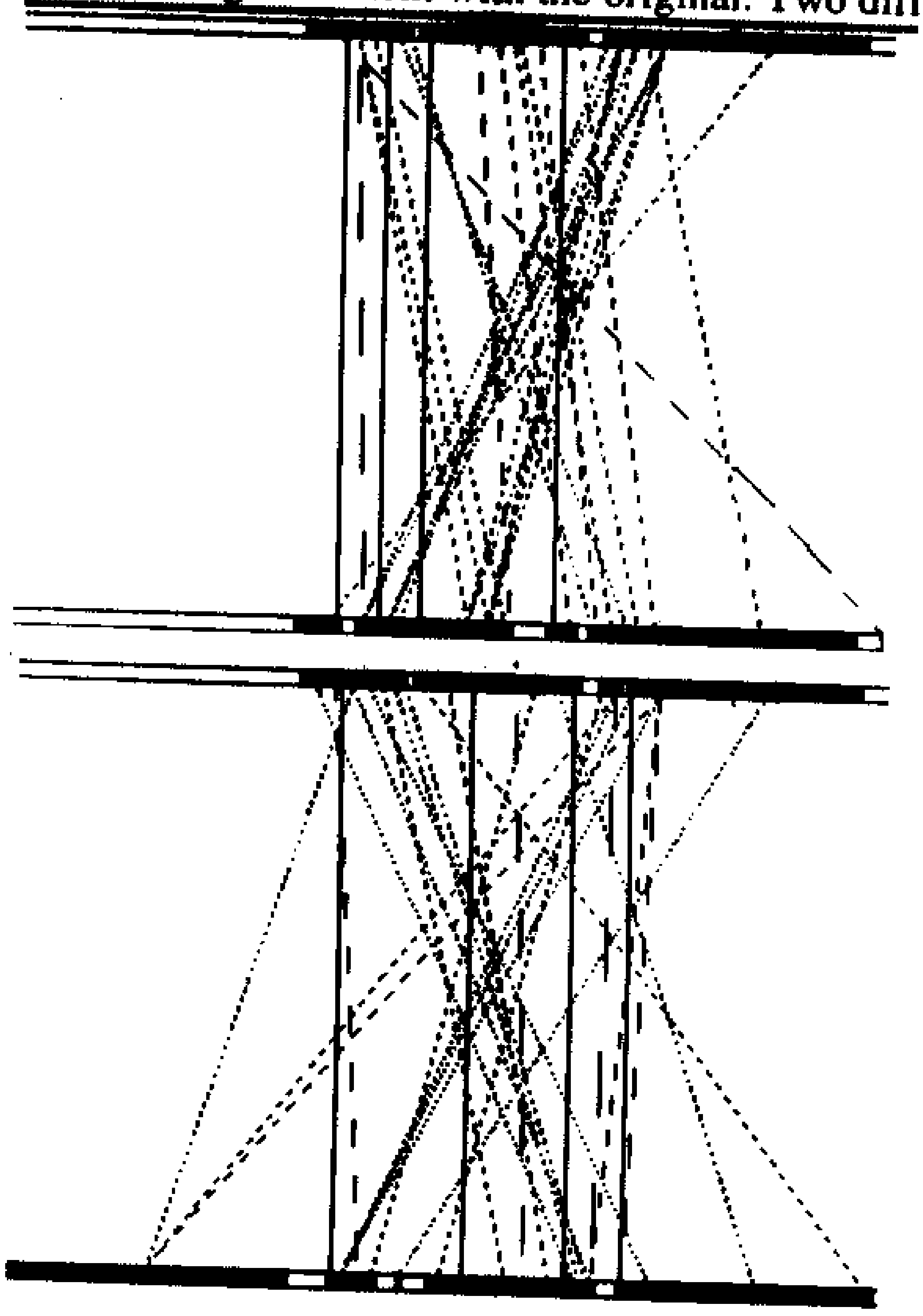
before merge

after merge

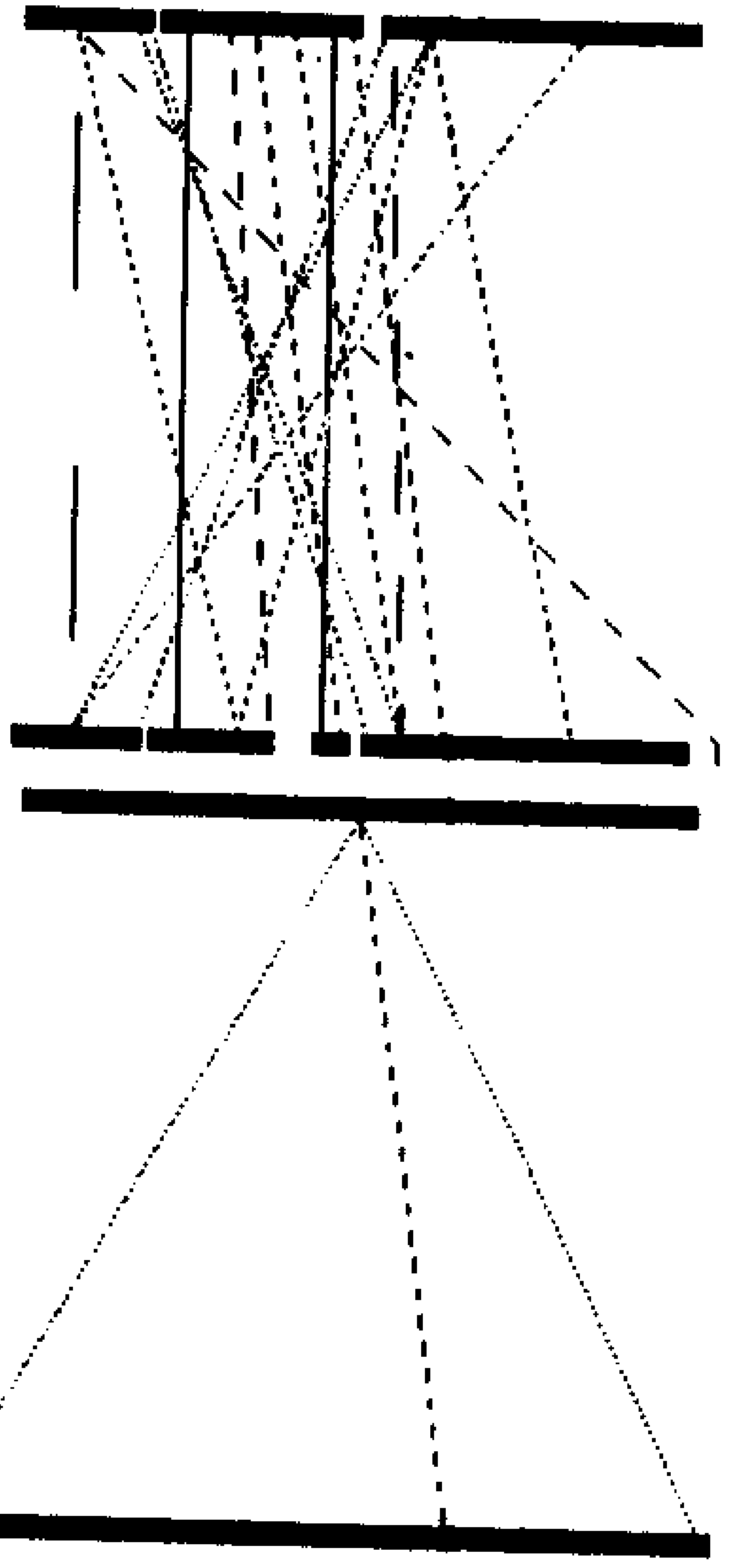
The relative sizes 34%, A part in document1 with 30% size was related to 84% of the second document.

Equivalent document:

. Since an equivalent document may discuss the same topic but not necessarily in the same section order, the equivalent documents relationship was tested by permuting the sections in a document and comparing the resulting document with the original. Two different permutations yielded different merged results.



before merge



after merge

The equivalent documents were identified as, documents with more than 80% of text in each part related to each other after merging and does not belong to any other class.

6 Contributions and Future work

The contributions from this work are in Text Categorization and Document Link Typing. The merging techniques and patterns were mentioned in [AllanJ]
Graph Feature Extraction method was formulated in this work. Some new Link types have also been proposed in addition to the ones mentioned in [AllanJ], for example, Specialization, Document part and Duplicates. Their patterns were also formed and tested with the features suggested.
Until recently, most of the work in the area of Text Categorization dealt with categorization within a simple (non-hierarchical) set of categories. This work explores the possibilities of using the Vector Space Model for categorization into a hierarchical set of classes.
The Future work in Categorization can focus on the following issues

1. Reasons for the failure of the algorithm in certain Categories can be explored and accordingly modifications made to improve the accuracy score of the system.
2. The work can be extended by not having a fixed hierarchy during categorization i.e., if existing categories do not fully characterize a new document then new categories can be introduced, while adding documents to the category hierarchy.
3. The evaluation score can be reformulated so that the number of categories returned is taken into account. (At present a very bad categorization process can return all the Categories and get a very good score, which is not correct).
4. The HyperLink relationships can be exploited while categorizing hyper-text documents.
5. Proper weighting scheme for easy discrimination of categories at each level of categorization can be formulated.

7 Appendix

7.1 Document Collection

A spider that crawls the World Wide Web was written to collect documents. A spider is designed based on the application. The documents required in this work were from the same Web-Site. Thus a spider was designed to crawl HTML pages from the same web-site. The main issues regarding the design of a spider customised for this application is that

1. It must keep track of the pages visited.
2. It must keep a list of pages to be visited
3. It must handle exceptions in networks, like time out, unknown host etc.,
4. It must have checkpoints so that it can start from where it stopped

Each time a new link is added to the list of "to be visited links", we must check whether it is in the history list. If space and time considerations are kept in mind while handling the issues, the performance of the spider can be improved.

Since the documents were from the same web-site, the data structure used was a tree simulating the URL directory structure of the web-site.

Each node in the tree consisted of a hash table and the list of names of child directories or files of the same directory. The data structure was very time efficient, as finding whether the URL is visited or yet-to-be-visited takes at-most the height of the tree. Thus a high improvement in performance was obtained, for handling the first and second issues. If each node had sufficient number of children, then an improvement in space also could be obtained. (Redundant Path string of files in the same directory must be stored separately, whereas in the data structure used, only the file name is added). However this data structure fails if documents to be collected are not from the same web-site.

8 Bibliography

1. [AllanJ] "Automatic Hypertext Construction", Phd thesis by James Allan at Cornell University in Feb 1995.
2. [AmitS] "Term Weighting Revisited" Phd Thesis by Amit Singhal at Cornell University.
3. [AlSkM] "Hierarchical Text Categorization" by Stephen D'Alessio, Aaron Kershenbaum, Keitha Murray and Robert Schiaffino.
4. [VANRBN] "Information Retrieval" an on-line book, by Van Rijsbergen
5. [FraYat] "Information Retrieval, Data Structures & Algorithms", by W.B.Frakes and R.B.Yates