Fuzzy Rough Set Based Web Query Expansion

Martine De Cock and Chris Cornelis
Fuzziness and Uncertainty Modelling Research Unit
Dept. of Applied Mathematics and Computer Science
Ghent University, Krijgslaan 281 (S9), 9000 Gent, Belgium
{Martine.DeCock, Chris.Cornelis}@UGent.be

Abstract

Fuzzy rough set theory is a candidate framework for query refinement. Indeed, a thesaurus defines an approximation space in which the query, which is a set of terms, can be approximated from the upper and the lower side. The upper approximation turns out to be too flexible however, resulting in query explosion, while the lower approximation is too strict, resulting in the empty query. Therefore we advocate the use of the lower approximation of the upper approximation, which differs from the upper approximation itself when the thesaurus is not transitive. The resulting technique seems especially useful in the presence of ambiguous query terms.

1 Introduction

One of the most common ways to retrieve information from the WWW is keyword based search: the user inputs a query consisting of one or more keywords and the search system returns a list of web documents ranked according to their relevance to the query. The same procedure is often used in e-commerce applications that attempt to relate the user's query to products from the catalogue of some company.

In the basic approach, documents are not returned as search results if they do not contain (one of) the exact keywords of the query. There are various reasons why such an approach might fall short. On one hand there are word mismatch problems: the user knows what he is looking for and he is able to describe it, but the query terms he uses do not exactly correspond to those in the document containing the desired information because of differences in terminology. This problem is even more significant in the context of the WWW than in other, older information retrieval applications, because of the very heterogeneous sources of information expressed in different jargon or even in different natural languages. Note that, on a more general level, a great deal of the semantic web efforts are concerned with this problem too, which is reflected in all the attention paid to the construction and the representation of ontologies, allowing agents to communicate with each other by providing a shared and common understanding that reaches across people and application systems (see e.g. [4]). In this paper we rely on a basic kind of ontology, called a thesaurus, which is a term-term relation.

Besides differences in terminology, it is also common for a user not to be able to describe accurately what he is looking for; the well known "I will know it when I see it" phenomenon. Furthermore, many terms in natural language are ambiguous. For example, a user querying for *java* might be looking for information about either the programming language, the coffee, or the island of Indonesia. To satisfy users who expects search engines to come up with "what they mean and not what they say", it is clear that more sophisticated techniques are needed than a straightforward returning of the documents that contain (one of) the query terms given by the user. One option is query refinement. Since web queries tend to be short — according to [12] they consist of 1 or 2 terms on average — we focus on query expansion, i.e. the process of adding related terms to the query.

Rough set theory [7] is an interesting candidate framework to aid in query refinement. Indeed, a thesaurus characterizes an approximation space in which the query, which is a set of terms, can be approximated from the upper and the lower side. By definition, the upper approximation will add a term to the query as soon as it is related to *one* of the words already in the query, while the lower approximation will only retain a term in the query if *all* the words that it is related too are also in the query. It is obvious that the lower approximation will easily result in the empty query, hence in practice it is often too strict for query refinement. The upper approximation on the other hand corresponds to a well known straightforward approach to query expansion. However, it is not hard to imagine cases where the upper approximation is too flexible as a query expansion technique, resulting not only in an explosion of the query, but possibly

even worse, in the addition of non relevant terms due to the ambiguous nature of one or more of the query words. This is due to the fact that the upper approximation expands each of the query words individually but disregards the query as a whole.

In this paper we therefore suggest to combine the flexibility of the upper approximation with the strictness of the lower approximation by applying them successively. As such, first we expand the query by adding all the terms that are known to be related to at least one of the query words. Next we reduce the expanded query by taking its lower approximation, thereby pruning away all previously added terms that are suspected to be irrelevant for the query. The pruning strategy targets those terms that are strongly related to words that do not belong to the expanded query.

Our technique can be used both with a crisp thesaurus in which terms are related or not, as with a graded thesaurus in which terms are related to some degree. Furthermore it can be applied for weighted as well as for non weighted queries. Whenever the user does not want to go through the effort of assigning individual weights to query terms, they are all given the heighest weight by default. When a graded thesaurus is used, our query expansion approach turns the original query automatically into a weighted query. The original user chosen terms maintain their heighest weight, and new terms are added with weights that do not only reflect the strength of the relationship with the original individual query terms as can be read from the thesaurus, but also take into account their relevance to the query as a whole. To be able to deal with graded thesauri and weighted queries, we rely on fuzzy rough set theory (see e.g. [3, 9]), representing the thesaurus as a fuzzy relation and the query as a fuzzy set.

The paper is structured as follows: in Section 2 we situate our approach among related work on query refinement. In Section 3 we recall the preliminaries of fuzzy rough set theory and illustrate the inadequacy of the lower and upper approximation as a standalone tool for query refinement. In Section 4 we provide a solution by using an alternative definition of upper approximation, recently introduced in [2], corresponding to the successive application of upper and lower approximation. With a proper choice of the fuzzy logical operators involved, the resulting expanded query is guaranteed to be a superset of the original query. We illustrate the potential of our proposal. A conclusion and outlook of our ongoing work on the topic is presented in Section 5.

2 Related Work on Query Refinement

Query refinement has found its way to popular web search engines, and is even becoming one of those features in which search engines aim to differentiate in their attempts to create their own identity. Simultaneously with search results, Yahoo![1] shows a list of clickable expanded queries in an "Also Try" option under the search box. These queries are derived from logs containing queries performed earlier by others. Google Suggest[2] also uses data about the overall popularity of various searches to help rank the refinements it offers, but unlike the other search engines, the suggestions pop up in the search box while you type, i.e. before you search. Ask Jeeves[3] provides a zoom feature, allowing users to narrow or broaden the field of search results, as well as view results for related concepts.

Query expansion goes back a long way before the existence of the WWW, however. Over the last decades several important techniques have been established. The main idea underlying all of them, is to extend the query with words related to the query terms. One option is to use an available thesaurus such as WordNet, expanding the query by adding synonyms [11]. Related terms can also be automatically discovered from the searchable documents though, taking into account statistical information such as co-occurrences of words in documents or in fragments of documents. The more times terms co-occur, the more they are assumed to be related. In [12] several of these approaches are discussed and compared. In global document analysis, the whole corpus of searchable documents is preprocessed and transformed into an automatically generated thesaurus. Local document analysis on the other hand only considers the top ranked documents for the initial query. In its most naive form, terms that appear most frequently in these top ranked documents are added to the query. Local document analysis is referred to as a pseudo-relevance feedback approach, because it tacitly assumes that the highest ranked documents are indeed relevant to the query. A true relevance feedback approach takes into account the documents marked as relevant by the user. Finally, in [1], correlations between terms are computed based on their co-occurrences in query logs instead of in documents.

Once the relationship between terms is known, either through a lexical aid such as WordNet, or automatically generated from statistical information, the original query can be expanded in various ways. The straightforward way is to extend the query with all the words that are related to at least one of the query terms. As mentioned in the introduction, this corresponds to taking the upper approximation of the query. This link between query expansion and rough set theory has been established in [10], even involving fuzzy logical representations of the term-term relations and the queries. In [11] it is pointed out however that such an approach requires sense resolution of ambiguous words. Indeed, the precision of retrieved documents is likely to de-

[1] http://search.yahoo.com/
[2] http://labs.google.com/suggest/
[3] http://www.ask.com/

crease when expanding a query such as *java, travel* with the term *applet*. Even though this term is highly related to *java* as a programming language, it has little or nothing to do with the intended meaning of *java* in this particular query, namely the island. An option to automate sense disambiguation is to only add a term when it is related to at least two words of the original query; experimental results are however unsatisfactory [11].

In [1] the most popular sense gets preference. For example, if the majority of users use *windows* to search for information about the Microsoft product, the term *windows* has much stronger correlations with terms such as *Microsoft*, *OS* and *software*, rather than with terms such as *decorate*, *door* and *house*. The approaches currently taken by Yahoo! and Google Suggest seem to be in line with this principle. Note though that these search engines do not apply query expansion automatically but leave the final decision up to the user. In [8] a virtual term is created to represent the general concept of the query. Terms are selected for expansion based on their similarity to this virtual term. In [12] candidate expansion terms are ranked based on their co-occurrence with all query terms in the top ranked documents.

Our approach differs from all techniques mentioned above. As will become clear in Section 4, we go further than the expansion of individual query terms, but we do not go as far as restricting ourselves to words that are related to at least two or preferably all terms of the initial query. Instead, we follow an approach where terms can be added as long as they are not strongly related to words that have nothing to do with the query at all. As such we want to contribute to the problem under study of automatic query disambiguation in search engines [5].

3 Fuzzy Rough Set Approach

3.1 Fuzzy Rough Sets

Throughout this paper, let X denote the universe of terms. A fuzzy set A in X is characterized by a $X \to [0,1]$ mapping, called the membership function of A [13]. For all x in X, $A(x)$ denotes the degree to which x belongs to A. Furthermore, a fuzzy relation R in X is a fuzzy set in $X \times X$. For all y in X, the R-foreset of y is the fuzzy set Ry defined by

$$Ry(x) = R(x,y) \tag{1}$$

for all x in X. A fuzzy relation is called reflexive if and only if

$$R(x,x) = 1 \tag{2}$$

for all x in X. Moreover, R is called symmetrical if and only if

$$R(x,y) = R(y,x) \tag{3}$$

for all x and y in X. For A and B fuzzy sets in X, inclusion can be defined as

$$A \subseteq B \text{ iff } (\forall x \in X)(A(x) \leq B(x)) \tag{4}$$

Triangular norms (t-norms for short) and implicators are commonly used as the fuzzy logical generalizations of conjunction and implication. A t-norm $\mathcal{T}$ is any increasing, commutative and associative $[0,1]^2 \to [0,1]$ mapping satisfying

$$\mathcal{T}(1,x) = x \tag{5}$$

for all x in $[0,1]$. An implicator is any $[0,1]^2 \to [0,1]$-mapping $\mathcal{I}$ satisfying the boundary conditions

$$\mathcal{I}(0,0) = 1 \tag{6}$$

$$\mathcal{I}(1,x) = x \tag{7}$$

for all x in $[0,1]$. Moreover we require $\mathcal{I}$ to be decreasing in its first, and increasing in its second component. Throughout this paper, let $\mathcal{T}$ denote a fixed left-continuous t-norm. It can be verified that the mapping $\mathcal{I}_\mathcal{T}$ defined by, for all x and y in [0,1],

$$\mathcal{I}_\mathcal{T}(x,y) = \sup\{\lambda | \lambda \in [0,1] \text{ and } \mathcal{T}(x,\lambda) \leq y\} \tag{8}$$

is an implicator, usually called the residual implicator of $\mathcal{T}$. In Tables 1 and 2, we mention some well known t-norms and their residual implicators. It holds that

$\mathcal{T}_\mathrm{M}(x,y)$	=	$\min(x,y)$
$\mathcal{T}_\mathrm{P}(x,y)$	=	$x \cdot y$
$\mathcal{T}_\mathrm{W}(x,y)$	=	$\max(x+y-1,0)$

Table 1. Well known t-norms (x and y in $[0,1]$)

$\mathcal{I}_{\mathcal{T}_\mathrm{M}}(x,y)$	=	$\begin{cases} 1, & \text{if } x \leq y \\ y, & \text{otherwise} \end{cases}$
$\mathcal{I}_{\mathcal{T}_\mathrm{P}}(x,y)$	=	$\begin{cases} 1, & \text{if } x \leq y \\ \frac{y}{x}, & \text{otherwise} \end{cases}$
$\mathcal{I}_{\mathcal{T}_\mathrm{W}}(x,y)$	=	$\min(1-x+y,1)$

Table 2. Well known residual implicators (x and y in $[0,1]$)

$$\mathcal{I}_{\mathcal{T}}(x,y) = 1 \text{ iff } x \leq y \tag{9}$$

for all x and y in $[0,1]$. Taking this into account helps to understand the following, commonly used generalization of transitivity: a fuzzy relation R in X is called $\mathcal{T}$-transitive if and only if

$$\mathcal{T}(R(x,y), R(y,z)) \leq R(x,z) \tag{10}$$

for all x, y and z in X. Indeed, when R is crisp, i.e. the associated mapping only takes on values in $\{0,1\}$, (10) corresponds to

$$(x,y) \in R \wedge (y,z) \in R \Rightarrow (x,z) \in R \tag{11}$$

The universe X together with a reflexive, symmetrical, and possibly also $\mathcal{T}$-transitive fuzzy relation R in X make up an approximation space (X,R). In this space, every fuzzy set A in X can be approximated from the lower and the upper side. Absorbing earlier suggestions (see e.g. [3]) in the same direction, the following definition was given in [9].

Definition 1 (Lower and Upper Approximation) *The lower and upper approximation of a fuzzy set A in the approximation space (X,R) are the fuzzy sets $R{\downarrow}A$ and $R{\uparrow}A$ defined by*

$$R{\downarrow}A(y) = \inf_{x \in X} \mathcal{I}_{\mathcal{T}}(R(x,y), A(x)) \tag{12}$$

$$R{\uparrow}A(y) = \sup_{x \in X} \mathcal{T}(R(x,y), A(x)) \tag{13}$$

for all y in X.

(A_1, A_2) is called a fuzzy rough set (in (X,R)) as soon as there is a fuzzy set A in X such that $R{\downarrow}A = A_1$ and $R{\uparrow}A = A_2$. When R and A are crisp, i.e. their mappings only take on values in $\{0,1\}$, these definitions of lower and upper approximation coincide with those of Pawlak's original rough set concept [7]. Indeed, in this case (12) reduces to

$$y \in R{\downarrow}A \text{ iff } (\forall x \in X)((x,y) \in R \Rightarrow x \in A) \tag{14}$$

while (13) corresponds to

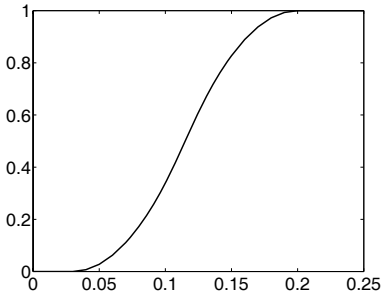
$$y \in R{\uparrow}A \text{ iff } (\exists x \in X)((x,y) \in R \wedge x \in A) \tag{15}$$

When the fuzzy relation R is $\mathcal{T}$-transitive, the lower and upper approximation of A are definable fuzzy sets, i.e.

$$R{\downarrow}(R{\downarrow}A) = R{\uparrow}(R{\downarrow}A) = R{\downarrow}A \tag{16}$$

$$R{\downarrow}(R{\uparrow}A) = R{\uparrow}(R{\uparrow}A) = R{\uparrow}A \tag{17}$$

(17) implies that when the fuzzy thesaurus R is $\mathcal{T}$–transitive, each query A can be expanded only once by taking the upper approximation. However, when R is not $\mathcal{T}$–transitive, a more gradual expansion process is possible, as we illustrate next.



$$S(x;\alpha,\gamma) = \begin{cases} 0, & \text{if } x \leq \alpha \\ \frac{2(x-\alpha)^2}{(\gamma-\alpha)^2}, & \text{if } \alpha \leq x \leq (\alpha+\gamma)/2 \\ 1 - \frac{2(x-\gamma)^2}{(\gamma-\alpha)^2}, & \text{if } (\alpha+\gamma)/2 \leq x \leq \gamma \\ 1, & \text{if } \gamma \leq x \end{cases}$$

Figure 3. S-function; $x, \alpha,$ and γ in $\mathbb{R}$, $\alpha < \gamma$

3.2 Thesaurus Construction

Figure 2 shows a small fuzzy thesaurus R. In constructing it, we did not use any direct human expert knowledge whatsoever regarding the semantics of the terms involved, but we only relied on the number of web pages found by a search engine for each pair of terms, as shown in Figure 1. Let D_{t_1} and D_{t_2} denote the number of web pages that contain term t_1, respectively term t_2; these numbers can be found on the diagonal in Figure 1. On the WWW there is a strong bias towards computer science related terms, hence the absolute number of web pages containing both term t_1 and t_2 cannot be used directly to express the strength of the relationship between t_1 and t_2. To level out the difference, we used the following measure

$$\frac{|D_{t_1} \cap D_{t_2}|}{\min(|D_{t_1}|, |D_{t_2}|)} \tag{18}$$

Finally we normalized the result using the S-function $S(.;0.03,0.20)$ (cfr. Figure 3), giving rise to the fuzzy thesaurus R of Figure 2.

Work on (fuzzy) rough sets often assumes that the relation characterizing the approximation space is transitive. Hence for comparison purposes we also constructed a $\mathcal{T}$-transitive fuzzy thesaurus by taking the $\mathcal{T}$-transitive closure of R, i.e. the smallest $\mathcal{T}$-transitive fuzzy relation in which R is included. It is known that, if the universe X is finite, this closure can be obtained by composing R with itself $|X|-1$ times [6]. Recall that in general the composition of fuzzy relations R and S in X is the fuzzy relation $R \circ S$ in X

# documents	mac	computer	apple	fruit	pie	recipe	store	emulator	hardware
mac	114000	18300	14900	1030	869	899	15800	672	15100
computer		375000	15600	3760	2220	3720	29500	1170	26900
apple			93400	5420	3810	4590	14300	401	17800
fruit				35400	2320	4080	7630	47	1630
pie					20400	4210	3740	30	1200
recipe						31500	6220	35	1690
store							312000	472	24900
emulator								4950	1050
hardware									178000

Figure 1. Number of thousands of web pages found by Google

R	mac	computer	apple	fruit	pie	recipe	store	emulator	hardware
mac	1.00	0.89	0.89	0.00	0.01	0.00	0.75	0.83	0.66
computer		1.00	0.94	0.44	0.44	0.56	0.25	1.00	0.83
apple			1.00	0.83	0.99	0.83	0.83	0.25	0.99
fruit				1.00	0.44	0.66	1.00	0.00	0.03
pie					1.00	1.00	0.97	0.00	0.06
recipe						1.00	1.00	0.00	0.03
store							1.00	0.34	0.75
emulator								1.00	1.00
hardware									1.00

Figure 2. Graded thesaurus

defined by

$$(R \circ S)(x, z) = \sup_{y \in X} \mathcal{T}(R(x, y), S(y, z)) \qquad (19)$$

for all x and z in X. The $\mathcal{T}$-transitive closure of R is the fuzzy relation $R^{|X|-1}$, using the following notation, for $n > 1$,

$$R^1 = R \text{ and } R^n = R \circ R^{n-1} \qquad (20)$$

Figure 4 depicts the $\mathcal{T}_{\mathrm{W}}$-transitive closure of the fuzzy thesaurus shown in Figure 2. In our running example, to compute upper and lower approximations, we will keep on using the t-norm $\mathcal{T}_{\mathrm{W}}$ as well as its residual implicator $\mathcal{I}_{\mathcal{T}_{\mathrm{W}}}$. Finally we constructed a non graded thesaurus taking the 0.5-level set of R, defined as

$$(x, y) \in R_{.5} \text{ iff } R(x, y) \geq 0.5 \qquad (21)$$

for all x and y in X. In other words, in the non graded thesaurus, depicted in Figure 5, two terms are related if and only if the strength of their relationship in the graded thesaurus R of Figure 2 is at least 0.5. It can be easily verified that $R_{.5}$ is not transitive. For example *fruit* is related to *store* and *store* is related to *hardware*, but *fruit* is not related to *hardware*. For comparison purposes, in the remainder, we also include the transitive closure $(R_{.5})^8$.

3.3 Query Refinement

We consider the query

	A	$R{\uparrow}A$	$R{\uparrow}(R{\uparrow}A)$	$R^8{\uparrow}A$
mac	0.00	0.89	0.89	0.89
computer	0.00	0.94	0.94	0.99
apple	1.00	1.00	1.00	1.00
fruit	0.00	0.83	1.00	1.00
pie	1.00	1.00	1.00	1.00
recipe	1.00	1.00	1.00	1.00
store	0.00	1.00	1.00	1.00
emulator	0.00	0.25	0.99	0.99
hardware	0.00	0.99	0.99	0.99

Figure 6. Upper approximation based query expansion with graded thesaurus

apple, pie, recipe

as shown in the second column in Figure 6. The intended meaning of the ambiguous word *apple*, which can refer both to a piece of fruit and to a computer company, is clear in this query. The disadvantage of using a $\mathcal{T}$-transitive fuzzy thesaurus becomes apparent when we compute the upper approximation $R^8{\uparrow}A$, shown in the last column. All the terms are added with high degrees, even though terms like *mac* and *computer* have nothing to do with the semantics of the original query. This process can be slowed down a

R^8	mac	computer	apple	fruit	pie	recipe	store	emulator	hardware
mac	1.00	0.89	0.89	0.88	0.88	0.88	0.88	0.89	0.89
computer		1.00	0.99	0.99	0.99	0.99	0.99	1.00	1.00
apple			1.00	0.99	0.99	0.99	0.99	0.99	0.99
fruit				1.00	1.00	1.00	1.00	0.99	0.99
pie					1.00	1.00	1.00	0.99	0.99
recipe						1.00	1.00	0.99	0.99
store							1.00	0.99	0.99
emulator								1.00	1.00
hardware									1.00

Figure 4. Transitive closure of graded thesaurus

$R_{.5}$	mac	computer	apple	fruit	pie	recipe	store	emulator	hardware
mac	1	1	1	0	0	0	1	1	1
computer		1	1	0	0	1	0	1	1
apple			1	1	1	1	1	0	1
fruit				1	0	1	1	0	0
pie					1	1	1	0	0
recipe						1	1	0	0
store							1	0	1
emulator								1	1
hardware									1

Figure 5. Non graded thesaurus

	A	$R_{.5}{\uparrow}A$	$R_{.5}{\uparrow}(R_{.5}{\uparrow}A)$	$(R_{.5})^8{\uparrow}A$
mac	0	1	1	1
computer	0	1	1	1
apple	1	1	1	1
fruit	0	1	1	1
pie	1	1	1	1
recipe	1	1	1	1
store	0	1	1	1
emulator	0	0	1	1
hardware	0	1	1	1

Figure 7. Upper approximation based query expansion with non graded thesaurus

little bit by using the non $\mathcal{T}$-transitive fuzzy thesaurus and computing $R{\uparrow}A$ which allows for some gradual refinement. However an irrelevant term such as *emulator* shows up to a high degree in the second iteration, i.e. when computing $R{\uparrow}(R{\uparrow}A)$. The problem is even more prominent when using a non graded thesaurus as shown in Figure 7.

4 Tight Upper Approximation

In Pawlak's original rough set theory [7], the approximation space is characterized by an equivalence relation R that partitions the universe in equivalence classes. The fuzzy relational counterpart of equivalence classes are foresets. In this context we also refer to them as soft similarity classes. It is very interesting to note that, unlike in the crisp case where equivalence classes are either equal or disjoint, soft similarity classes can partly overlap, even when the fuzzy relation is reflexive, symmetrical and $\mathcal{T}$-transitive, i.e. a so-called fuzzy $\mathcal{T}$-equivalence relation. In other words, an object y can belong to some degree to several soft similarity classes at the same time.

It is observed in [2] that this property does not only lie at the heart of fuzzy set theory, but it is also crucial in the decision on how to define lower and upper approximations in fuzzy rough set theory. Indeed, in traditional rough set theory y belongs to the upper approximation of A if and only if the equivalence class to which y belongs has a non

empty intersection with A. But what happens if y belongs to several soft similarity classes at the same time? Do we then require that all of them have a non empty intersection with A? Most of them? Or just one? And then, which one? Based on these questions, new definitions of lower and upper approximations in fuzzy rough set theory are proposed in [2]. Very interesting for web query expansion is the following definition of tight upper approximation.

Definition 2 (Tight upper approximation) The tight upper approximation of a fuzzy set A in the approximation space (X, R) is the fuzzy set $R{\downarrow}{\uparrow}A$ defined by

$$R{\downarrow}{\uparrow}A(y) = \inf_{z \in X} \mathcal{I}_{\mathcal{T}}(Rz(y), \sup_{x \in X} \mathcal{T}(Rz(x), A(x))) \quad (22)$$

for all y in X.

One can easily verify that

$$R{\downarrow}{\uparrow}A = R{\downarrow}(R{\uparrow}A) \quad (23)$$

The terminology "tight" refers to the fact that "all" soft similarity classes are taken into account. Informally, a term y is added to the query to the degree to which all terms that are related to y are also related to A. In this way, if a term y is strongly related to any term z that is not clearly related to any of the query terms, y is not added to the query because it might bring on irrelevant search results.

It is important to point out that

$$A \subseteq R{\downarrow}{\uparrow}A \subseteq R{\uparrow}A \quad (24)$$

always holds, guaranteeing that the tight upper approximation indeed leads to an expansion of the query — none of the original terms are lost — and at the same time is a pruned version of the upper approximation. When R is a fuzzy $\mathcal{T}$-equivalence relation, the upper approximation and the tight upper approximation coincide (see (17)). However, as we show below, this is not necessarily the case when R is not $\mathcal{T}$-transitive.

The main problem with the query expansion process described in the previous section, even if it is gradual, is a fast growth of the number of less relevant or irrelevant keywords that are automatically added. This effect is caused by the use of a flexible definition of the upper approximation in which a term is added to a query as soon as it is related to one of its keywords. However, using the tight upper approximation a term y will only be added to a query A if all the terms that are related to y are also related to at least one keyword of the query. First the usual upper approximation of the query is computed, but then it is stripped down by omitting all terms that are also related to other terms not belonging to this upper approximation. In this way terms that are sufficiently relevant, hence related to most keywords in A, will form a more or less closed context with few or no links outside, while a term related to only one of the keywords in A in general also has many links to other terms outside $R{\uparrow}A$ and hence is omitted by taking the lower approximation.

	A	$R{\uparrow}A$	$R^8{\uparrow}A$	$R{\downarrow}{\uparrow}A$
mac	0.00	0.89	0.89	0.42
computer	0.00	0.94	0.99	0.25
apple	1.00	1.00	1.00	1.00
fruit	0.00	0.83	1.00	0.83
pie	1.00	1.00	1.00	1.00
recipe	1.00	1.00	1.00	1.00
store	0.00	1.00	1.00	0.83
emulator	0.00	0.25	0.99	0.25
hardware	0.00	0.99	0.99	0.25

Figure 8. Comparison of upper and tight upper approximation based query expansion with graded thesaurus

	A	$R_{.5}{\uparrow}A$	$(R_{.5})^8{\uparrow}A$	$R_{.5}{\downarrow}{\uparrow}A$
mac	0	1	1	0
computer	0	1	1	0
apple	1	1	1	1
fruit	0	1	1	1
pie	1	1	1	1
recipe	1	1	1	1
store	0	1	1	1
emulator	0	0	1	0
hardware	0	1	1	0

Figure 9. Comparison of upper and tight upper approximation based query expansion with non graded thesaurus

The last column of Figure 8 shows that the tight upper approximation is different from and performs clearly better than the traditional upper approximation for our purpose of web query expansion: irrelevant words such as "mac", "computer" and "hardware" are still added to the query, but to a significantly lower degree. The difference becomes even more noticable when using a non graded thesaurus as illustrated in Figure 9.

5 Concluding Remarks

Since web queries tend to be short — 1 or 2 terms on average — expanding them with related terms is an interesting option for improving rearch results. In the open domain search challenge posed by the web, many terms are ambiguous, i.e. they have more than one possible meaning. An important task for a web query expander therefore is to avoid the addition of irrelevant words, i.e. those words related to meanings of the original terms that were not intended by the user. In this paper we have proposed a new way to address this problem using only a thesaurus, i.e. a term-term relation, besides the original query.

As indicated above, an expansion of individual ambiguous query terms by taking the upper approximation of the query in the space characterized by the thesaurus as proposed in [10] does not give appropriate results. To perform some kind of sense resolution, the web query expander needs to take the query as a whole into account, rather than working on the level of individual query terms. Adding a term to the query if it is related to at least two of the query words does not seem to be a good approximation to sense disambiguation either however [11].

Our proposal consists of two steps. In the first step, the web query expander acts on the level of individual query terms, adding all related terms. When one or more query terms are ambiguous, it can be expected that many of the added terms are irrelevant for the intended meaning of the original query. Hence we apply a second step in which terms are pruned away to the extent to which they are related to words that have nothing to do with the query as a whole.

In this paper, we have demonstrated how the so-called tight upper approximation, i.e. a successive application of the upper and the lower approximation from fuzzy rough set theory, can be used to this purpose. The given example clearly shows potential. It is still an open question to be explored whether fuzzy rough set based query expansion is robust enough to improve search on the web using a thesaurus with ten thousands of links between terms.

Acknowledgement

Martine De Cock would like to thank the Fund for Scientific Research—Flanders for supporting the research reported on in this paper.

References

[1] H. Cui, J.-R. Wen, J.-Y. Nie and W.-Y. Ma (2002) Probabilistic query expansion using query logs. In: Proceedings of WWW2002 (the 11th International World Wide Web Conference), ACM Press, p. 325–332

[2] M. De Cock, C. Cornelis, and E. E. Kerre (2004) Fuzzy Rough Sets: Beyond the Obvious. In: Proceedings of FUZZIEEE2004 (2004 IEEE International Conference on Fuzzy Systems), vol. 1, p. 103–108

[3] D. Dubois and H. Prade (1990) Rough fuzzy sets and fuzzy rough sets, International Journal of General Systems, 17, p. 191–209

[4] D. Fensel, F. van Harmelen, I. Horrocks, D. L. Guinness and P. F. Patel-Schneider (2001) OIL: An Ontology Infrastructure for the Semantic Web, IEEE Intelligent Systems, 16, p. 38–45

[5] A. Fischer (2003) What's it Going to Take to Beat Google? Search Engine Watch, June 13, 2003

[6] H. Naessens, H. De Meyer, B. De Baets (2002) Algorithms for the Computation of T-Transitive Closures, IEEE Transactions on Fuzzy Systems 10(4), p. 541-551

[7] Z. Pawlak (1982) Rough sets, International Journal of Computer and Information Sciences, 11(5), p. 341–356

[8] Y. Qui, H. Frei (1993) Concept based query expansion, In: Proceedings of ACM SIGIR 1993 (16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval), p. 160–169

[9] A. M. Radzikowska, E. E. Kerre (2002) A comparative study of fuzzy rough sets, Fuzzy Sets and Systems, 126, p. 137–156

[10] P. Srinivasan, M. E. Ruiz, D. H. Kraft, J. Chen (2001) Vocabulary Mining for Information Retrieval: Rough Sets and Fuzzy Sets, Information Processing and Management 37, p. 15–38

[11] E. M. Voorhees (1994) Query expansion using lexical-semantic relations. In: Proceedings of ACM SIGIR 1994 (17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval), p. 61–69

[12] J. Xu and W. B. Croft (1996) Query Expansion Using Local and Global Document Analysis, In: Proceedings of ACM SIGIR 1996 (19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval), p. 4–11

[13] L. A. Zadeh (1965) Fuzzy Sets, Information and Control 8, p. 338-358