A simple heuristic method is proposed to use an existing thesauri for weighted indexing.

INTRODUCTION

Since 1976 [1] people have tried to use the idea of fuzzy set theory in information retrieval. Although a lot of theoretical papers and reviews had appeared on this subject [1], [2], [3], [4], [5], [6], very little use of the idea was made in actual implementations. So it seemed that fuzzy set theory was a dead issue in information retrieval. However, good ideas never die, they just come back in a different form. In the case of fuzzy set theory, the idea was incorporated in knowledge-assisted document retrieval processes [7], [8]. So, fuzzy sets became a part of the artificial intelligence stream which now - a -days pervades every scientific and technological enterprise [9], [10]. Although everybody is not convinced of the utility of knowledge structures in information retrieval [11], it may be confidently stated that in the near future, a lot of effort will be put into the application of artificial intelligence systems for indexing and retrieval purposes. In this paper, a heuristic approach is proposed based on the fuzzy set theory to use an existing thesaurus for weighted indexing. Retrieval, then, can be done by using whatever proper method.

The use of thesauri in weighted retrieval has already been studied by Reisinger [12], Redecki [13], Mazur [14] among others. Miyamoto et al. [15] have experimented it with a computer-generated thesaurus. These investigations, however, seem needlessly complicated, partly because of the interweaving of indexing and retrieval phase and partly because of too restricting mathematical requirements. More realistic is the approach taken by Sommar and Dennis [16]. They designed a weighted term information retrieval system, which has been in effective use by Celanese Fibres Co. Here indexing is done by using a thesaurus (without using weights) but weighted terms specified in the search strategy are expanded to include their narrower terms. These are selected from the thesaurus by computer. Their system is a good example of an unsophisticated, but practical use of a thesaurus.

In this paper, an elaborate and simple approach is proposed employing the use of an existing (unweighted) thesaurus and considering only the indexing phase. This has, among other things, the advantage that no new thesaurus need be constructed.

DESCRIPTION OF A THESAURUS

A thesaurus consists of two sets : a set of signs which stand for concepts, called the descriptors of the thesaurus and a set of relations between these signs. There exists moreover a bijective correspondence between the descriptors and the concepts and also a bijective correspondence of the relations between descriptors and the relations between concepts. So a thesaurus may be described as a relational system [1]. In this section most of the examples of thesaurus structure are taken from the Medical Subject Headings (MeSH). 1985, the thesaurus used in Index Medicus.

Following procedure may be adopted to obtain a bijective correspondence between the descriptors and the concepts:

Homographs should be removed by
marking the descriptors accordingly, e.g. helix (snails) and helix (part of the ear).

Synonymy may also be defined properly. Synonyms are terms with meanings so closely related that the terms are regarded as interchangeable.

Quasi-synonyms are terms of which the meanings are generally regarded as different for ordinary usage, but are treated as synonyms for indexing purposes.

Most of the thesauri solve the problem of synonymy by distinguishing a preferred synonym. This is indicated by USE, or SEE. All other relevant synonymous or quasi-synonymous descriptors are indicated by UF = or X.

Example: Broca area SEE frontal lobe: frontal lobe X Broca area.

To achieve a one-to-one correspondence of the relations between descriptors and the relations between concepts, hierarchical relations, are defined. The hierarchical relationship has a converse, and the terms that share it are known as broader terms and narrower terms. BT (or < ) is the symbol for a broader term and NT (or >) for a narrower term. There are two kinds of hierarchical relations: the generic relation, which is the link between a class and its members and the hierarchical whole-part relationship (partitive relation) which implies the link between the name of the part and the whole.

Examples:

The term “face” is broader than “chin”, but narrower than “head” (partitive relation);
The term “beetles” is broader than “tenebrio”, but narrower than “insects” (generic relation).

It may also be noted that BT and NT are defined only on the set of preferred terms.

According to Reisinger [12] the narrower terms of the same broader term are called related terms, denoted by RT, or “see related”, but in most of the existing thesauri the relation RT is often used for all kinds of terms that are conceptually associated with each other.

Example: “breast” see related: “mastectomy”.

Synonymy is an equivalence relation. It may be reflexive, symmetric or transitive. The preferred term can be considered as the representative for the class of all synonymous terms. The hierarchical relation on the other hand is irreflexive and asymmetrical. BT or NT does not only denote the immediate broader term or narrower term respectively but is also transitive and therefore it is a strict partial order. The generated relation RT is symmetric and transitive as per the Reisinger’s definition, otherwise RT is only symmetrical.

As BT is a strict partial order in a finite set, it can be considered as a finite graph, denoted by T. On this graph we can define a distance function $d_T : T \times T \rightarrow N \cup \{\infty\}$ (the set of nonnegative integers and infinity) by setting $d_T(x,y)$ equal to the number of edges in a shortest path between x and y; $(x,y) \in T \times T$. If $x$ and $y$ can not be joined then $d_T(x,y) = \infty$.

A GENERALIZED SYNONYM FUNCTION ON A THESAURUS

A function $w : T \times T \rightarrow [0,1]$ is defined to indicate how closely the terms of the thesaurus are related. If $x$ and $y$ are different preferred terms, then

$$w(x,y) = 1 - \frac{2}{\pi} \arctg (d_T(x,y));$$

This gives the following table of values:

<table>
<thead>
<tr>
<th>$d_T(x,y)$</th>
<th>$w(x,y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>0.295</td>
</tr>
<tr>
<td>3</td>
<td>0.205</td>
</tr>
<tr>
<td>4</td>
<td>0.156</td>
</tr>
<tr>
<td>5</td>
<td>0.126</td>
</tr>
<tr>
<td>6</td>
<td>0.105</td>
</tr>
<tr>
<td>7</td>
<td>0.090</td>
</tr>
<tr>
<td>8</td>
<td>0.079</td>
</tr>
<tr>
<td>9</td>
<td>0.070</td>
</tr>
<tr>
<td>10</td>
<td>0.063</td>
</tr>
<tr>
<td>$\infty$</td>
<td>0</td>
</tr>
</tbody>
</table>

For each preferred term $x_i$ a graph $G_i$ is
formed which indicates its closeness with its equivalent terms: the descriptors which are interchangeable in every context. In actual thesauri all quasi-synonymous terms are at a distance "one" from the preferred term, but it is suggested to expand this relation so as to obtain a finer structure. If xj and yj are synonymous terms, the following equation may represent the relationship between them:

\[ W(x_j, y_j) = 1 - \frac{0.8}{\pi} \arctg \left( \frac{2}{3} \right) \]

The numbers 0.8 and 3 are chosen to yield reasonable results in an actual artificial intelligence environment, these numbers can be subjected to a learning (feedback) procedure. The above proposal yields:

<table>
<thead>
<tr>
<th>Dgj (Xj, Yj)</th>
<th>w (Xj, Yj)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.918</td>
</tr>
<tr>
<td>2</td>
<td>0.850</td>
</tr>
<tr>
<td>3</td>
<td>0.800</td>
</tr>
<tr>
<td>4</td>
<td>0.764</td>
</tr>
<tr>
<td>5</td>
<td>0.737</td>
</tr>
<tr>
<td>6</td>
<td>0.718</td>
</tr>
<tr>
<td>7</td>
<td>0.703</td>
</tr>
<tr>
<td>8</td>
<td>0.691</td>
</tr>
</tbody>
</table>

It may be noted that w (Xj, Yj) is always > 0.6.

Finally, if x and y are non-synonymous non-preferred terms, then w (x, y) = w (xj, yj), where xj is the preferred term for x and yj is the preferred term for y.

It is obvious that the function w as defined above is reflexive when w (x, x) = 0 and symmetric when w (x, y) = w (y, x), but it does not satisfy the triangular inequality.

**USE OF THE GENERALIZED SYNONYMY RELATION FOR WEIGHTED INDEXING**

Weighted indexing can be a heavy task indeed. If D is the set of documents, then weighted indexing consists in associating with every \( d \in D \) (every document of the collection) a function \( \mu_d \) from T to the interval \([0, 1]\):

\[ \mu_d : T \rightarrow [0, 1] : t \rightarrow \mu_d (t) \]

The number \( \mu_d (t) \) is called the index weight of the term t in d. Whether this index weight is based on relevance, weighting, [17], or a fuzzy membership value [1] or is obtained in still another way, is immaterial for indexing purposes, although it may be noted that it coincides probably most easily with the fuzzy set approach. Therefore, using the generalized synonymy function, indexing can be done in a very thorough and yet simple way. It is required to know only the list of descriptors. Having assigned a few terms and index weights with regard to a particular document:

\[ \mu_d (tk) : k = 1, \ldots, n \text{ for every } t \in T : \]

\[ \mu_d (t) = \max (w (t, tk)) \]

The actual calculation of every \( \mu_d (t) \) is, of course, done with the help of computer.

**CONCLUSION**

The method of indexing described in this paper has the following advantages:

1) Indexing is complete because a strict positive weight is assigned for every descriptor in every document.

2) The indexer need not bother about preferred terms. On the contrary he might in some cases consider a quasi-synonym to be the best.

3) Related terms get index weight which do not differ much.

4) The weights given by the indexer remain usually unchanged. Changes can only occur if the indexer has used a large number of related terms (for instance quasi-synonyms) with different weights. In this sense one can say that the system has some "intelligence", for it corrects logical errors.

It has some disadvantages also, the most obvious one being the requirement of
enormous amount of space necessary to store every index weight. If necessary, this, however, can easily be reduced by using a threshold value: index weights which fall below the threshold value may be considered zero. This threshold should be chosen small enough so that indexing remains reasonably complete.

ACKNOWLEDGEMENT

The authors are thankful to Mr. M. Dekeyser and Mr. L. Egghe for the help they have provided during the preparation of the paper.

REFERENCES

1. Tahani V: A fuzzy model of document retrieval systems. Inf Processing Mgmt, 1976, 12, 177-188.


