Information retrieval by fuzzy relations and hierarchical co-occurrence[1]

Part I

László T. Kóczy[2]
Dept. of Telecommunication and Telematics
Technical University of Budapest
Budapest H-1521 Hungary
Fax: +36-1-463-3107 Phone: +36-1-463-4190 E-mail: koczy@boss.ttt.bme.hu

Tamás D. Gedeon
Dept. of Information Engineering
School of Computer Science and Engineering
University of New South Wales
Sydney 2052 Australia
Fax: +61-2-9385-5995 Phone: +61-2-9385-3965 E-mail: tom@cse.unsw.edu.au

Abstract

The study treats the problem of automatic indexing and retrieval of documents where it cannot be guaranteed that the user queries include the actual words that occur in the documents that should be retrieved. Fuzzy tolerance and similarity relations are presented and the notion of hierarchical co-occurrence is defined that allows the introduction of two or more hierarchical categories of words in the documents. If the query is based on a single keyword it is possible to extend the query to the compatibility (or equivalence) class of the queried word and so, directly matching documents or a class of matching words established by some sample document collection and then documents matching with words in this latter class can be retrieved. Various methods will be proposed and illustrated, with the intention of real application in legal document collections.

1. Introduction

An information retrieval system allows users to efficiently retrieve documents that are relevant to their current interests. The main problem is that the collection of documents from which the selected ones have to be retrieved might be extremely large, and often heterogeneous from various points of view: especially in the structure and the use of terminology. This is very obvious with areas where the language of the documents is close to natural language usage like in legal texts that form the main target of this study.

A user typically specifies their interests via individual words or sets of words (phrases), that are fragments of natural language texts. There is no guarantee that the words specified in the query always exactly match the words used in the various documents in the collection, even though the contents of the documents might be

[1] Supported by the Australian Research Council, Grant No. A49600961.
[2] Visiting Professor at the Dept. of Information Engineering, School of Computer Science and Engineering, University of New South Wales, E-mail: koczy@cse.unsw.edu.au.

relevant in the context of the query. If e.g. only a synonym is used in the text that has not been included in the query, a very relevant document might be completely left out of consideration. There are also often documents with related concepts that might be important for the user, but they are not aware of the fact of certain areas or concepts being tightly connected with the target topic(s) in the query, and so, important information might be lost if the search is not extended to these related topics.

2. Hierarchical co-occurrence

In this study a method will be introduced that is based on fuzzy relations, especially similarity (equivalence) and tolerance (compatibility) relations, but possibly also full or partial orderings, and that allows the "concentric" extension of searches based on what we will call hierarchical co-occurrence of words and phrases. By hierarchical co-occurence the following is meant: Almost every document has a certain hierarchical structure concerning the importance of the words or concepts occurring in it. It can be assumed that every document has a *title* which contains certainly relevant information concerning the contents. Most documents also contain *sub-titles*, etc. and some of them have a collection of *keywords* at the beginning of the text. A finer classification of approaches useful for automatic indexing of the context can be found in [1] and [2]:

1. Frequency-keyword approach. (In this context all informative words in the text are called keywords, however, in the next, we will restrict the usage of this term to words occurring on higher logical hierarchical levels.)
2. Title-keyword approach. (using only higher hierarchical levels in the document, such as titles, subtitles, headings, etc.)
3. Location method. (Using the introduction and/or conclusion of the document or of each paragraph.)
4. Cue method. (It is based on semantic observations concerning the effect of some special words or expressions in the vicinity of a given word, such as "significant" or "impossible".)
5. Indicator-phrase method. (Is also based on semantic contexts, like "The main aim of this paper is...", etc.)
6. Structural observations.

For the purposes of the hierarchical co-occurrence approach especially the methods 1, 2, and 3 will be important, combined with some aspects of 6. We do not reject the significance of approaches 4 and 5, but in the first introduction of our new method, the former ones will be directly considered as the main sources of information to determine the relational system in the given collection of documents.

The basic idea of automatic indexing based on co-occurrence is that words or phrases occurring frequently together in the same document or even paragraph are connected in their meaning in some way. Certainly, this will not mean that such words are necessarily synonyms or have related meanings, as often antonyms occur together just as frequently as synonyms, not to speak about more sophisticated semantic connections. The simplest idea is to check words in the sense of approach 1, and instead of linking documents with words, establishing a matrix or co-occurrence graph indicating the mutual co-occurence of pairs of words and phrases. A finer

model will be introduced when the degree of co-occurrence is described by a membership degree in the sense of fuzzy logic.

A more sophisticated approach is the hierarchical approach. In this, the supposed semantic structure of the documents is taken into consideration in the following way: We assume that the title is descriptive for the contents of the paper. The words occurring in the title, except the non-important words like articles, or connectives, in this particular context, should be very important for the whole contents of the document. Similarly, the sub-title of each section, sub-section, etc. of the document is assumed to be descriptive for the contents of the relevant sub-unit. In this sense, there is a hierarchical semantic structure in the document that contains at least two levels (1: title and eventual keywords, 2: text), but possibly more than two (e.g. 1: title and keywords, 2: sub-titles, 3: texts) that can be represented by a tree graph as in Fig. 1. In the case of sub-sub-titles, etc., the number of levels increases in a similar way.)

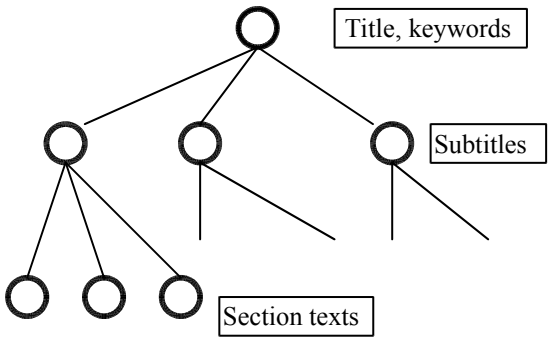

Figure 1.

If this concept is compared with the automatic indexing methods listed above it is found that the terminology needs a slight change: In order to avoid confusion, the term “keyword” will be restricted to a similar concept as was introduced under the term “Title-keyword approach”, including essential words in the titles, the “keywords” in the usual sense (comprising the expressions listed under the heading “Keywords”), possibly the subtitles (depending on how many levels are considered), and finally, depending on the problem, even the essential words in the special location areas (introduction and conclusion). From now on, the term keyword will mean all the words that are somehow highlighted in a document by their special positions, and so, it is reasonable to assume that they contain references to the most significant aims of that document. However, if more than two (keyword and general word) levels are considered in the model, it will be necessary to introduce additional terminology: “keywords” for the words occurring in the title and the “Keywords” section (and maybe in the introductory and conclusion part of the whole document), “sub-keywords” for the terms occurring in the sub-titles (and corresponding introductions and conclusions), etc., and “words” for the lowest level comprising the contents of the whole document. Let us denote the set of keywords for a given collection of documents $D = \{ D_1, D_2, \ldots, D_n \}$ by $K(D)$, and if there is a further hierarchy of the keyword levels, by $K_1(D)$, $K_2(D)$, etc., and the set of all significant words by *W*. Then it is advisable to define these sets so that

$$K_1(D) \subset K_2(D) \subset \ldots \subset K_m(D) \subset W$$

where *m* denotes the number of hierarchical levels taken into consideration ($m \geq 1$).

The main idea is now the following. If a certain word or phrase is frequently occurring together with another one in the same document, the two might have connected meaning or significance. If a word or phrase is frequently occurring in a document, or segment of a document of which the keywords (in the title, etc.) are certain other words, the former ones would belong to the class of related concepts of the latter ones. The more frequent is the co-occurrence, including the concept of "hierarchical co-occurrence" as well (meaning that certain words $\{w_i\}$ appear often in texts that are titled or marked by certain other words $\{W_j\}$, where very likely $\{W_j\} \subset \{w_i\}$, however, even $\{W_j\} \cap \{w_i\} = \varnothing$ cannot be excluded!), the more it is likely that any user querying for any W_j will be interested in documents containing w_i in the text - even if the queried word does not appear in the title of these latter documents, maybe not in the text at all.

As an example let us take somebody who is interested in articles on Soft Computing or Computational Intelligence. In many overview articles on these subject, the term Fuzzy Logic will occur frequently. However, it is very likely that in an article on Fuzzy Logic none of the terms Soft Computing or Computational Intelligence will occur. In this case it is quite clear that the connection between SC and FL is hierarchical in the meaning, and the structure of many documents will follow this, as shown in Fig. 2.

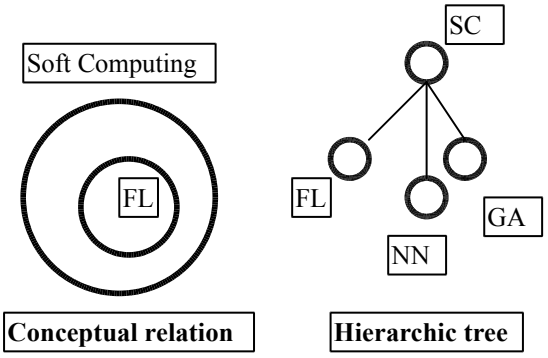

Figure 2.

The left hand side of the picture expresses that Fuzzy Logic is a special branch of Soft Computing, and so, it is a subset of the topic marked by the keyword SC. The right hand side shows that articles on SC include those related to Fuzzy Logic, Neural Networks, Genetic Algorithms, etc. In the next section we attempt to introduce a model that is suitable for finding documents *not containing the words "Soft Computing"* but dealing e.g. with Fuzzy Logic, by querying for "Soft Computing", and *not asking for "Fuzzy Logic"* at all.

3. Fuzzy relations

In this section, a short overview will be given on fuzzy relations in general, and a few important types of fuzzy and crisp relations that will be referred to in the next sections. In this section we also present some simple examples in order to

introduce the method proposed in the next part of the study. For further details on fuzzy relations it is recommended that the reader consult [3] or some other textbook.

A fuzzy set A is always defined in terms of a universe of discourse $X=\{x\}$ and a mapping μ_A from this set to the unit interval $[0,1]$: $\mu_A: X \to [0,1]$, where $\mu_A(x)$ is called the membership function of the fuzzy set A, and its concrete values for any $x=x_0$ are the membership grades of x_0 in A. A fuzzy relation is a fuzzy set of the Cartesian product of two or more sets as the universe, so e.g. a binary fuzzy relation R is defined by the mapping $\mu_R: X \times Y \to [0,1]$ where $X=\{x\}, Y=\{y\}$ and consequently $X \times Y = \{(x,y)\}$. It is a special case when $Y=X$, i.e. the binary relation is over the Cartesian square of a given universe

Binary fuzzy relations of $X \times X$ are categorized according to their properties in a similar manner to ordinary (crisp) relations. Equivalence relations ($\equiv$) in the crisp sense are defined by the fulfillment of three properties: reflexivity ($x \equiv x$ is always true), symmetry ($x \equiv y \Rightarrow y \equiv x$), and transitivity ($x \equiv y \wedge y \equiv z \Rightarrow x \equiv z$). The fuzzy analog of equivalence is called the *similarity relation* ($\cong$), and essentially the same three properties hold, except that transitivity has to be formulated in a somewhat different manner:

$$\mu_{\cong}(x,x)=1, \mu_{\cong}(x,y)=\mu_{\cong}(y,x), \mu_{\cong}(x,z) \geq \min\{\mu_{\cong}(x,y), \mu_{\cong}(y,z)\}.$$

Compatibility relations are reflexive and symmetric, but not necessarily transitive as well, so they form a wider class than equivalence. The fuzzy analog is called *tolerance relation* ($\approx$), and it has the first two properties as above:

$$\mu_{\approx}(x,x)=1, \mu_{\approx}(x,y)=\mu_{\approx}(y,x).$$

Although in this paper mainly the above two types of relations will be discussed, also full and partial orderings will be introduced. A crisp ordering relation ($\leq$) is reflexive, antisymmetric and transitive, the second meaning that $x \leq y \wedge y \leq x \Rightarrow x=y$. A full or linear ordering assumes that for all pairs in $X \times Y$ either $x \leq y$ or $y \leq x$ must be true. In a partial ordering, a pair of x and y might be incomparable, i.e. $(x,y) \notin \leq$. *Fuzzy orderings* are defined by the following:

$$\mu_{\pi}(x,x)=1, \mu_{\pi}(x,y)>0 \Rightarrow \mu_{\pi}(y,x)=0, \mu_{\pi}(x,z) \geq \min\{\mu_{\pi}(x,y), \mu_{\pi}(y,z)\}.$$

(In the above properties of relations $x,y,z \in X$ holds everywhere.)

A rather convenient way to represent binary fuzzy relations of finite element universes is the use of matrices, where columns and rows correspond to the elements of the component universes X and Y and elements of the matrix are the membership degrees themselves:

	y_1	y_2	y_3	...
x_1	$\mu(x_1,y_1)$	$\mu(x_1,y_2)$	$\mu(x_1,y_3)$	...
x_2	$\mu(x_2,y_1)$	$\mu(x_2,y_2)$	$\mu(x_2,y_3)$	...
...	...	...	...	...

The same information can be visualized by a bipartite graph as in Fig. 3.

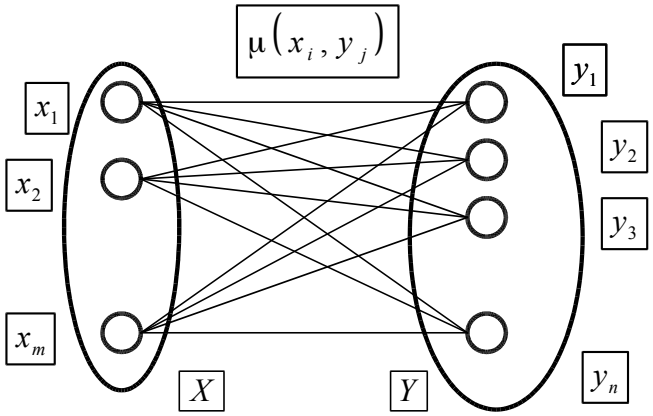

Figure 3.

Similarly, relations of $X \times X$ can be described by quadratic matrices as in Table 1:

	x_1	x_2	x_3	...
x_1	$\mu(x_1, x_1)$	$\mu(x_1, x_2)$	$\mu(x_1, x_3)$	...
x_2	$\mu(x_2, x_1)$	$\mu(x_2, x_2)$	$\mu(x_2, x_3)$	...
...	...	...	...	...

Table 1.

where e.g. similarity and tolerance relations have only 1-s in the diagonals ($\mu(x_i, x_i)$), and are symmetrical, etc. The graphic equivalent of the above matrix is a graph as in Fig. 4.

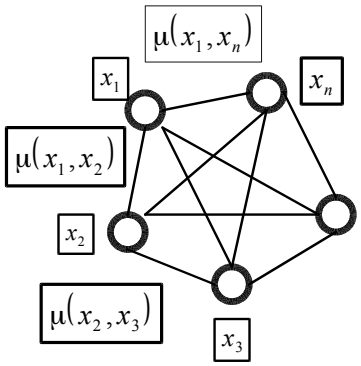

Figure 4.

Selecting an arbitrary $\alpha \in [0,1]$ in a fuzzy graph, the α-cut of the graph contains only those edges to which at least α belongs as the membership degree. If X_i is a node of the graph G representing a similarity relation, the set of all nodes $E(X_i) = \{ X_j \in G \mid \mu(X_i, X_j) \geq \alpha \}$ represents the *equivalence* (similarity) *class* of X_i.

Because of the transitivity and reflexivity properties of the similarity relation it is obvious that

$$X_j, X_k \in E(X_i) \Rightarrow \mu(X_j, X_k) \geq \alpha \text{ and also that } X_i \in E(X_i).$$

Consequently, similarity relations generate α-partitions of the graph. The partition can be represented by an empty graph, where each class in the partition is a node in this new graph.

Tolerance relations behave in another way as tolerance is not transitive. While every node is necessarily an element of its own *tolerance cluster*: $X_i \in T(X_i)$, it is not true that other nodes in $T(X_i)$ are also connected by edges to each other with at least the same degree of membership as the defining node is to both nodes in the class. If an $\alpha \in [0,1]$ is selected, the α-cuts of tolerance classes of the nodes will usually not be complete graphs themselves. On the other hand, if the maximal sub-graph $C_\alpha(X_i)$ of $T(X_i)$ containing X_i itself is selected where every node has at least α membership degree (α-clique), the set of $C_\alpha(X_i)$-s will form a cover of *G*, so that

$$\bigcup_i C_\alpha(X_i) = G \text{ but usually } i \neq j \Rightarrow C_\alpha(X_i) \cap C_\alpha(X_j) \neq \varnothing.$$

The graph generated by $C_\alpha(X_i)$ will usually not be empty, as some nodes of *G* belong to two or more *compatibility classes* simultaneously. (Obviously, the structure of the cover and the generated graph will depend on the selected cut as well.) An example is shown in Table 2 and Fig. 5. Graph *G* contains six nodes, $X_1 \ldots X_6$, Table 2 shows all $\mu(X_i X_j)$. Apparently the relation represented by *G* is not a similarity relation as it is not transitive. Let us take e.g. $\{X_3, X_4, X_5\}$, here

$$\mu(X_3, X_5) = 0.2 < \min\{\mu(X_3, X_4), \mu(X_4, X_5)\} = \min\{0.7, 0.8\} = 0.7,$$

what contradicts the properties of similarity. On the other hand, all $\mu(X_i X_i) = 1$ (the elements in the diagonal of the matrix are all 1-s, the relation is reflexive), and the matrix is symmetrical (the relation is symmetrical itself), consequently *G* represents a tolerance relation. Let us choose $\alpha = 0.7$ and take the α-cut of *G*. Remaining edges are indicated by bold numbers (the elements of the diagonal represented by bold italics). All other degrees are under the boundary of the chosen cut, and so, will fall away from the α-cut of *G*. In Figure 5 all edges above the boundary are indicated with their respective degrees of membership, while the remaining edges are shown without their degrees. G_α, the α-cut of *G* is a crisp graph that represents a crisp compatibility relation that is the α-cut of the original tolerance relation given by *G*.

Let us construct now the compatibility classes of the relation G_α. (It should be mentioned however that searching compatibility classes is an NP-complete task that needs a very long time for larger graphs, cf. E.g. [4]. There exist some faster algorithms for solving this problem approximately, however in this paper we do not intend to go into details of computational complexity questions outside of the main target problem. In this study we just suppose that compatibility classes have been found by either exhaustive search - like in the example - or by a parallel algorithm, or by an approximative algorithm. This can be done as establishing the compatibility

classes has to be done only once, before the information retrieval service is started, in order to have a "logical map" of the knowledge in the data base in question, as it will be seen in the next sections.) The maximal compatibility classes in G_α ($\alpha = 0.7$) are the following:

$$C_\alpha = \{ C_1 = \{ X_1, X_2, X_6 \}, C_2 = \{ X_3, X_4, X_6 \}, C_3 = \{ X_4, X_5, X_6 \} \} .$$

	X_1	X_2	X_3	X_4	X_5	X_6
X_1	***1.0***	**0.7**	0.2	0.5	0.3	**0.8**
X_2	**0.7**	***1.0***	0.0	0.6	0.1	**0.9**
X_3	0.2	0.0	***1.0***	**0.7**	0.2	**0.7**
X_4	0.5	0.6	**0.7**	***1.0***	**0.8**	**0.8**
X_5	0.3	0.1	0.2	**0.8**	***1.0***	**0.9**
X_6	**0.8**	**0.9**	**0.7**	**0.8**	**0.9**	***1.0***

Table 2.

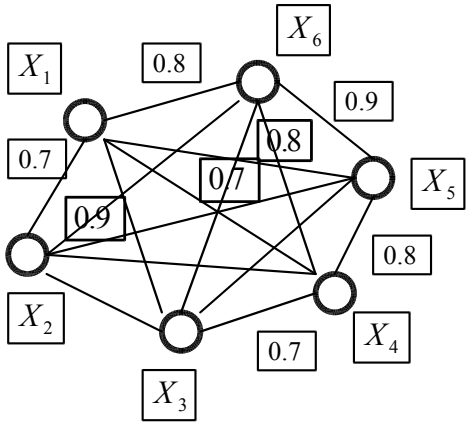

Figure 5.

It is not always necessarily so, but these classes cover the whole graph, and there is no such class which can be omitted so that the remaining still cover *G*. The set of compatibility classes is indicated in Figure 6.

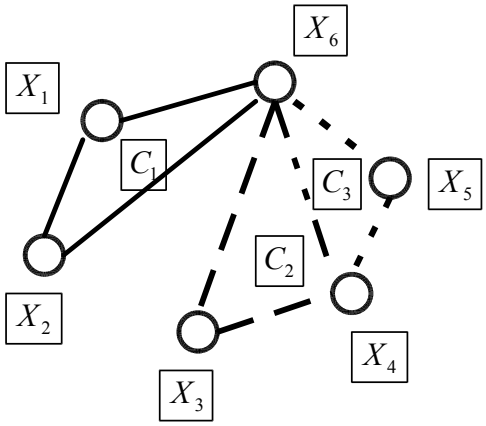

Figure 6.

The graph structure of the compatibility classes themselves can be seen in Fig. 7.

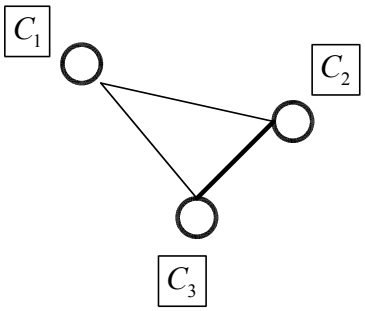

Figure 7.

The class structure is presented in a crisp graph, although the connection between the second and third classes is "stronger" than that between the other two pairs (thick line), as there are two overlapping nodes in the first case and only one in the other two, which fact could be taken into consideration by weighting the edges of the class graph, e.g. by attaching fuzzy membership degrees to its edges.

Finally, it has to be mentioned that relations over $X \times Y$ and $Y \times Z$ can be combined to a single relation over $X \times Z$ by one of the *composition* operations. The definition of the most popular max-min composition is

$$\mu(x_i, z_k) = \max_{y_j \in Y} \left\{ \min\left\{ \mu(x_i, y_j), \mu(y_j, z_k) \right\} \right\} .$$

The operation is illustrated by a very simple example: Let $X = \{ x \}, Y = \{ y_1, y_2, y_3 \}, Z = \{ z \}$ and the membership degrees for the relation $P(X, Y)$ be $\{ 0.3, 0.5, 0.7 \}$, further on the membership degrees for $Q\{ Y, Z \}$ be $\{ 0.6, 0.4, 0.2 \}$, always in the increasing sequence of the subscripts of *y*. Then the result of the relational composition $R(X, Z) = P(X, Y) \circ Q(Y, Z)$ for the only existing pair of elements (x, z) will be $\max\{ \min\{ 0.3, 0.6 \}, \min\{ 0.5, 0.4 \}, \min\{ 0.7, 0.2 \} \} = 0.4$.

4. Fuzzy relations established by co-occurence and importance measures

In this section we introduce a way of establishing complex relations based on the absolute and relative simple and weighted word counts in documents, and parts of documents.

The basic hypothesis is that the frequency of occurrence of significant words in a certain document is connected with the importance of that word in the document. Another additional assumption will be that pairs of words occurring frequently in the same document or the same part of a document might be connected in the meaning (might be synonymous, antonymous, or otherwise related).

In the referred works [1], [2] attempts have been made to find ways to index documents automatically. The main point in this is the frequency of the words (in the whole document or in some parts of it, as it was summarized in Section 2). In [5] the concept of *fuzzy importance degree* (also referred to as "measure") was introduced. If

the [0,1]-normalized frequency of word w_i in the title/keyword section of document D_j is denoted by T_{ij} (keyword frequency, or title-keyword frequency), the normalized frequency of the same in the introduction/conclusion parts of the document is L_{ij} (location-keyword frequency), and the frequency in connection with cue words is C_{ij}, finally, if these three factors are weighted by $\lambda_1, \lambda_2, \lambda_3$ (where $\lambda_1 + \lambda_2 + \lambda_3 = 1$), by the following value the normalized fuzzy importance degree is calculated by the convex combination of the three frequencies:

$$F_{ij} = \lambda_1 T_{ij} + \lambda_2 C_{ij} + \lambda_3 L_{ij} .$$

(As a matter of course, any one or two of λ_i can be equal to 0.) Obviously, F_{ij} is a fuzzy membership degree that expresses the connection of w_i and D_j ($\mu(w_i, D_j)$). If the hierarchical structure of the document is taken into consideration as illustrated in Fig. 1, fuzzy importance degrees of level one (F_{ij} itself), level two, etc. can be introduced (${}^k F_{ij}^2 = \lambda_1 {}^k T_{ij}^2 + \lambda_2 {}^k C_{ij}^2 + \lambda_3 {}^k L_{ij}^2$, the right superscripts indicating that level 2 titles, i.e. sub-titles, level 2 introductions and conclusions, and to some extent, cue words located in some significant parts of the sub-sections were calculated; and the left superscripts referring to the index of the sub-document, i.e. meaning "part *k*" in this case, this latter extending to multiple component superscripts if necessary, e.g. *k,l* meaning "*k*th sub-section. *l*th sub-sub-section.)

Another way of expressing the importance of a word in the document is just calculating its normalized frequency in its whole text ($K_{ij} = \nu(w_i, D_j)$) that will be called fuzzy occurrence degree. As a matter of course, the frequency within any sub-section, sub-sub-section, etc. can be calculated, and so the frequencies ${}^k K_{ij}^2$, etc. can be determined.

From now on it will be assumed that both fuzzy importance degrees: the normalized keyword frequencies F_{ij}, etc. and the normalized word frequencies of (overall) occurrence K_{ij} obtained by automatic indexing of the relevant document and its sub-sections, etc. are available.

If the importance degree of each significant word in each document in a full or sample collection is available, the *fuzzy co-occurrence degrees* can be calculated. By co-occurrence the similarity or logical equivalence of the importance degrees or (normalized) relative frequencies will be understood. Fuzzy logical equivalence can be defined in various ways (all of these being extensions of the Boolean logical equivalence operation $A \equiv B \Leftrightarrow (A \wedge B) \vee (\neg A \wedge \neg B)$). In this study, two straightforward definitions of fuzzy equivalence will be used. The first one is based on the Zadeh-style fuzzy operators

$$\mu_{\neg A}(x) = 1 - \mu_A(x), \mu_{A \tilde{\wedge} B}(x) = \min\left(\mu_A(x), \mu_B(B)\right) \text{ and}$$
$$\mu_{A \tilde{\vee} B}(x) = \max\left(\mu_A(x), \mu_B(B)\right)$$

where $\tilde{\neg}$, $\tilde{\wedge}$ and $\tilde{\vee}$ stand for fuzzy negation, conjunction and disjunction, resp., and has the form

$$\mu_{A\cong B}(x) = \max\left\{ \min\left\{ \mu_A(x), \mu_B(x) \right\}, \min\left\{ 1 - \mu_A(x), 1 - \mu_B(x) \right\} \right\} .$$

The second one is based on the algebraic fuzzy operations (where the negation is identical with the above), being

$$\mu_{A \tilde{\wedge} B}(x) = \mu_A(x)\mu_B(x) \text{ and } \mu_{A \tilde{\vee} B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) .$$

(In the next, complicated denotations will be simplified such that the fuzzy logical operation will not be differentiated by the wave above the operator, as it is usually clear from the context if it is a fuzzy operation, further, membership functions will be usually denoted just by the symbol of the referred fuzzy set or statement, so e.g. the algebraic fuzzy disjunction being defined simply as $A \vee B = A + B - AB$.) For more details on fuzzy operators and operations see [3].

When introducing *hierarchical co-occurrence* the following method is meant: First the hierarchical structure of and the way of indexing the document in that particular model are determined. (Determine the number of levels in the document. Determine the weights λ_i . For each hierarchical level and within it, for each section, sub-section, etc. determine the text unit in question, and if necessary, its special location parts, like the introduction, etc.) Then for each text unit (including its title, etc.) determine the *fuzzy importance degree* and the *fuzzy occurrence degree* as well, and the *fuzzy equivalence of these two degrees* will result in the *hierarchical fuzzy co-occurrence degree* of the given document, section, etc. Its formal definition is as follows:

$$H_{i_1 i_2 j} \Leftrightarrow F_{i_1 j} \equiv K_{i_2 j}$$

for the main text, and

$${}^{k}H^{k}_{i_1 i_2 j} \Leftrightarrow {}^{k}F^{l}_{i_1 j} \equiv {}^{k}K^{l}_{i_2 j}$$

for sub-section number *k* in level *l* (assumedly *l*=2, here), all for keyword W_{i_1} and word w_{i_2} in document D_j .

As a matter of course, *non-hierarchical co-occurence of pairs of words* in the text can be calculated in a similar manner:

$$N_{i_1 i_2 j} \Leftrightarrow K_{i_1 j} \equiv K_{i_2 j} ,$$

this formula standing for words w_{i_1} and w_{i_2} in document D_j .

If a sample collection of documents is fixed e.g. for training the information retrieval system, the average degrees of hierarchical fuzzy co-occurrence can be calculated by

$$H_{ij} \equiv \frac{\sum_{k=1}^{n} H_{ijk}}{n},$$

where n is the number of documents in the sample collection, and i stands now for the subscript of the keyword, j for that of the general text word in question. Similarly, the average non-hierarchical co-occurrence degree can be defined by

$$N_{ij} \equiv \frac{\sum_{k=1}^{n} N_{ijk}}{n}$$

(using the subscripts in the same way as with H), and this index can be determined directly for keywords in the titles, special location parts, and cue word neighborhoods for any pair of keywords in the same manner as well (N_{ij}^{W}).

All fuzzy co-occurrence degrees defined here can for the bases of fuzzy relations describing the mutual relations of pairs of words in a collection of documents.

5. Direct queries by non-hierarchical and hierarchical co-occurrence of words

Let us explain the meaning of these degrees by a very simple example. The user is querying for the word “game”. Let us suppose that this is one of the keywords in the model. The simplest information retrieval system would just search for documents where this word occurs wherever in the text (including the title). Then probably most of documents will be left out that refer to some kind of particular game, and do not mention (frequently) the word “game” itself. It is obvious that information retrieval by direct occurrence of a word will usually be very restricted and will not satisfy most of the users.

If the non-hierarchical co-occurrence relation of the keywords is known, it can be examined what the most frequent words are that occur jointly with “game”, e.g. in titles only ($\lambda_1 = 1, \lambda_2 = \lambda_3 = 0$). Let us assume that N_{ij}^{W} will be maximal for the keywords “gamble”, “sport”, and “play”. Certainly, these are not synonyms to each other, however, all belong to one of the meanings of the original word. Consequently, a more refined search can be done for all documents that contain “game” itself or one of the latter three in the title. A similar search in the whole texts using the N_{ij} based full text co-occurrence relation might lead to another (assumedly wider) collection of words, and all documents indicated by high values in this relation graph can be retrieved as being of possible interest for the user.

If the hierarchical co-occurrence relation defined by H_{ij} is known as well, then words occurring frequently in documents that contain “game” in the title will be also indicated, as e.g. golf, cricket, baseball, football, poker, baccarat, Black Jack, chess, etc. By knowing the hierarchical connections, all documents can be retrieved that have a reference or have frequent reference to one or several of these words, even if they do not contain “game” in the title at all! It remains a problem however that golf and baccarat have little to do with each other and most of the users querying for

games will probably search for either various kinds of gambling or various kinds of sports, but not for both at the same time. So, a considerable part of the retrieved documents will be very likely useless for the user, and because of this, a secondary "manual indexing" will be necessary in order to select relevant documents out of the too large amount of potentially interesting documents.

In the next section a hierarchical relational map will be introduced that might enhance the effectiveness of queries, both in the sense of extending the search for documents that have no lexical but semantic coincidence with the queried word(s), and in the sense that words and phrases with too distant semantic relations to the queried word(s) will be excluded from the circle of retrieved documents by applying the tolerance classes established in the map.

6. Complex hierarchical relational map of document collections

By using the fuzzy importance and co-occurrence degrees, and the fuzzy relation classes discussed in the previous sections, it is possible to establish a complex hierarchical relational map of a sample document collection. In order to do that, it is necessary to decide the levels and weighting factors to be taken in consideration, and then do the keyword and general word counts in the whole collection. After having these values, all frequencies must be normalized for the unit interval [0,1], e.g. by mapping the highest keyword and general word counts in the collection to 1, and mapping all other proportionally:

$$I_{normalized,i} = \frac{I_{absolute,i}}{\max_{j=1}^{n} \left(I_{absolute,j} \right)}$$

where I denotes any keyword or word index in the sense of the former equations, and n is the number of documents in the sample. By this, the normalized indices can be interpreted as fuzzy membership degrees and can be used directly in the formulae given in Section 4. As a result, the following relations and corresponding graphs will be established:

- Keyword co-occurrence relation/graph G_W (established by the normalized co-occurrences N_{ij}^{W})
- Word co-occurrence relation/graph G_w (established by the normalized co-occurrences N_{ij})
- Fuzzy importance degree (keyword-document occurrence) relation/graph G_{WD} (established by the fuzzy importance degrees F_{ij})
- Word-document occurrence relation/graph G_{wD} (established by the normalized occurrences K_{ij}
- Hierarchical co-occurrence relation/graph G_{Ww} (established by the hierarchical co-occurrences H_{ij})
- Further hierarchical co-occurrence relations for multilevel models

In Fig. 8 the structure of these relations can be seen for two hierarchical levels.

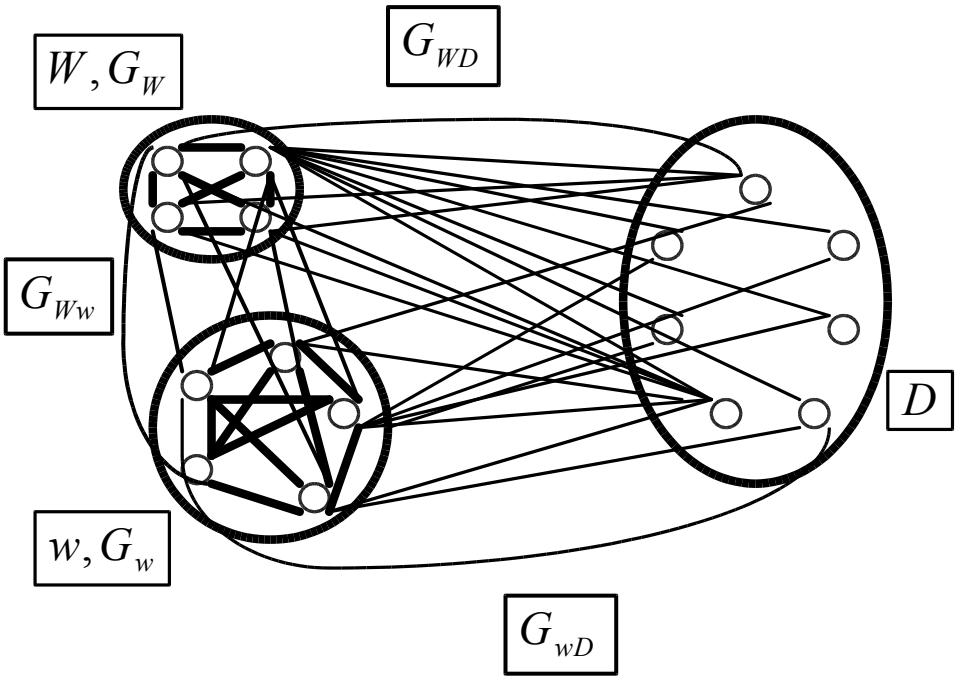

Figure 8.

There are three sets of nodes: the set of documents *D*, the set of keywords *W* and the set of words *w*. (It has to be mentioned that in practice, it is reasonable to assume that $W \subset w$.) There is no relation established among the elements of *D*, even though it could be reasonable to find the degree of similarity or tolerance between pairs of documents however, it is supposed that the number of documents even in the sample collection is rather high (e.g. several thousand), and so, the number of pairs would be even higher (in the order of several millions). There is a relation over the elements of *W*, represented by G_W, where the membership degrees are defined by $\mu(W_i, W_j) = N_{ij}^W$; and there is another relation over *w*, represented by G_w, where $\mu(w_i, w_j) = N_{ij}$.

There is the bipartite graph G_{Ww} over $W \times w$, where $\mu(W_i, w_j) = H_{ij}$ expressing the hierarchical co-occurence of keyword-general word pairs.

Finally, there are two more bipartite graphs representing the importance degree and frequency of occurrence of keywords and words, resp., G_{WD} over $W \times D$, where

$$\mu(W_i, D_j) = F_{ij}, \text{ and } G_{wD} \text{ over } w \times D, \text{ where } \mu(w_i, D_j) = K_{ij}.$$

The bipartite graphs represent also mappings in the following sense:

$$G_{Ww}: W \to w, G_{WD}: W \to D, G_{wD}: w \to D.$$

The image of every keyword W_i is a fuzzy set of words in *w*, and also a fuzzy set of documents in *D*, where by knowing the membership degrees attached to every pair, the degree of belonging to the set is defined by the degree of the relation between them. E.g.

$$C_D(W_i) = G_{WD}(W_i) = \langle D, \mu_{W_i}(D_j) \rangle, \mu_{W_i}(D_j) = F_{ij} .$$

Also the image of every word w_i in *w* is a fuzzy set in *D*, defined by G_{wD}.

If hierarchical search is done, the starting item is always a keyword. As G_{Ww} is a relation from *W* to *w*, and G_{wD} is one from *w* to *D*, there is another way of mapping the keywords to the documents, by applying relational composition $G_{Ww} \circ G_{wD}$ that will be shortly denoted by G°_{WD} to differentiate it from the direct relation G_{WD}.

As fuzzy relations indicate the degree of membership (e.g. in a relation), it is usually advisable to set a threshold value τ between 0 and 1, and considered are all matches that are at least equal to τ. If it is necessary, $\tau' < \tau$ should be chosen to extend the circle of retrieved documents. If the relation is at least as strong as the chosen threshold, it will be called *matching*.

By having the above relational map the following search methods can be proposed:

Method 1. (*Search by keyword occurrence*) If given is the keyword W_i, all documents matching this keyword will be retrieved. $\Delta = G_{WD\tau}(W_i)$. (Δ denotes the set of documents retrieved, the subscript refers to the τ-cut of the relation.) For an illustration see Fig. 9.

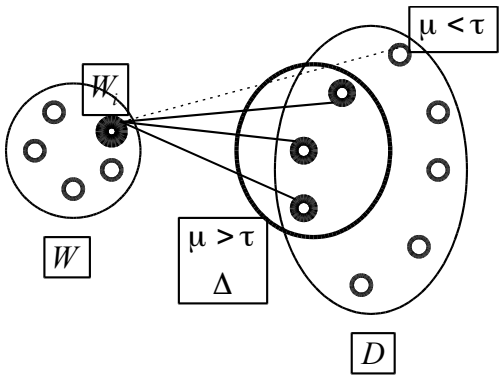

Figure 9.

In the figure the queried keyword is indicated by a dark node. All matching documents in the collection (thick line nodes in *D*) are connected to it by solid lines, while a document having less membership than the threshold in relation G_{WD} is shown by dashed line connection. This latter is not considered to be matching and is left out of the class of retrieved documents Δ.

Method 2. (*Search by keyword and hierarchical co-occurence*) Determine the set of words that match the keyword. All documents that match any of the matching words will be retrieved. $\Delta = G_{wD\tau_2}(G_{Ww\tau_1}(W_i))$. ($\tau_1$ and τ_2 might be different or identical thresholds determining the level of matching.) The method is illustrated in Fig. 10.

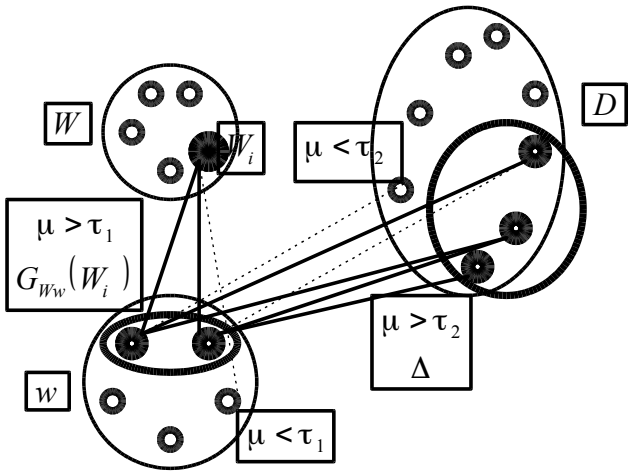

Figure 10.

Denotations are similar as in the previous example, the class of matching words in *w* is indicated by thick line nodes and solid lines show membership in G_{Ww} over threshold τ_1, while the dashed line goes to a word below this value. In *D* all documents are included in class Δ where there is a relation at least as strong as τ_2 with at least one of the matching words. (Membership in Δ is defined by

$$\mu_\Delta(D_j) = \max_{w_k}\left\{ G_{wD}(w_k, D_j | w_k \in C_{\tau_1}(W_i))\right\} .)$$

Method 3. (*Search by keyword compatibility/equivalence relations and occurrence*) Determine the compatibility or equivalence class of the given keyword in *W* for given threshold σ. This is denoted by $C_{W\sigma}(W_i)$. Search all documents matching the compatibility class of the original keyword. $\Delta = G_{WD\tau}(C_{W\sigma}(W_i))$.

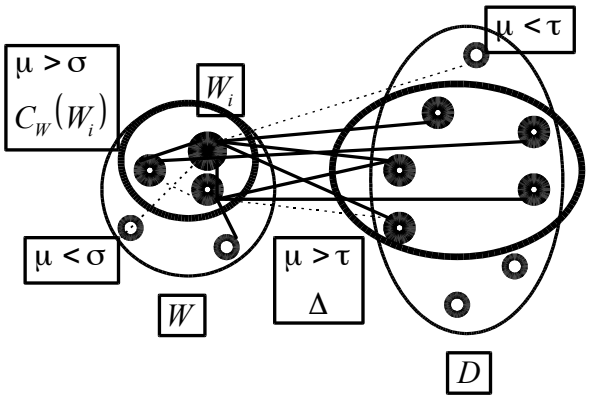

Figure 11.

The figure presents the compatibility class belonging to the queried word. As tolerance is not transitive, elements of the class are not necessarily connected by membership above threshold σ, and elements of *W* connected above threshold to

elements of the compatibility class do not belong themselves to the class. Keywords connected to the queried word with less membership than the threshold do not belong to the class. All documents that are connected with at least one of the keywords in the compatibility class of the queried word stronger than τ are included into Δ.

Method 4. (*Search by keyword compatibility/equivalence relations and hierarchical co-occurrence*) Determine the compatibility class in *W* and all matching words in *w*. All documents matching the image of the compatibility class of the original keyword will be retrieved. $\Delta = G_{wD\tau_2}\left(G_{Ww\tau_1}\left(C_{W\sigma}(W_1)\right)\right)$.

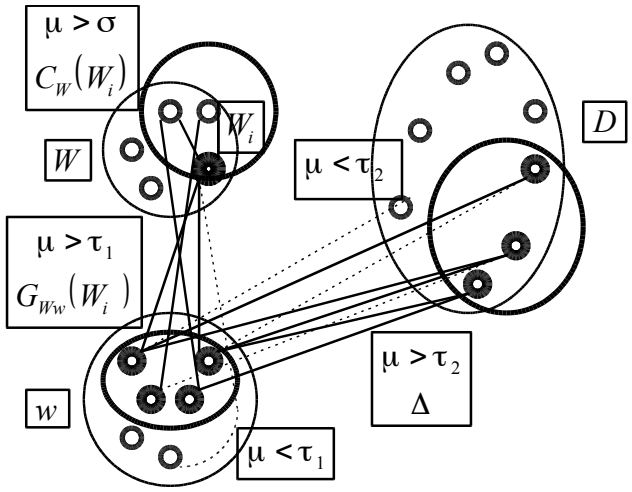

Figure 12.

In Part I of this study methods were discussed where the user starts their query by a single keyword. Based on similar definitions and mathematical tools it is possible to extend the approach to queries where a group of keywords or words is determined at the beginning, and so the system will establish the relevant classes in *W* and *w* by tolerance or equivalence relations.

References

[1] A. H. H. Ngu, T. D. Gedeon and J. Shepherd: Discovering indexing parameters for information filtering, Proceedings of 2nd International Conference on Intelligent Systems, Singapore, 1994, pp. 195-200.

[2] R. A. Bustos and T. D. Gedeon: Learning synonyms and related concepts in document collections, in: J. Alspector, R. Goodman and T. X. Brown (eds.): Applications of Neural Networks to telecommunications 2, Lawrence Erlbaum, 1995, pp. 202-209.

[3] G. Klir and T. Folger: Fuzzy Sets, Uncertainty and Information, Prentice Hall, Englewood Cliffs, NJ, 1988.

[4] M. R. Garey and D. S. Johnson: Computers and Intractability. A Guide to the Theory of NP-Completeness, W. H. Freeman and Co., San Francisco, 1979.

[5] T. D. Gedeon, S. Singh, L. T. Kóczy and R. A. Bustos: Fuzzy relevance values for information retrieval and hypertext link generation, Proceedings of EUFIT ’96, Aachen, 1996, pp. 826-830.