On the Destination of Retrieval Links in Hypertext Systems

Dusan Cakmakov

Faculty of Mechanical Eng., University of Skopje, Karpos II bb, Skopje, Macedonia

Information browsing in hypertext systems is provided by links with source and destination anchors within hypertext documents. From the point of view of link following operations, links can be static or dynamic. While for static links, source and destination are well defined before the link is used, dynamic links have no defined source and/or destination until the links are generated.

In this paper, we focus on the so-called retrieval links which use information retrieval technology to enable dynamic connections between hypertext nodes. We address the important problem of automatic determination of optimal destinations for the retrieval links within hypertext documents. A new approach based on the density distribution of important term weights within destination documents is described. The power of such a similarity measure is in local text comparison where we do not dispose of the real term weights. The implementation algorithm and characteristic examples are also presented.

Keywords: information retrieval, text or document passage, hypertext, hypermedia, link destination

1. Introduction

A number of approaches for incorporating information retrieval (IR) strategies in hypertext systems have been proposed (Frisse and Cousins, 1988), (Croft and Turtle, 1989), (Frei and Stieger 1995), (Agosti et al., 1997), (Allen, 1997). IR offers many tools for creation of content-based links in hypertext systems. Thus, the retrieval links are dynamic content-based links with source and destination in hypertext documents (DeRose, 1989). They usually use well-established concepts of text document indexing and similarity estimation between the link source and the link destinations. While the retrieval link sources can be easily made narrower by selecting an appropriate part in the current document, automatic determination of the retrieval link destination within documents is not a straightforward task. The main problem is the absence of information about the content of arbitrary part of a document.

The typical implementation of retrieval links is as follows. The hypertext documents are indexed by extracting content meaningful terms which are presented as word stems in the document indexes. Each stem has an associated weight factor computed using, for example, the inverse document frequency method (Salton, 1989). The source of a retrieval link is usually part of the current document. The user can select any document area to generate retrieval links. The system will extract useful terms from the selected area and create a query composed of stems of selected terms and calculated weight factors. Then, the similarity between the query and the document indexes is calculated, and the documents are ranked according to the level of similarity. The most similar documents are presented to the user for link following operations. In this way, retrieval links offer destinations that are whole documents.

The described approach imposes a natural problem: how to automatically find optimal retrieval link destinations within hypertext documents? In other words, how to direct the user to the most significant part of the document enabling them to get the relevant information immediately? This problem is in the spirit of hypertext systems, where because of information overload the user should be more directly led to the right information. Recently, it has been shown elsewhere that such link locations can improve the quality of retrieval process (Salton, 1993b) which in hypertext systems with whole documents as link destinations has been shown to be relatively weak (Savoy, 1993).

Recently, the problem of finding an appropriate way to link document parts has been discussed elsewhere (Bernstein, 1989)(Salton et al., 1993a), (Chang, 1993), (Salton et al., 1996), (Kaszkiel and Zobel, 1997). Most of the approaches are based on text decomposition in pieces, typically paragraphs, and on the calculation of the similarity between the query and pieces using the same method as for whole documents. It is clear that such an approach is static with fixed document parts as link destinations and does not offer a real solution when the optimal document part does not correspond to a fixed text object.

According to (Salton and Allan, 1993b), the typical retrieval scenario is performed in two steps. At the first step, a global text comparison is performed to reduce the number of documents where retrieval link destination can be placed. For this purpose a standard vector model can be used. At the second step, a local text comparison is used to locate those document parts that best satisfy the query. To perform this step, the indexes for specific document parts, typically paragraphs, have to be kept or a direct comparison between the query and specific document parts has to be made. In many cases, a paragraph or a sentence could be an acceptable link destination because they are natural text units. However, it is not difficult to imagine cases when two or more adjacent sentences, paragraphs or a text part that starts somewhere in the paragraph and ends somewhere in the next paragraph can be the best destination. Previous approaches avoid to consider this problem because it is impractical to keep indexes or to compare the query with all document parts. So, previous approaches make compromise and locate only natural text units, paragraphs or sentences. Why not offer a more general solution, especially if it is possible to do in an elegant and efficient way, without loosing in system performances?

In this paper, we propose an efficient technique for automatic determination of arbitrary retrieval link destinations within hypertext documents. The technique is based on density estimation of important term weights in the destination document. The important terms are obtained in the process of similarity estimation between the query and the document indexes. Then, the density function for important term weights within the destination document is calculated. Using the density function we locate the offset interval with maximal density of important term weights. Finally, we expand the found offset interval in the appropriate document part which will be the optimal place for retrieval link destination.

The term densities was used by some authors to locate document passages in an indirect way. Thus, (Callan, 1994) uses overlapping text windows of fixed length and (Buckley et al., 1995) use proper normalisation of the similarity measure by the passage length.

However, our technique has several advantages:

- It is able to locate an arbitrary document part as a retrieval link destination;
- It can dynamically control the destination size;
- It does not process separately any document part.

To achieve the necessary information for density estimation we use document indexes in an inverted file, where as well as term weight factors, we keep the term locations. That is a drawback of our technique because including term-location information in document indexes expands the size of the indexes, possibly by a factor of three, since each occurrence of a term in the text should be registered (Salton, 1989). Additionally, the index structure becomes more sensitive to document changes. In the cases of document decomposition in paragraphs and sentences the index is burdened by information about the terms in the corresponding document pieces and the problems with document changing are also more expressive. In hypertext and other strongly interactive systems, the performances are essential. The problem of document changing is a general one, not only related to retrieval links in hypteerxt systems or to the locating of the optimal document parts. The changes in a set of documents or in document contents lead to rebuilding of the index structure.

The index with term locations has another advantage since it provides fast implementation of non content-based links where the user wants to follow links from a selected term or phrase toward nodes with the same term or phrase, because we can directly find the term or phrase locations in the destination document.

2. Retrieval Link Destinations

Before describing our technique, we assume that the document indexes are expanded with the offset addresses of the index terms. It is a trivial task which should be performed in the process of document indexing.

This index will be used to reduce number of documents where destination could be placed (global level) and then to locate the best place for retrieval link destination within the document (local level). There are two logical alternatives of this approach. The first alternative is to use full text match through the whole set of documents. In this case, the problem of document changing is solved at heavy price of system performances. In hypertext and other strongly interactive systems the performances are essential. The second alternative is to use an index without offset addresses on global level and then perform full text match on local level. In this case, it is unsatisfactory from the performance point of view to match the query with all document parts. Additionally, it is not desirable to process documents directly when the document indexes already exist.

Finding optimal destinations for retrieval links within documents is performed in three steps.

At the first step, we identify important terms within the destination document and the corresponding weights. Informally speaking, we extract terms which have significant participation in the similarity score between the query and the document index. Let Q be a query composed of selected terms from the link source (usually the index terms from the selected area in the source document)

$$Q = \{(t_{q_1}, w_{q_1}), (t_{q_2}, w_{q_2}), \ldots, (t_{q_m}, w_{q_m})\}$$

and let I be a document index

$$I = \{(t_{i_1}, w_{i_1}), (t_{i_2}, w_{i_2}), \ldots, (t_{i_n}, w_{i_n})\}$$

where (t_{q_k}, w_{q_k}) and (t_{i_k}, w_{i_k}) stand for the term and the associated weight in the query and the document index respectively.

Then, the set T of important terms will be

$$T = \{(t_1, w_1), (t_2, w_2), \ldots, (t_r, w_r)\}$$

where

$$t_k \in \{t_{q_1}, t_{q_2}, \ldots, t_{q_m}\} \cap \{t_{i_1}, t_{i_2}, \ldots, t_{i_n}\}$$
$$w_k \in \{w_{q_1}, w_{q_2}, \ldots, w_{q_m}\}$$

such that for

$$t_k = t_{q_k} = t_{i_k}, w_k = w_{q_k} \quad \text{and} \quad w_{q_k} \cdot w_{i_k} \geq val$$

for some threshold value val. The important term weights are taken from the query (terms from the selected document part) because they represent the real user needs. In some cases, it is useful to use $val = 0$ and include all selected terms from the query which are in the document index, especially if the query vector contains a small numbers of terms.

So, after the step 1, we have the set O of important term offsets s_k within the destination document and corresponding weights w_{s_k} taken from the query

$$O = \{(s_1, w_{s_1}), (s_2, w_{s_2}), \ldots, (s_l, w_{s_l})\}.$$

Let us note that $l \geq r$ since each term can have more locations in the destination document and $w_{s_k} \in \{w_1, w_2, \ldots, w_k\}$.

At the second step, we estimate the density of important term weights within the destination document and locate the area (offset interval) with the greatest density.

Density estimates are all based upon the Lebesgue density theorem (Devroye, 1985), but for practical purposes we need an approximated version: the density estimate $d(x, h)$ of a random vector $(x_1, x_2, \ldots, x_n)$ is given by

$$d(x, h) = \sum_{i=1}^{n} \frac{I_{[x_i \in S_{xh}]}}{n\lambda(S_{xh})}$$

where I is the indicator function, S_{xh} is the closed sphere of radius h centred at x and λ is the Lebesgue measure.

When h is small, the variance of $d(x, h)$ increases because fewer points are expected to fall in S_{xh}. We can additionally simplify $d(x, h)$ because we need not estimate density exactly but only compare the regions of concentration of the important term weights to find the most appropriate one. Thus, we should omit n, include important term weights, and knowing that the measure λ in our case is an ordinary interval length, we achieve the following formula

$$den(x, h) = \frac{1}{2h} \sum_{i=1}^{l} w_{s_i} \cdot I_{[s_i \in [x-h, x+h]]}$$

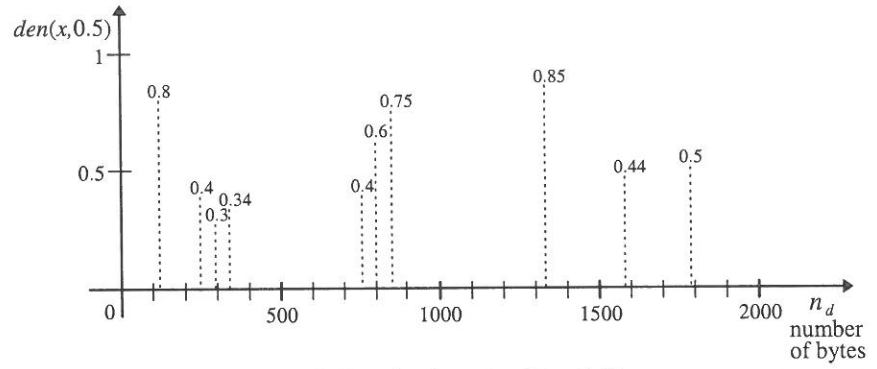


Fig. 1. Density function ($h = 0.5$)

where the values of x and h are given in bytes. Through the paper we shall use the term "density" for the above function, although it is apparently different from the real density function $d(x, h)$.

An example of a density function for important term weights in a document is given in Fig. 1. We assume that the term weights are normalised and belong to the $[0, 1]$ interval.

In this example $h = 0.5$ and the density is everywhere 0, except on important term offsets where it has jumps corresponding to the important term weights. n_d is the length of the document in bytes.

So, a locally estimated function is not useful for our purpose because we want to estimate the density of important term weights on some text intervals which will be an optimal retrieval link destination. The optimal destination could be a text interval whose length, for example, is about one sentence or one paragraph. If we want to offer a destination about one sentence long, the parameter h should be set with a value approximately one half of the average sentence length.

In Fig. 2 and Fig. 3, the density functions for $h = 50$ and $h = 250$ are presented. The first could correspond to sentence destination and the second to paragraph destination. Let us note that density is marked until 0.02 on Fig. 2 and until 0.01 on Fig. 3, because we use the number of bytes to represent the interval lengths which results in low density values.

As can be seen from Fig. 2, the optimal offset interval $[n_1, n_2]$ is located on the part of density function where it achieves maximum value. The determined offset interval is $2h$ bytes long with the middle in the point M, for which the maximum is found. Formally,

$$\forall x \in [0, n_d],\ den(x, h) \leq den(\mathrm{M}, h)$$
$$\text{and} \quad [n_1, n_2] = [\mathrm{M} - h, \mathrm{M} + h].$$

If the maximum is not unique, we discriminate between two cases.

In the first case, the maximum is repeated on a whole segment with adjacent points (frequent case) and we choose a maximal point somewhere in the segment.

In the second case, when the maximum is repeated after the density function has decreased

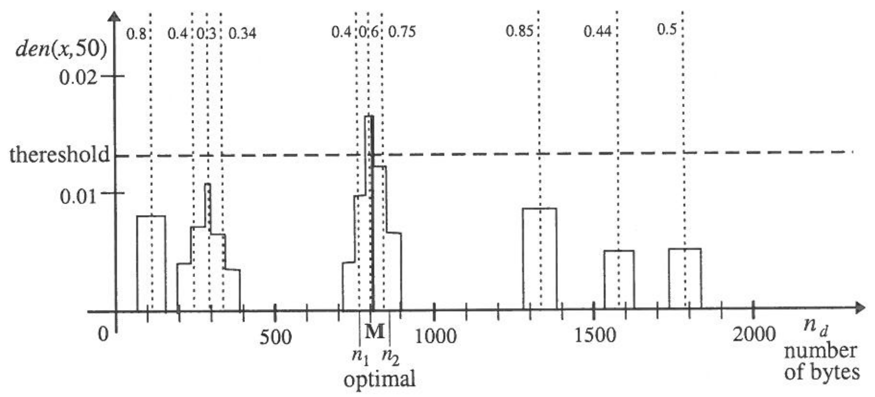


Fig. 2. Density function ($h = 50$)

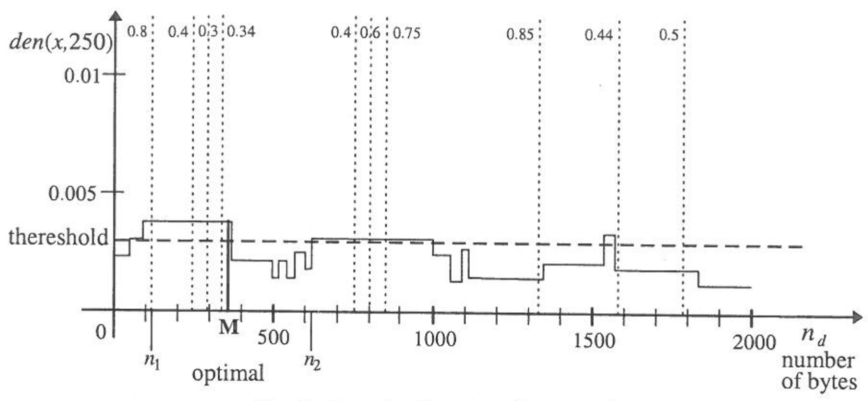


Fig. 3. Density function ($h = 250$)

(rare case), it is reasonable to assume that the first density maximum will produce the optimal offset. There are also many alternatives to unify the maximum in the second case using some additional criterion. For example, one can examine points around the maximum looking for greater values, or compare the offset interval width produced by the density function for the same predefined density value (threshold line on Fig. 2). The second approach can be used at the same time as a threshold for the estimation of the utility of the whole procedure. If the density function is lower than a suitable chosen threshold, there is no reason to look for an optimal destination and the whole document could be presented to the user.

The choice of this threshold value depends on h. For small h values, the density function achieves greater values and vice versa. A possible exponential choice for the threshold value could be $1/1.5h$ (see Fig. 2 and Fig. 3). This threshold, for example, corresponds to accumulated term weights of 1.3 in offset interval of 100 bytes which is a reasonable minimum. There are also possible more or less rigorous choices of the threshold value by using various decreasing by h functions.

The optimal destination interval in the case $h = 250$ is completely different. This is in concordance with our intuition, because on longer distances the first part of the document has higher density values (see Fig. 3).

As the choice of h determines the density function, it should be made in advance. A simple automatic approach is to determine h in such a way that the destination interval occupies the same percentage of the destination document as the link source in the source document. In that case

$$h = (l_s/2)(n_d/n_s)$$

where l_s is the length of the link source (query) and n_s is total length of the source document.

Alternatively, as the choice of h determines the width of the destination offset interval, it can be the user's responsibility to estimate its value explicitly or implicitly by specifying in bytes, text objects or percentage of the destination document what text length he wants to follow.

Fig. 3 is a good example that shows how by the changing of the value of h we can change the density function and get a completely different optimal offset interval.

The offset addresses n_1 and n_2 will probably be within a sentence. Such a destination might not be appropriate enough and we go on to step three.

At the third step, we try to improve the retrieval link destination. The simplest solution is to expand the $[n_1, n_2]$ interval to the start and end of a sentence. It means that we should find a new offset interval $[n_3, n_4]$ $(n3 \leq n_1 \& n_4 \geq n_2)$ where n_3 is at the start and n_4 is at the end of a sentence. It is also possible to use various adjustments (extensions or compression) to the destination.

Here, it is important to note that such an adjustment changes the value of h, which could result in losing optimality. However, it seems desirable to offer the user destination with completed sentences. The alternative is clear. If we do not want to lose anything of the destination optimality we have to present the user the destination produced after the step 2.

3. Implementation Remarks

In this section, we will focus on the implementation aspects of the proposed technique and the algorithm complexity.

The complexity of the first step, finding important terms, is unimportant because it is included in the standard procedure of retrieval link calculation where, in the process of similarity estimation between the query and the document index, we should only mark the document index terms which participate in the similarity score.

The second step, finding offset interval with maximal density of the important term weights, could be sufficiently time consuming. To find the maximal density of the important term weights we can simply calculate its values starting from each document offset and choose the maximal one. Such a straightforward approach is unattractive from the performance point of view.

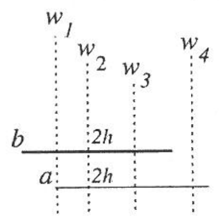


Fig. 4. A case of interval location

We will show that the maximal value of density for fixed h is achieved in an interval $[a, a+2h]$ and one maximum is exactly in the point $a+h$, where a is the offset of an important term.

Let us suppose that the maximum is achieved in an interval $[b, b+2h]$ where b is not an important term offset (see Fig. 4).

In that case we translate the interval to the right until the nearest important term offset is reached. It is clear that

$$\max_{x\in[b,b+2h]}\{den(x,h)\} \leq \max_{x\in[a,a+2h]}\{den(x,h)\}$$

because we can only gain but do not lose weights from important term locations. One maximum is in $a+h$ because from that point we cover the whole $[a, a+2h]$ interval and accumulate all important term weights in that interval.

Now, we can locate the maximum density by examining intervals with length $2h$ starting from important term offsets and accumulate the weights of the important terms in each interval.

The following algorithm uses the described approach to determine the optimal retrieval link destination.

Algorithm: Determine optimal retrieval link destination.
Input: $O = \{(s_1, w_{s_1}), (s_2, w_{s_2}), \ldots, (s_l, w_{s_l})\}$ - Offset locations of important terms, s-start, w-weight, h - half width of expected destination, n_d - Document length in bytes.
Output: $[n_1, n_2]$ - Optimal retrieval link destination

```
procedure $RLD(O, h, n_d)$
/* sort important term ofsets */
sort set $O$ according to the values $s_1, s_2, \ldots, s_l$
$s_{l+1} = n_d$
$optimald = 0$
for $i = 1$ to $l$ do
        $lend = maxd = 0$
        $j = i$
        /* accumuulate term weights in $[s_i, s_i + 2h]$ */
        while $lend \leq 2h$ and $j \leq l$ do
            $lend = lend + s_{j+1} - s_j$
            $maxd = maxd + w_{s_j}$
            $j = j + 1$
        endwhile
        /* is new maximum greater */
        if $maxd > optimald$ then
            $optimald = maxd$
            $n_1 = s_i$
        endif
        /* exit cicle if examined nterval exceds document length */
        if $s_i + 2h > n_d$ exit
endfor
/* is maximal density enough high */
if $optimald/2h \geq 1/1.5h$ then
$n_2 = \min(n_1 + 2h, n_d)$ else
            $n_1 = 1$
            $n_2 = n_d$
return
```

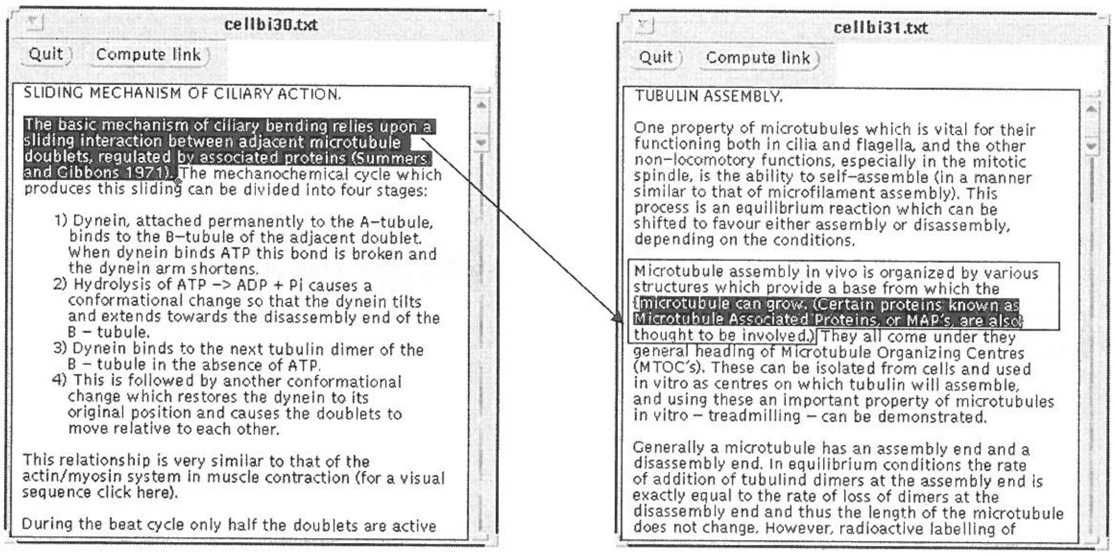
The above algorithm, in the case of more maximal values, determines the interval which corresponds to the first one.

The main **for** cycle operates l times or less because of the statement **if** $(s_i+2h \geq n_d)$ **exit**. The nested **while** cycle operates on the number of important terms in the intervals $s_i + 2h$ for each i. The greater the value of h decreases operations of the **for** cycle and increases the operations in the **while** cycle. The maximal number of operations is reached when both cycles operate the same number of times, $(l/2) * (l/2) = l^2/4$.

As the preparation sort activity takes only $l * \log l$ steps, the maximal time complexity of the algorithm is $O(l^2)$ where l is the total number of the offset addresses of important terms.

Here, it is important to emphasise that l^2 is not an usual square factor because l increases little with the document growth. The number of important terms is usually small because they are determined by the intersection between the retrieval link source (selected document part which after stop words elimination usually contains a small number of different terms) and the index from destination document.

The third step of the technique, in it simplest variant (expanding the $[n_1, n_2]$ interval until the start and end of corresponding sentences) is low time consuming and has no influence on the performances.



SLIDING MECHANISM OF CILIARY ACTION.

The basic mechanism of ciliary bending relies upon a sliding interaction between adjacent microtubule doublets, regulated by associated proteins (Summers and Gibbons 1971). The mechanochemical cycle which produces this sliding can be divided into four stages:

1) Dynein, attached permanently to the A-tubule, binds to the B-tubule of the adjacent doublet. When dynein binds ATP this bond is broken and the dynein arm shortens.
2) Hydrolysis of ATP -> ADP + Pi causes a conformational change so that the dynein tilts and extends towards the disassembly end of the B – tubule.
3) Dynein binds to the next tubulin dimer of the B – tubule in the absence of ATP.
4) This is followed by another conformational change which restores the dynein to its original position and causes the doublets to move relative to each other.

This relationship is very similar to that of the actin/myosin system in muscle contraction (for a visual sequence click here).

During the beat cycle only half the doublets are active

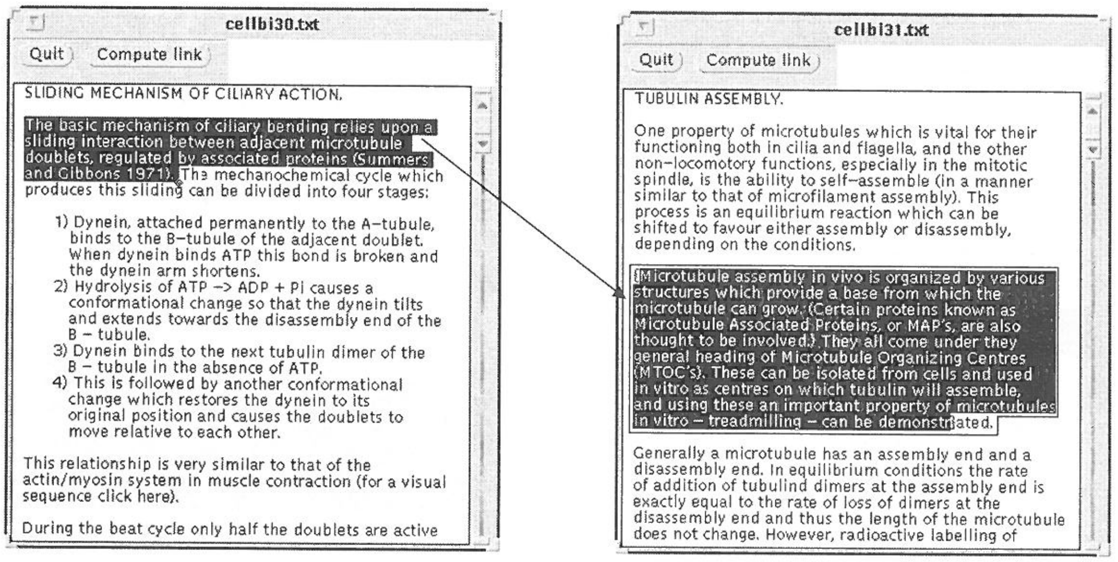
TUBULIN ASSEMBLY.

One property of microtubules which is vital for their functioning both in cilia and flagella, and the other non-locomotory functions, especially in the mitotic spindle, is the ability to self-assemble (in a manner similar to that of microfilament assembly). This process is an equilibrium reaction which can be shifted to favour either assembly or disassembly, depending on the conditions.

Microtubule assembly in vivo is organized by various structures which provide a base from which the microtubule can grow. (Certain proteins known as Microtubule Associated Proteins, or MAP's, are also thought to be involved.) They all come under they general heading of Microtubule Organizing Centres (MTOC's). These can be isolated from cells and used in vitro as centres on which tubulin will assemble, and using these an important property of microtubules in vitro – treadmilling – can be demonstrated.

Generally a microtubule has an assembly end and a disassembly end. In equilibrium conditions the rate of addition of tubulind dimers at the assembly end is exactly equal to the rate of loss of dimers at the disassembly end and thus the length of the microtubule does not change. However, radioactive labelling of

Example 1. $(h = 50)$



SLIDING MECHANISM OF CILIARY ACTION.

The basic mechanism of ciliary bending relies upon a sliding interaction between adjacent microtubule doublets, regulated by associated proteins (Summers and Gibbons 1971). The mechanochemical cycle which produces this sliding can be divided into four stages:

1) Dynein, attached permanently to the A-tubule, binds to the B-tubule of the adjacent doublet. When dynein binds ATP this bond is broken and the dynein arm shortens.
2) Hydrolysis of ATP -> ADP + Pi causes a conformational change so that the dynein tilts and extends towards the disassembly end of the B – tubule.
3) Dynein binds to the next tubulin dimer of the B – tubule in the absence of ATP.
4) This is followed by another conformational change which restores the dynein to its original position and causes the doublets to move relative to each other.

This relationship is very similar to that of the actin/myosin system in muscle contraction (for a visual sequence click here).

During the beat cycle only half the doublets are active

TUBULIN ASSEMBLY.

One property of microtubules which is vital for their functioning both in cilia and flagella, and the other non-locomotory functions, especially in the mitotic spindle, is the ability to self-assemble (in a manner similar to that of microfilament assembly). This process is an equilibrium reaction which can be shifted to favour either assembly or disassembly, depending on the conditions.

Microtubule assembly in vivo is organized by various structures which provide a base from which the microtubule can grow. (Certain proteins known as Microtubule Associated Proteins, or MAP's, are also thought to be involved.) They all come under they general heading of Microtubule Organizing Centres (MTOC's). These can be isolated from cells and used in vitro as centres on which tubulin will assemble, and using these an important property of microtubules in vitro – treadmilling – can be demonstrated.

Generally a microtubule has an assembly end and a disassembly end. In equilibrium conditions the rate of addition of tubulind dimers at the assembly end is exactly equal to the rate of loss of dimers at the disassembly end and thus the length of the microtubule does not change. However, radioactive labelling of

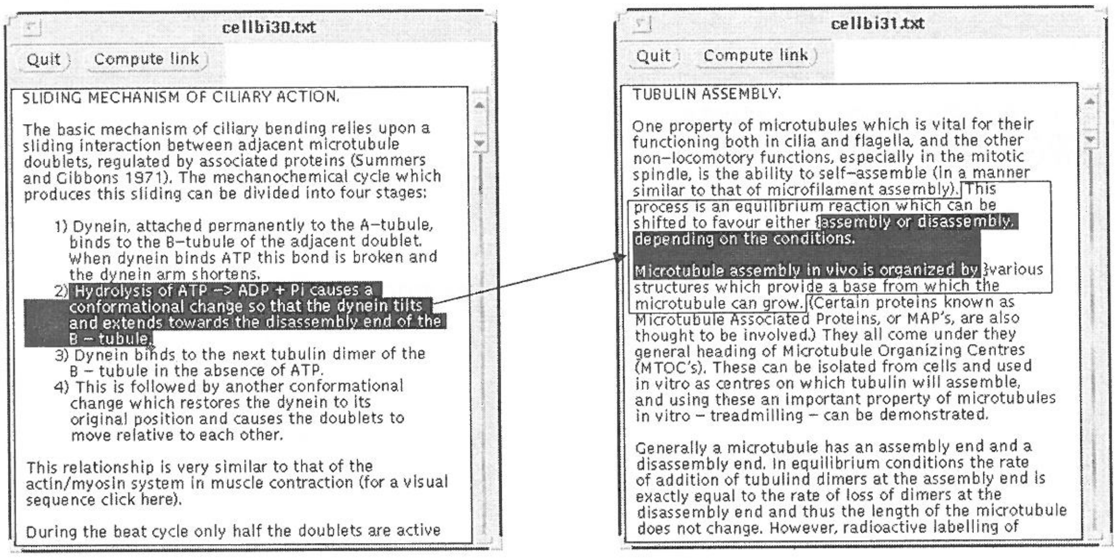
Example 2. $(h = 250)$

Fig. 5. Retrieval link destinations. Illustrates the most common case of nested destination places for the same link source and different values of h

4. Examples

The following six examples shown in Figures 5-7 present results of our technique applied to two text documents. The domain of the documents is cell biology. The retrieval link sources are presented in inverse text mode. The retrieval link destinations after step two are enclosed in {}. Final link destinations after step three are enclosed within the full line. As our aim is to present the link sources and destinations, we do not want to cover with menus the important text parts. Thus, window options are maximally simplified and offer only determination of retrieval link destinations ("Compute link" action).

In its automatic variant (determination of h using the percentage of the retrieval link source), our technique need not change the current user interface. The alternative needs only specification of a number representing the desired length of the link destination (the value of h) in bytes, text objects or percentage of the destination document.



SLIDING MECHANISM OF CILIARY ACTION.

The basic mechanism of ciliary bending relies upon a sliding interaction between adjacent microtubule doublets, regulated by associated proteins (Summers and Gibbons 1971). The mechanochemical cycle which produces this sliding can be divided into four stages:

1) Dynein, attached permanently to the A–tubule, binds to the B–tubule of the adjacent doublet. When dynein binds ATP this bond is broken and the dynein arm shortens.
2) Hydrolysis of ATP –> ADP + Pi causes a conformational change so that the dynein tilts and extends towards the disassembly end of the B – tubule.
3) Dynein binds to the next tubulin dimer of the B – tubule in the absence of ATP.
4) This is followed by another conformational change which restores the dynein to its original position and causes the doublets to move relative to each other.

This relationship is very similar to that of the actin/myosin system in muscle contraction (for a visual sequence click here).

During the beat cycle only half the doublets are active

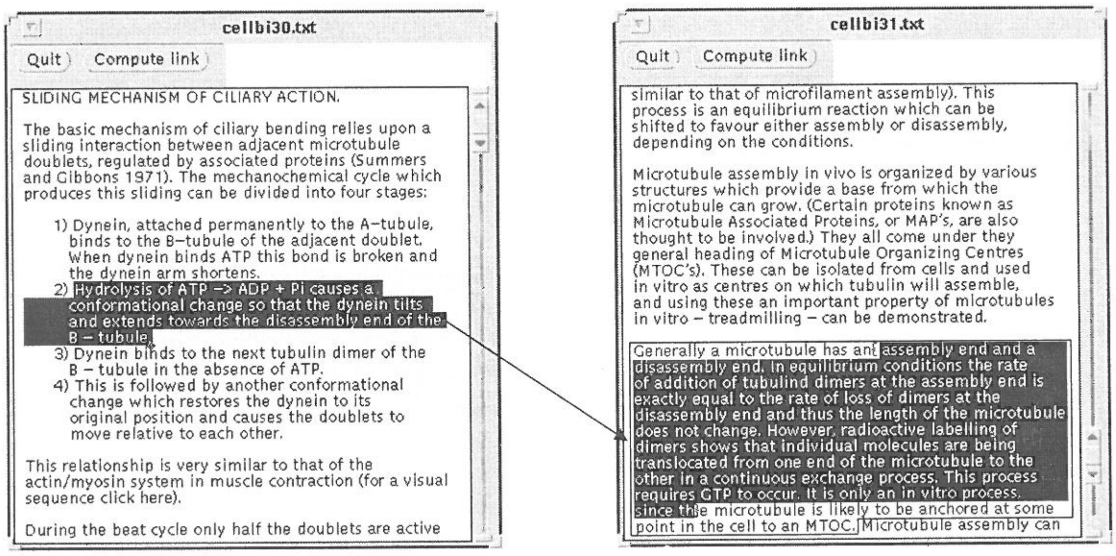
TUBULIN ASSEMBLY.

One property of microtubules which is vital for their functioning both in cilia and flagella, and the other non–locomotory functions, especially in the mitotic spindle, is the ability to self–assemble (in a manner similar to that of microfilament assembly). This process is an equilibrium reaction which can be shifted to favour either {assembly or disassembly, depending on the conditions.

Microtubule assembly in vivo is organized by }various structures which provide a base from which the microtubule can grow. (Certain proteins known as Microtubule Associated Proteins, or MAP's, are also thought to be involved.) They all come under they general heading of Microtubule Organizing Centres (MTOC's). These can be isolated from cells and used in vitro as centres on which tubulin will assemble, and using these an important property of microtubules in vitro – treadmilling – can be demonstrated.

Generally a microtubule has an assembly end and a disassembly end. In equilibrium conditions the rate of addition of tubulind dimers at the assembly end is exactly equal to the rate of loss of dimers at the disassembly end and thus the length of the microtubule does not change. However, radioactive labelling of

Example 3. ($h = 50$)



SLIDING MECHANISM OF CILIARY ACTION.

The basic mechanism of ciliary bending relies upon a sliding interaction between adjacent microtubule doublets, regulated by associated proteins (Summers and Gibbons 1971). The mechanochemical cycle which produces this sliding can be divided into four stages:

1) Dynein, attached permanently to the A–tubule, binds to the B–tubule of the adjacent doublet. When dynein binds ATP this bond is broken and the dynein arm shortens.
2) Hydrolysis of ATP –> ADP + Pi causes a conformational change so that the dynein tilts and extends towards the disassembly end of the B – tubule.
3) Dynein binds to the next tubulin dimer of the B – tubule in the absence of ATP.
4) This is followed by another conformational change which restores the dynein to its original position and causes the doublets to move relative to each other.

This relationship is very similar to that of the actin/myosin system in muscle contraction (for a visual sequence click here).

During the beat cycle only half the doublets are active

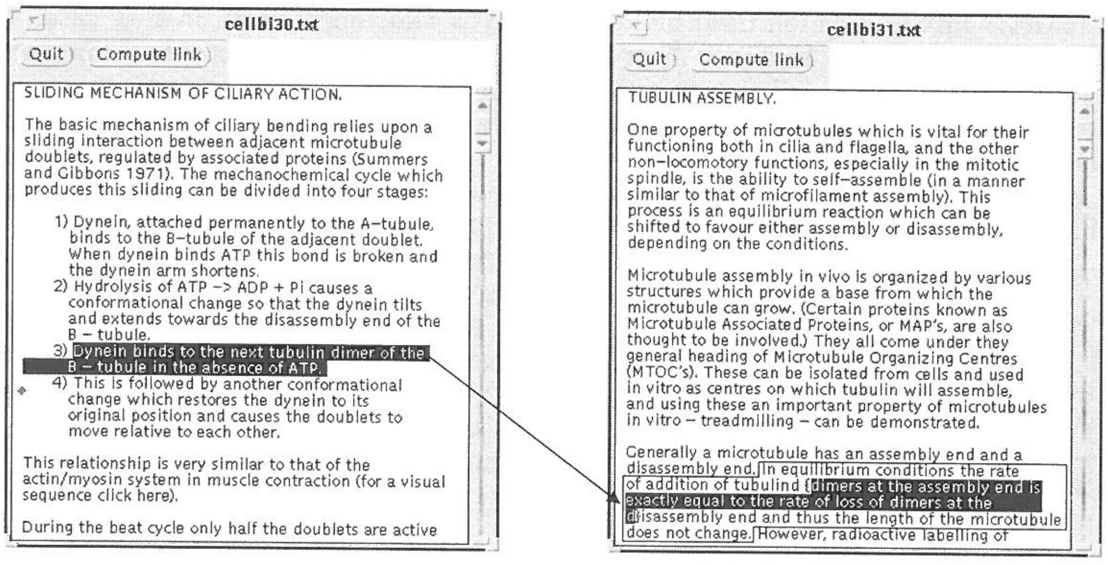
similar to that of microfilament assembly). This process is an equilibrium reaction which can be shifted to favour either assembly or disassembly, depending on the conditions.

Microtubule assembly in vivo is organized by various structures which provide a base from which the microtubule can grow. (Certain proteins known as Microtubule Associated Proteins, or MAP's, are also thought to be involved.) They all come under they general heading of Microtubule Organizing Centres (MTOC's). These can be isolated from cells and used in vitro as centres on which tubulin will assemble, and using these an important property of microtubules in vitro – treadmilling – can be demonstrated.

Generally a microtubule has an{ assembly end and a disassembly end. In equilibrium conditions the rate of addition of tubulind dimers at the assembly end is exactly equal to the rate of loss of dimers at the disassembly end and thus the length of the microtubule does not change. However, radioactive labelling of dimers shows that individual molecules are being translocated from one end of the microtubule to the other in a continuous exchange process. This process requires GTP to occur. It is only an in vitro process, since th}e microtubule is likely to be anchored at some point in the cell to an MTOC. Microtubule assembly can

Example 4. ($h = 250$)

Fig. 6. Retrieval link destinations. Illustrates the case where because of different values of h the destination places are non-intersected for the same link source



SLIDING MECHANISM OF CILIARY ACTION.

The basic mechanism of ciliary bending relies upon a sliding interaction between adjacent microtubule doublets, regulated by associated proteins (Summers and Gibbons 1971). The mechanochemical cycle which produces this sliding can be divided into four stages:

1) Dynein, attached permanently to the A–tubule, binds to the B–tubule of the adjacent doublet. When dynein binds ATP this bond is broken and the dynein arm shortens.
2) Hydrolysis of ATP –> ADP + Pi causes a conformational change so that the dynein tilts and extends towards the disassembly end of the B – tubule.
3) Dynein binds to the next tubulin dimer of the B – tubule in the absence of ATP.
4) This is followed by another conformational change which restores the dynein to its original position and causes the doublets to move relative to each other.

This relationship is very similar to that of the actin/myosin system in muscle contraction (for a visual sequence click here).

During the beat cycle only half the doublets are active

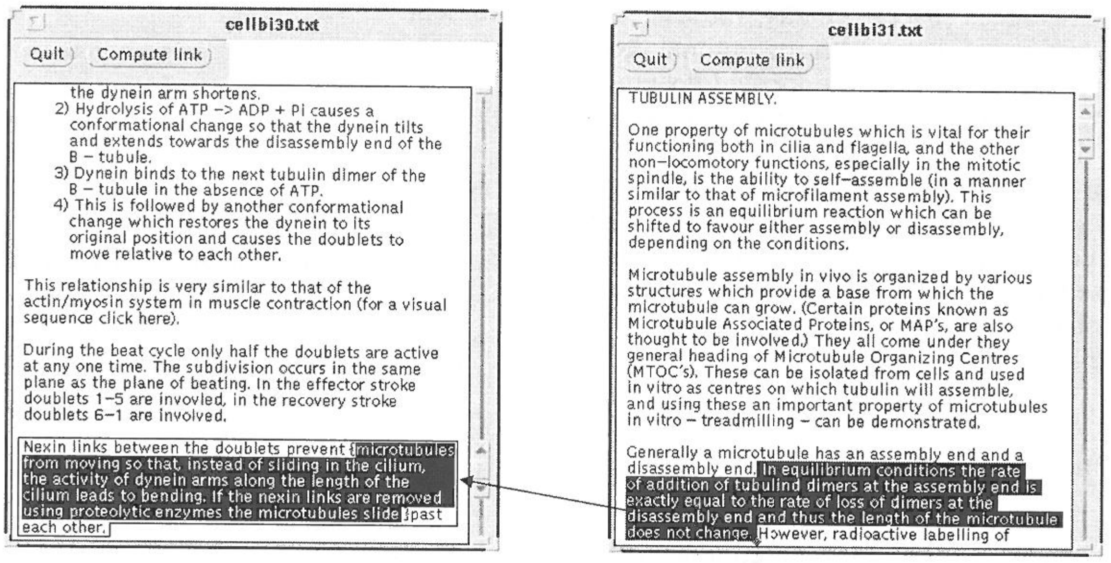
TUBULIN ASSEMBLY.

One property of microtubules which is vital for their functioning both in cilia and flagella, and the other non–locomotory functions, especially in the mitotic spindle, is the ability to self–assemble (in a manner similar to that of microfilament assembly). This process is an equilibrium reaction which can be shifted to favour either assembly or disassembly, depending on the conditions.

Microtubule assembly in vivo is organized by various structures which provide a base from which the microtubule can grow. (Certain proteins known as Microtubule Associated Proteins, or MAP's, are also thought to be involved.) They all come under they general heading of Microtubule Organizing Centres (MTOC's). These can be isolated from cells and used in vitro as centres on which tubulin will assemble, and using these an important property of microtubules in vitro – treadmilling – can be demonstrated.

Generally a microtubule has an assembly end and a disassembly end. In equilibrium conditions the rate of addition of tubulind {dimers at the assembly end is exactly equal to the rate of loss of dimers at the disassembly end and thus the length of the microtubule does not change. However, radioactive labelling of

Example 5. (h =same percentage of the destination document as link source in the source document)



the dynein arm shortens.
2) Hydrolysis of ATP –> ADP + Pi causes a conformational change so that the dynein tilts and extends towards the disassembly end of the B – tubule.
3) Dynein binds to the next tubulin dimer of the B – tubule in the absence of ATP.
4) This is followed by another conformational change which restores the dynein to its original position and causes the doublets to move relative to each other.

This relationship is very similar to that of the actin/myosin system in muscle contraction (for a visual sequence click here).

During the beat cycle only half the doublets are active at any one time. The subdivision occurs in the same plane as the plane of beating. In the effector stroke doublets 1–5 are invovled, in the recovery stroke doublets 6–1 are involved.

Nexin links between the doublets prevent {microtubules from moving so that, instead of sliding in the cilium, the activity of dynein arms along the length of the cilium leads to bending. If the nexin links are removed using proteolytic enzymes the microtubules slide }past each other.

TUBULIN ASSEMBLY.

One property of microtubules which is vital for their functioning both in cilia and flagella, and the other non–locomotory functions, especially in the mitotic spindle, is the ability to self–assemble (in a manner similar to that of microfilament assembly). This process is an equilibrium reaction which can be shifted to favour either assembly or disassembly, depending on the conditions.

Microtubule assembly in vivo is organized by various structures which provide a base from which the microtubule can grow. (Certain proteins known as Microtubule Associated Proteins, or MAP's, are also thought to be involved.) They all come under they general heading of Microtubule Organizing Centres (MTOC's). These can be isolated from cells and used in vitro as centres on which tubulin will assemble, and using these an important property of microtubules in vitro – treadmilling – can be demonstrated.

Generally a microtubule has an assembly end and a disassembly end. In equilibrium conditions the rate of addition of tubulind dimers at the assembly end is exactly equal to the rate of loss of dimers at the disassembly end and thus the length of the microtubule does not change. However, radioactive labelling of

Example 6. The link source is the link destination from the example 5.

Fig. 7. Retrieval link destinations with automatic determination of h. Illustrates expected non-symmetry of the operation

The first example in Fig. 5 shows retrieval link source and destination of one sentence. The destination is determined for $h = 50$ which approximately corresponds to one sentence.

In the second example the destination is determined for $h = 250$ which corresponds to the group of sentences or a small paragraph.

These examples show the most common case when destination document parts are nested depending on the length of the destination.

The third ($h = 50$) and fourth ($h = 250$) examples in Fig. 6 show the case where, because of the different length of the destination (different values of h), we have non-intersecting text intervals as the optimal retrieval link destinations for the same retrieval link source.

The last two examples presented in Fig. 7 are produced by using automatic determination of the length of the retrieval link destinations. The h is determined in such way that the destination interval occupies the same percentage of the destination document as the link source interval in the source document. The sixth example uses a retrieval link destination from the fifth example as the retrieval link source to illustrate the expected non-symmetry of the operation. This non-symmetry is based on the possibility that found destination has other important terms whose influence can result in different source part when link is resolved "backwards".

The link following action, common for all hypertext systems, is the natural place for activating the procedure of determination of optimal retrieval link destination. The user marks a part of the current text document determining his or her query and selects "Compute link" from the action menu. The system will determine a number of most relevant documents to the query using document indexes and the standard retrieval strategy. Then, the procedure of determination of optimal destination within the determined documents takes place. Finally, the user will be led to the document with the optimal destination and positioned at the located document part. Actually, the similarity between query and whole documents is performed as a preparation task for finding important terms and initial document selection and ranking. This corresponds to the concept of global/local text comparison proposed in (Salton et al., 1993a) and increases the probability that "relevant document parts" will be found sooner.

4. Conclusion

In this paper, we have considered a technique for the automatic determination of appropriate retrieval link destinations within documents in a hypertext system. For that purpose, an efficient technique which uses density estimation of important term weights within the destination document has been proposed.

This density estimation can be seen as a kind of similarity measure between the query and arbitrary document parts, which uses only query term weights, normalised by the length of the corresponding document part. The power of such a similarity measure is in local text comparison where we do not dispose of the real document term weights.

In essence, the results of our technique are very close to others which use document decomposition and to obtain similarity treat each document part as a document itself. In most cases, the results will be the same, because the interval with maximal density of the important term weights will produce the maximal similarity with the query as the important terms have high weights in the query and exist in the corresponding document part. However, our destination is not connected to fixed text objects (section, paragraph or sentence). The approach to the problem using the concept of density makes our technique more elegant and simple to use. It seems difficult to achieve the advantages of our approach by straightforward application of an IR technique to smaller document parts which is the basic intention of other proposed approaches.

We believe that our technique can be useful in all situations where it is necessary for a given query to determine the most relevant part of a document.

References

[1] AGOSTI, M., CREASTANI F., MELLUCI, M., On the Use of Informationn Retrieval Technniques for the Automatic Construction of Hypertext, *Information Processing and Management*, Vol.33, No.2 (1997), 133–144.

[2] ALLEN, J., Building Hypertext Using Information Retrieval, Information Processing and Management, Vol.33, No.2 (1997), 145–159.

[3] BERNSTEIN, M., An Apprentice that Discovers Hypertext Links, In Rizk A., Streitz N., & Andre J. (Ed.), *Hypertext: Concepts Systems and Applications, Proceedings of the European Conference on Hypertext*, (1991) France, Cambridge University Press, 212–223.

[4] BUCKLEY, C., SINGHAL, A., MITRA, M., New Retrieval Approaches Using SMART : TREC 4, *Proceedings of the 4th TREC Conference (TREC-4)*, (1995) Marylend, 25–49.

[5] CALLAN, J.P., "Passage-Level Evidence in Document Retrieval", *Proceedings of the 17th ACM SIGIR Conference*, (1994) Berlin, 301–310.

[6] CHANG, D.T., HieNet: A User-Centered Approach for Automatic Link Generation, *Hypertext'93: Proceedings of the ACM Hypertext Conference*, (1993) Sietl, 145–158.

[7] CROFT, W.B., TURTLE, H., A Retrieval Model for Incorporating Hypertext Links, *Hypertext'89: Proceedings of the ACM Hypertext Conference*, (1989) Pittsburgh, 213–224.

[8] DEROSE, S.J., Expanding the Notion of Links, *Hypertext'89: Proceedings of the ACM Hypertext Conference*, (1989) Pittsburgh, 249–257.

[9] DEVROYE, L., GYORFI, L., *Nonparametric Density Estimation*, John Wiley & Sons, New York 1985.

[10] FREI, H.P., STIEGER, D., The Use of Semantic Links in Hypertext Information Retrieval, *Information Processing and Management*, Vol.31, No.1 (1995), 1–13.

[11] FRISSE, M.E., COUSINS, S.R., (1988), Searching for Information in a Hypertext Medical Handbook, *Communications of the ACM*, Vol.31, No.7 (1988), 23–46.

[12] KASZKIEL, M., ZOBEL, J., Passage Retrieval Revisited, *Proceedings of the 20th ACM SIGIR Conference*, (1997) Philadelphia, 178–185.

[13] SALTON G., *Automatic Text Processing*, Addison–Wesley, New York (1989).

[14] SALTON, G., ALLAN, J., BUCKLEY, C., Approaches to Passage Retrieval in Full Text Information Systems, *Proceedings of the 16th ACM SIGIR Conference*, (1993a) New York, 49–58.

[15] SALTON, G., ALLAN, J., Selective Text Utilisation and Text Traversal, *Hypertext'93: Proceedings of the ACM Hypertext Conference*, (1993b) Seattle, 131–144.

[16] SALTON, G., SINGHAL, A., BUCKLEY, C., MITRA, M., Automatic Text Decomposition Using text Segments and Text Themes, *Hypertext'96: Proceedings of the ACM Hypertext Conference*, (1996) , 53–65.

[17] *Savoy, J.,* Effectiveness of Information Retrieval Systems Used in Hypertext Environment, *Hypermedia*, Vol.5, No.1 (1993), 23–46.

Received: November, 1997
Revised: August, 1998
Accepted: September, 1998

Contact address:
Dusan Cakmakov
Faculty of Mechanical Eng.
University of Skopje
Karpos II bb
Skopje
Macedonia
e-mail: dusan@ereb.mf.ukim.edu.mk
phone: +389 91 363566
fax: +389 91 362298

DUSAN CAKMAKOV obtained the degree in mathematics and computer science from the Faculty of Mathematics & Computer Science and his MS and PhD in computer science from the faculty of Electrical Eng. & Computer Science, University of Skopje, Macedonia in 1982, 1988 and 1992 respectively.

In 1994 he was research scientist on Hypermedia and information retrieval at the University of Southampton, UK and in 1996 visiting professor at the University Paris XIII, France.

Currently, Dr. Cakmakov is an Associate Professor in mathematics and computer science at the faculty of Mechanical Eng., University of Skopje, Macedonia. His current research interests include text and image retrieval systems, multimedia systems and pattern recognition.