Fuzzy relational thesauri in information retrieval: automatic knowledge base expansion by means of classified textual data

Domonkos Tikk[1,2,3], Jae Dong Yang[3], Péter Baranyi[1,2], and Anikó Szakál[4]

[1] Dept. of Telecommunications & Telematics, Budapest University of Technology and Economics, 1117 Budapest, Magyar Tudósok Körútja 2., Hungary, e-mail: tikk@ttt.bme.hu

[2] Intelligent Integrated Systems Japanese–Hungarian Laboratory 1111 Budapest, Műegyetem rakpart 3., Hungary, e-mail: baranyi@alpha.ttt.bme.hu

[3] Department of Computer Science, Chonbuk National University Chonju 561–756, Korea, e-mail: jdyang@cs.chonbuk.ac.kr

[4] Department of Information Technology, Bánki Donát Polytechnic H-1081 Budapest, Népszínház utca 8., Hungary, e-mail szakal@zeus.banki.hu

Abstract—**In our ongoing project we develop a tool which provides domain engineers with a facility to create fuzzy relational thesauri (FRT) describing subject domains. The created fuzzy relational thesauri can be used as knowledge base for an intelligent information agent when answering user queries relevant to the described domains, or for textual searching on the web. However, the manual creation of (fuzzy) thesauri is quite tedious process if the source of data from which the domain engineer may select concepts and instances for the thesaurus is not well organized or structured. That is the typical case of textual data bases. In order to ease FRT creation process we make use of a small starting FRT and our text categorization technique that temporarily expands FRT while doing the supervised learning part of text categorization. This by-product of categorization is then used for enlarging automatically or semi-automatically the final FRT.**

I. Introduction

Information agent technology emerged as a major field of interest among practitioners and researchers of various fields with the immense grow of available information on the internet since the early 90s. The vast amount of heterogeneous information source available on the internet demands advanced solutions for acquiring, mediating, and maintaining relevant information on behalf of users or other agents. In general, intelligent information agents are autonomous computational software entities that are developed to provide pro-active resource discovery, create a link between information consumers and providers, and offer value-added information services and products [1]. The agents are assumed to deal with difficulty associated with the enormous amount of information overload of users, preferably in real-time.

Research supported by the National Scientific Research Foundation (OTKA) Grants No. D034614 and T34212, and by the Hungarian Ministry of Education Grant No. FKFP 0180/2001. A part of this work was done while D. Tikk was visiting at Department of Computer Science, Chonbuk National University.

There are different approaches to build information agents including user programming, machine learning, and knowledge engineering. When using *knowledge engineering* for the construction, the agent is set out a large deal of domain specific knowledge about the application and the user. However, a significant drawback of this method is that it requires substantial efforts from the knowledge engineer to build up and maintain knowledge bases. Moreover, the agent is highly domain-specific and its knowledge is relatively fixed [1]. In order to facilitate the construction and maintenance of the agent's knowledge our project aims to assist domain experts (and possible users) with a useful tool. This tool helps to create *fuzzy relational thesauri* which describe the subject domain by means of its major concepts/instances and the various kind of relationship between these concepts/instances. In our approach the concepts in the fuzzy relational thesauri are represented by keywords of the subject domain.

The build-up of a fuzzy relational thesaurus (FRT) is considerably tedious process when the knowledge engineer has to take into account all the important concepts (in our approach: keywords) of the actual domain, together with their possible synonyms and associated words. In order to facilitate this procedure we retrieve the keywords from a sample text archive, and then apply text categorization in this text retrieval method. he text archive can be the collection of documents selected by a domain expert, or obtained as the result of web searches. The next step is the categorization of the archive by means of our FRT based hierarchical (or structured) text

categorization method [2]. This approach requires

- a small starting FRT typically consisting of the major concepts of the subject domain;
- labelling the documents for the supervised learning with category names corresponding to the concepts of the starting FRT;
- one-to-one correspondence between category names and FRT terms.

he categorization process uses FRT as the implementation of the hierarchical category system (often called taxonomy), i.e. the categories are represented by terms (concepts) in FRT. In order to prevent FRT from being incoherent and far from the view of its creator, categorization is proceeded on a copy of the starting FRT. The approach assigns to every category a set of terms, so-called local dictionaries, which contains words being typical for the category in the text corpus and are stored in FRT. From the aspect of categorization local dictionaries are useful to discriminate documents belonging to different categories. From the aspect of FRT creation, local dictionaries, being in fact a by-product of the categorization approach, collect the most frequent words of categories and so describe them efficiently. The size, contents and the preparation technique of the local dictionaries vary depending on various parameters of the categorization.

For the final expansion of the starting FRT, elements of local dictionaries can be inserted permanently to FRT. We provide an option for the knowledge engineer (maintaining the FRT) to supervise, filter and modify elements of local dictionaries before added to the thesaurus. In such a way the FRT is maintained in accordance with the knowledge of the domain expert and reflects his/her view on the subject domain the best.

This paper is organized as follows. Section II describes the role of FRT in information retrieval systems. Section III describes the FRT management system developed we utilize in next. Section IV provides a brief description of our FRT based text categorization approach with special emphasize on the creation of local dictionaries. In section V we present the utilization of the local dictionaries, and finally section VI summarizes the paper.

II. Fuzzy relational thesaurus in information retrieval

A thesaurus in an information retrieval system (IRS) can be considered as a *knowledge base* that represents the conceptual model of certain subject domain [3]. The thesaurus has similar role in an IRS as a knowledge base in expert systems. In IRS's the query may be seen as a concept, and documents as objects. An object is in the answer to the query to the degree it is an instance of the concept. With the above terminology, a document satisfies a query to the degree to which it is classified as that query.

In what follows we assume that the knowledge base represented by the thesaurus is in term-centered form which is maintained by the domain expert. A fuzzy relational thesaurus (FRT) [3], is a particular representation form, which is an extension of traditional thesauri schemes enriched by the strength of the term relationships and a richer set of binary relations. In FRT every term relation has a strength represented by a number of the unit interval. Formally:

Definition II.1 [3] *Let X be the set of all terms, and $R(X, X)$ is an arbitrary fuzzy term relation. Then for an ordered pair of terms, $(x, y) \in X^2$, $R(x, y) \in [0, 1]$ is the belief (or certainty) that x implies y. This strength can be considered as the membership degree of (x, y) in the fuzzy term relation $R(X, X)$.*

Note that since (x, y) is an ordered pair, the symmetry of the relation is not a necessary condition. The term relation may comprise the following possibilities:

- General relations: RT – related term; ST – synonym term; BT – broader term; NT – narrower term.
- Abstraction relations: KO – kind of (specialization of, subclass of); IO – instance of; FO – has feature object (e.g. $FO(\mathrm{IA}, \mathrm{CM})$) where IA = information agent, CM = construction method.
- Domain specific relations of the form $fo(x, y)$, where fo is the name of the feature object associated with x, and y is the concepts fo-feature object of y, e.g. $CM(\mathrm{IA}_1, \text{knowledge engineering})$.

These relations are organized in a partial ordering based on their specificity (see Figure 1). The ordering is exploited in the construction of the FRT and in query answering. The strength of actual term relations is determined by the domain engineer, but a default value is assigned for each relation type in order to ease and reduce the engineer's work (for details see section III).

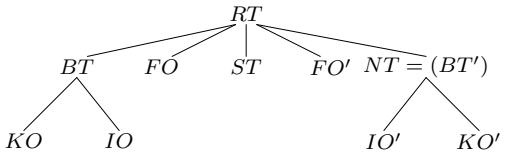

Fig. 1. Partial ordering of relations. $'$ means inverse. The lower a relation in the tree, the more specific it is (redrawn from [3]).

The answer to a query is determined by the relation of the query terms and the document terms. In order to calculate the strength of relation between an arbitrary pair of terms, the transitive closure [4] of the term relations are computed using the partial ordering of the relations depicted on Figure 1. The transitive closure is computed with respect to a specified t-norm, e.g. the minimum or the algebraic product. In [3] the authors proposed an effective way for this computation, with time complexity

of order $O(m \log m)$, where m is the pairs of terms being in relation in the knowledge base.

III. Project description

In [3] the authors assumed that the domain expert is assisted by a software tool, a kind of "FRT management system" which help him in the construction task. In the next we describe one of our ongoing project which aims at developing this software. Because of the lack of space for this description is relatively brief. For further details we refer to our paper [5], which is available also on the internet.

We propose a new approach for managing domain specific thesauri, where object-oriented paradigm is applied to thesaurus construction and query-based browsing. This approach provides an object-oriented mechanism to assist domain experts in constructing thesauri; it determines a considerable part of relationship degrees between terms by inheritance and supplies domain experts with information available from a thesaurus being constructed. It enables domain experts to incrementally construct the thesaurus as well, since the automatically determined degree of relationships can be refined whenever a more sophisticated thesaurus is needed. It may minimize domain expert's burden caused from the exhaustive specification of individual relationship. This approach also provides a query-based browsing, which allows users to easily verify thesaurus terms before they are used in usual boolean queries.

Our thesaurus is called object-based thesaurus. All its relationships are represented in terms of two levels: concept level and instance level. The former defines the relationships between concepts, and the latter specifies the relationships between instances. The relationships in the object-oriented paradigm are employed to directly implement the object-based thesaurus, providing a formal specification in the construction of the thesaurus. For example, BT/NT relationship is redefined as a generalization hierarchy, while a vague relationship, RT is refined into aggregation and association which have more concrete semantics. The aggregation and association of a concept acts here as a property in the generalization hierarchy that its sub-concept inherits. Such an inheritance of the relationship turns out to be a useful mechanism to structurally specify semantics between concepts.

We assign a default strength to each relationship type. For example we assign 0.7 as to every association relation, 0.8 to every part-of relation, and 0.9 to every sub-concept-of relation. We worked out a mechanism to calculate the strength of a relationship between any arbitrary pair of concepts based on object-oriented paradigm. We also have a technique to resolve multiple inheritance situation. For more details we refer to [5].

IV. Text categorization for fuzzy thesaurus creation

Without any facility for preprocessing the domain engineer should create the domain descriptive FRT based on raw textual data and/or his/her experience. Our goal is to facilitate this work thereby providing categorized textual data and the set of frequent keywords assigned to each category.

Automated text categorization is a thriving research topic in the area of information retrieval. The objective is to assign a document to an appropriate category selected from a finite set. This task was traditionally made manually by domain expert, but in last decades the exponential increase of online available textual data various stimulated developments of automated approaches. For a survey see [6]. These methods are usually based on a probabilistic model or various kind of learning strategies. For learning purpose a set of training documents with manually assigned class labels is assumed to be available.

As we mentioned earlier, our FRT based approach is applicable if categories are organized in topic hierarchy, also called *taxonomy*. Document categorization in taxonomy is a recently emerged field of text mining (see also [7], [8], [9]. On the internet topics are often organized in hierarchy (see, e.g., `www.yahoo.com`, `www.ibm.com/patents`), because it improves the perspicuity of the subject domain if the number of categories at a level is kept under a certain threshold. This is also the principle when inserting e-mails in a prescribed folder system.

Now we describe shortly our FRT based text categorization method [2]. The FRT is basically the implementation of the taxonomy which stores and maintains adaptive local dictionary for each category. For categorization the type relationship is uniform, but their strength has importance when learning. In graph theory, term hierarchy of FRT and taxonomy can be represented by digraphs. An FRT can be applied to categorize documents belonging to the topic hierarchy of a taxonomy, if there exist a subgraph of the FRT which is isomorphic with the digraph of the taxonomy. Therefore each category is represented by an FRT term.

We use the most common representation framework, the *vector space* model, where a document D is described by a vector of word frequency values, $\mathbf{W}_D$, and vector of FRT terms appearing in the document $\mathbf{T}_D$, respectively:

$$\mathbf{W}_D = (w_1^D, \cdots, w_p^D); \qquad \mathbf{T}_D = (t_1^D, \ldots, t_q^D), \qquad (1)$$

where p is the total number of unique terms appearing in the document collection, and q is the number of terms in the FRT.

The set of unique terms is determined automatically by removing the stopwords from the total term set (like e.g. "the" in English) and then applying stemming on

the remaining set. There are numerous possible weighting schemes in the literature to determine the values of weights w_i^D and t_i^D. The best and most sophisticated method is the entropy weighting, which was found to outmatch 6 others in [10], but perhaps the most popular is the tf×idf weighting [11] and its versions, which defines w_i^D in proportion to the number of occurrence of the term i in the document, f_i^D, and in inverse proportion to the number of documents in the collection for which the term occurs at least once:

$$w_i^D = f_i^D \cdot \log\left(\frac{N}{n_i}\right).$$

For t_j, we use the number of appearance of jth FRT-term in the document D. The vectors in (1) are normalized before any further processing is done.

We remark that the available raw text can also be processed by other approaches for FRT refinement. The set up of association relationship can be supported by measuring the co-occurrence of keywords in training documents [12]. This may overcome the creation of erroneous relations between term and also helps to discover the synonyms terms.

The categorization is done in iteratively, alternating two phases. First, the vector $\mathbf{T_D}$ is used to infer the category (or categories) of D (see [2]). In the beginning of the categorization process FRT contains only few terms, therefore the categorization is usually not efficient. In order to endow FRT with good categorizing ability, in the second (learning) phase terms being typical for the category/ies of D are added to the FRT. These two phases are proceed alternatively, while the performance of categorization achieves a certain level or cannot be improved significantly.

For the purpose of FRT expansion one of the most important issue of the categorization is the determination of terms assigned to a category of a document, called local dictionary $L_D(C)$, inserted to FRT in the learning phase. The selection employs the K-nearest neighbour technique. First, the K nearest neighbours of D is selected in the sense of a similarity measure (usually cosine). Let then J denote the index set of documents among the neighbours of D being classified also in C, and containing common words with C. $L_D(C)$ is an importance ordered set of words appearing in D and in the documents of J, D_J. The importance value of the ith word is determined as the sum of frequency value in D and in D_J: $\sum_{j\in D_J\cup D} w_i^j$. The size of $L_D(C)$ can be controlled by two parameters, p_1 the maximum number of words, and α_1 the minimum (cumulated) frequency value of a word ($p_1 \in \mathbf{N}$, $\alpha_1 \in [0,1]$). Thus, the final $L_D(C)$ is an ordered set with cardinality at most $|p_1|$, where every word has at least α_1 weight.

The performance of the categorization focusing on a single category is usually measured by two quantities: recall and precision. Let a and b be the number of documents correctly or incorrectly assigned to the given category, respectively, while c be the number of documents incorrectly rejected from the given category. With this terminology:

$$\text{recall} = \frac{a}{a+c}; \quad \text{precision} = \frac{a}{a+b}$$

Usually only one of these measures does not give realistic result about the performance, therefore the F-measure, their combination is usually applied:

$$F_\beta = \frac{(\beta^2+1)\text{precision}\cdot\text{recall}}{(\beta^2\cdot\text{precision})+\text{recall}}$$

where the parameter β is commonly set to 1.

The classifier can commit two kinds of mistake. It can incorrectly assign a category to a document and hence reduce the *precision*, or it can reject incorrectly a category and hence degrade the *recall* of the classifier. The performance of the categorization depends on the above two quantities to be maximized. In the first case our approach decreases the relationship wait along the *topic path*, i.e. between the erroneously selected category C and the terms in $\mathbf{T_D}$ which supported C. In the latter case new terms are inserted to FRT in order to enhance the classification performance. The insertion is proceeded in such a way that the new terms are forced to be as deep in the taxonomy as possible. This is because a term deep in the hierarchy supports all the categories along the path between the term and the root. For more details see [2].

V. Expansion of the FRT

The text categorization helps the knowledge engineers to create an FRT describing the subject domain efficiently in two ways. First, local dictionaries are provided to each category. The user is offered options to filter the words, modify the weight and to specify the kind of relationship according to his/her view about the subject domain. Second, new documents can be classified and thus new data are provided for further investigation and consideration.

We illustrate the described method on a simple example. We collected documents in the domain of *electronic appliances* from the web. The created a 211 and a 327 element document sets. We proceeded tests on both document sets. The unique term set (p in Eq. (1)) consisted of 3731, and 5793 words, respectively. The FRT was created by a semi-automatic thesaurus construction software [5]. In our application the depth of the taxonomy is three. The collected documents were classified into the following six top level topic: Audio, Computer, Computer Components, Computer Peripheral Device, House-Hold Appliances, Office Communications Appliances. We had 30 (31) 2nd level, and 40 (58) 3rd level topics in the case of 211 (327) document set. Each category had at least two training examples.

TABLE I

TOTAL NUMBER OF INSERTED WORDS IN TERMS OF p_1, α_1 AND THE NUMBER OF ITERATION it

327 element document set				211 element document set			
P_1	α_1	inserted words	it	P_1	α_1	inserted words	it
5	0.1	1069∼1198	10	7	0.05	766∼769	12
5	0.25	630	3	5	0.1	708∼723	10
5	0.2	796	15	10	0.02	860	10
7	0.05	1139∼1204	12	7	0.05	1139∼1204	12

The size of the total number of inserted terms depends on the parameter p_1 and α_1. The content of the local dictionaries can be modified if the stemming is omitted, or if other preprocession is performed on the corpus. The classifier applies different methods assigning categories which may also influence the size and contents of the local dictionaries. Table I shows the total number of inserted words in terms of the above parameters.

The starting FRT contains 584, and 1779 terms in case of 211 and 327 element document set, resp. The total number of inserted words is comparable with these quantities. When stemming is proceeded in the preparation phase, a list of original versions is also provided to the user as synonyms of stemmed words. This option allows the user to select the best possible term to be inserted. The presented facility greatly reduces the required time for FRT creation even at the final insertion the user is prompted at each word.

VI. CONCLUSION

We develop a fuzzy relational thesaurus management system. The aim of the system is to create a knowledge base for intelligent agent. FRT creation is a tedious process, which may be ease if the majority of data for the construction is supplied in a structured form. Therefore, we provide categorized textual data and local dictionaries assigned to categories to help the knowledge engineer in the build-up.

The information is retrieved by means of an FRT based text categorization approach, that classifies documents into hierarchical topic taxonomies. The categorization process creates local dictionaries as a by-product while improving its classification performance.

REFERENCES

[1] M. Klusch, "Information agent technology for the Internet: A survey," *Data & Knowledge Engineering*, vol. 36, pp. 337–372, 2001.

[2] D. Tikk, J. D. Yang, and S. L. Bang, "Text categorization using fuzzy relational thesauri," Submitted to *IEEE Trans. on Knowledge and Data Engineering*.

[3] H. L. Larsen and R. R. Yager, "The use of fuzzy relational thesaurus for classificatory problem solving in information retrieval and expert systems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 1, 1993.

[4] L. A. Zadeh, "Similarity relations and fuzzy orderings," *Inform. Sci.*, pp. 171–200, 1971.

[5] J. H. Choi, J. J. Park, J. D. Yang, and D. K. Lee, "An object-based approach to managing domain specific thesauri: semiautomatic thesaurus construction and query-based browsing," Technical Report TR 98/11, Dept. of Computer Science, Chonbuk National University, 1998, http://cs.chonbuk.ac.kr/∼jdyang/publication/techpaper.html.

[6] L. Aas and L. Eikvil, "Text categorisation: A survey," Raport NR 941, Norwegian Computing Center, 1999.

[7] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan, "Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies," *The VLDB Journal*, vol. 7, no. 3, pp. 163–178, 1998.

[8] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Ng, "Improving text classification by srinkage in a hierarchy of classes," in *Proc. of the ICML-98*, http://www-2.cs.cmu.edu/∼mccallum/papers/hier-icml98.ps.gz.

[9] W. Chuang, A. Tiyyagura, J. Yang, and G. Giuffrida, "A fast algorithm for hierarchical text classification," in *Proc. of the 2nd Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'00)*, London–Greenwich, UK, 2000, pp. 409–418.

[10] S. T. Dumais, "Improving the retrieval information from external soures," *Behaviour Research Methods, Instruments and Computers*, vol. 23, no. 2, pp. 229–236, 1991.

[11] G. Salton and M. J. McGill, *An Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[12] L. T. Kóczy and T. D. Gedeon, "Fuzzy tolerance relations based on hierarchical co-occurrence of words for intelligent information retrieval," Technical Report, 2000.