Semantic Relations and Information Discovery

D. Cai and C. J. van Rijsbergen

Department of Computing Science, University of Glasgow, G12 8RZ, Scotland
{caid;keith}@dcs.gla.ac.uk

Abstract

The treatment of semantic relations between terms is essential in information retrieval (IR). Each term in a thesaurus might have classes of synonymous, narrower, broader, or related terms. The ability to express formally the semantic relations is a core issue in applying mathematical tools to IR. In the application of evidential theory, particularly, the problems become more apparent since evidential theory is based on set theory and individual terms have to be expressed as subsets of the frame of discernment. Four basic questions are to be faced: How to establish the frame of discernment using a thesaurus? How to express terms using subsets of the frame? How to apply evidential functions to represent documents or queries using the term subset-expressions? What are appropriate agreement measures for ranking documents against a given query? This study attempts to answer these questions.

1 Introduction

The objective of an IR system is to identify latent useful information in response to user information needs. The effectiveness of an IR model depends mainly on three central issues: document representation, query representation and agreement (similarity) measurement. Many important studies focus on the three issues, and some good formal methods have been developed.

In IR, each document is characterized by a set of *index terms* that appear in the document. There exist complex semantic relations between index terms. Generally, a *weighting* function, which maps each index term to a numerical quantity related to a given document, is used to represent the importance of the index term concerning the document. It should be pointed out that to arrive at a precise representation of a document by means of a set of weights of index terms is difficult. This is because it is very hard to obtain sufficient statistical data for estimating the importance of index terms. It is also very hard to explicate the complicated semantic relations between index terms. Extensive studies on document representation can be found in, for instance [1, 11, 17, 20, 29, 30, 28, 32, 33, 34].

Query representation is also an obstacle to developing an effective retrieval system. In practice, the original queries are usually imprecise and incomplete descriptions of information needs. A retrieval system cannot thus be expected to produce ideal retrieval results by using a poor query representation. Some thorough investigations into query representation and query expansion can be found in, for instance [4, 6, 7, 8].

An agreement measure determines the degree to which individual documents are relevant to the query. To be successful, the determination should be performed in such a way that retrieval output and actual outcome are, on average, in close agreement. The choice of agreement measure is essential for effective retrieval. The relevance problem has been studied by many researchers, for instance [2, 9, 15, 23].

The treatment of semantic relations between terms has long been a significant subject of interest in IR. Terms in a thesaurus might have a class of synonymous terms, a class of narrower terms, a class of broader terms, or a class of related terms. For example, semantically, terms *cows* and *goats* are narrower than term *mammals*. In a semantic net, "term t_1 is narrower than term t_2" is expressed by an arrow from t_1 to t_2 to impose a specification/generalization relation on terms. An example for related terms is that terms *ducks*, *geese*, *hens* might be related to terms *eggs*, *feathers*. It is essential to normalize a thesaurus for treating term semantic relations in IR applications. This study presents a method for the normalization.

The ability to express formally the semantic relations of terms is a core issue in IR. Terms are not mutually exclusive, and naive probabilistic methods may not be adequate for handling the issue. The evidential method appears to be more convenient than the usual probabilistic methods [32] to express term semantic relations, to represent *objects* (i.e., documents and queries), and to rank documents against a given query. Some retrieval methods based on evidential theory (Dempster-Shafer's theory of evidence [25]) have been proposed [17, 18, 24, 26]. However, in applications of evidential theory, the construction of a σ-algebra and a base for this σ-algebra in order to construct a probability space is an arduous task [26]. This study proposes a method to establish the frame of discernment, and to express each term by a subset of the frame.

In this study, we are also concerned with the application of evidential theory to some practical IR problems: we formally discuss the representations of objects by means of evidential functions; we introduce agreement measures over the evidential representations for ranking documents against the query.

This paper is organized as follows. In Section 2, we suggest a method for normalizing a thesaurus semantically. In Section 3, we propose a method for establishing a frame of discernment and for expressing terms as subsets of the frame. In Section 4, after introducing basic concepts of evidential theory, a novel method for representing objects based on evidential functions is proposed, and the agreement measures for ranking documents based on the evidential representations are introduced.

2 Normalization of a Thesaurus

This section concentrates on how to normalize a thesaurus semantically. We introduce notation for the normalization. The notation is used to describe the relations found in a general thesaurus. An example thesaurus $\aleph$, taken from [26], is used throughout this paper. It is given in Table 1. We will denote T as the set of 28 terms contained in $\aleph$.

2.1 Treatment of Synonymous Terms

We first introduce two pieces of notation, $\Leftarrow$ and $\Rightarrow$.

☞ "$\Leftarrow$" is used to denote 'use-for'.

"$t_1 \Leftarrow t_2, t_3, ..., t_l$" means that t_1 and $t_2, t_3, ..., t_l$ are synonymous terms, and that t_1 is their representative.

☞ "$\Rightarrow$" is used to denote 'use'.

"$t_1 \Rightarrow t_2$" means that t_1 and t_2 are synonymous terms, and that t_2 is their representative.

The *synonymous* relation between terms is an equivalence relation. That is, it has the following properties:

reflexive: $t \Leftarrow t$;

symmetric: if $t_1 \Leftarrow t_2$, then $t_2 \Leftarrow t_1$;

transitive: if $t_1 \Leftarrow t_2$, and $t_2 \Leftarrow t_3$, then $t_1 \Leftarrow t_3$.

Removing the tuples with relations $\supset$, $\subset$ and $\cap$ from the thesaurus, and then with symmetry, further removing relation $\Rightarrow$, we can obtain a *synonym-normalized* thesaurus, denoted $\mathcal{S}_{(k,t)}$. For the example thesaurus $\aleph$, we obtain Table 2.

In contrast to thesaurus $\aleph$, we can see that the terms that do not have relations $\Leftarrow$ or $\Rightarrow$, but have relations $\supset$, $\subset$ and $\cap$ remain in the first column of thesaurus $\mathcal{S}_{(k,t)}$ (e.g., terms 'domestic-birds', 'eggs', etc.). In other words, only those terms that have only relation $\Rightarrow$ are removed from the first column of $\aleph$ (e.g., terms 'barnyard-birds', 'farm-animals', etc.).

In what follows, we will call the terms listed in the first column of thesaurus $\mathcal{S}_{(k,t)}$ *key-terms*, and denote $K = \{k_1, k_2, ..., k_n\}$ as the set of the key-terms. Obviously, $K \subseteq T$ and $n = |K| \leq |T|$. For instance, for thesaurus $\aleph$, we have $n = 15$ key-terms.

Table 1. An example thesaurus $\aleph$

Term	Relation	Term(s)
animals	$\Leftarrow$	animal
	$\supset$	birds, domestic-animals, mammals
barnyard-birds	$\Rightarrow$	poultry
birds	$\Leftarrow$	bird
	$\subset$	animals
	$\supset$	domestic-birds, poultry
	$\cap$	eggs, feathers
cows	$\Leftarrow$	cow
	$\subset$	domestic-mammals
domestic-animals	$\Leftarrow$	farm-animals
	$\subset$	animals
	$\supset$	domestic-birds, domestic-mammals, poultry
domestic-birds	$\subset$	birds, domestic-animals
	$\supset$	poultry
domestic-mammals	$\subset$	domestic-animals, mammals
	$\supset$	cows, goats
ducks	$\Leftarrow$	duck
	$\subset$	poultry
eggs	$\cap$	birds, poultry
farm-animals	$\Rightarrow$	domestic-animals
farmyard-birds	$\Rightarrow$	poultry
feathers	$\cap$	birds, poultry
geese	$\Leftarrow$	goose
	$\subset$	poultry
goats	$\Leftarrow$	goat
	$\subset$	domestic-mammals
hens	$\Leftarrow$	chick, chicken-cock, hen
	$\subset$	poultry
mammals	$\Leftarrow$	mammal
	$\subset$	animals
	$\supset$	domestic-mammals
	$\cap$	milk
milk	$\cap$	mammals
poultry	$\Leftarrow$	barnyard-birds, farmyard-birds
	$\subset$	birds, domestic-animals, domestic-birds
	$\supset$	ducks, geese, hens
	$\cap$	eggs, feathers

Table 2. A synonym-normalized thesaurus $\mathcal{S}_{(k,t)}$

Key-Term	Relation	Term(s)
animals	$\Leftarrow$	animal
birds	$\Leftarrow$	bird
cows	$\Leftarrow$	cow
domestic-animals	$\Leftarrow$	farm-animals
domestic-birds	-	
domestic-mammals	-	
ducks	$\Leftarrow$	duck
eggs	-	
feathers	-	
geese	$\Leftarrow$	goose
goats	$\Leftarrow$	goat
hens	$\Leftarrow$	chick, chicken-cock, hen
mammals	$\Leftarrow$	mammal
milk	-	
poultry	$\Leftarrow$	barnyard-birds, farmyard-birds

2.2 Treatment of Ordering Terms

Let us introduce two further pieces of notation $\supset$ and $\subset$.

☞ "$\supset$" is used to denote 'narrower-terms'.

"$t_1 \supset t_2, t_3, ..., t_l$" means that term t_1 has narrower terms $t_2, t_3, ..., t_l$. The *narrower* relation between terms is a partial ordering relation. That is, it has the following properties:

irreflexive: $t \not\supset t$;

asymmetric: if $t_1 \supset t_2$ then $t_2 \not\supset t_1$;

transitive: if $t_1 \supset t_2$ and $t_2 \supset t_3$, then $t_1 \supset t_3$.

☞ "$\subset$" is used to denote 'broader-terms'.

"$t_1 \subset t_2, t_3, ..., t_l$" means that term t_1 has broader terms $t_2, t_3, ..., t_l$. The *broader* relation between terms is a partial ordering relation. That is, it has the following properties:

irreflexive: $t \not\subset t$;

asymmetric: if $t_1 \subset t_2$ then $t_2 \not\subset t_1$;

transitive: if $t_1 \subset t_2$ and $t_2 \subset t_3$, then $t_1 \subset t_3$.

Clearly, term t_i has a narrower term t_j if and only if term t_j has a broader term t_i. This implies that we can use only one of these two ordering relations in a thesaurus to obtain an *ordering-normalized* thesaurus, denoted $\mathcal{O}_{(k_i,k_j)}$.

For the example thesaurus $\aleph$, removing the tuples with relations $\Leftarrow$, $\Rightarrow$ and $\cap$, and then further removing relation $\subset$, we obtain Table 3.

Table 3. An ordering-normalized thesaurus $\mathcal{O}_{(k_i,k_j)}$

Key-Term	Relation	Key-Term(s)
animals	$\supset$	birds, domestic-animals, mammals
birds	$\supset$	domestic-birds, poultry
domestic-animals	$\supset$	domestic-birds, domestic-mammals, poultry
domestic-birds	$\supset$	poultry
domestic-mammals	$\supset$	cows, goats
mammals	$\supset$	domestic-mammals
poultry	$\supset$	ducks, geese, hens

The narrower and broader relations of terms, forming a hierarchical structure, are given in Fig. 1. Generally, in IR, a *hierarchical structure*, represented using a *directed acyclic graph* (using arrows), is a tree-like structure, which embeds relations $\supset$ or $\subset$ in a form such that no key-term appears on a level below that of its narrower key-term.

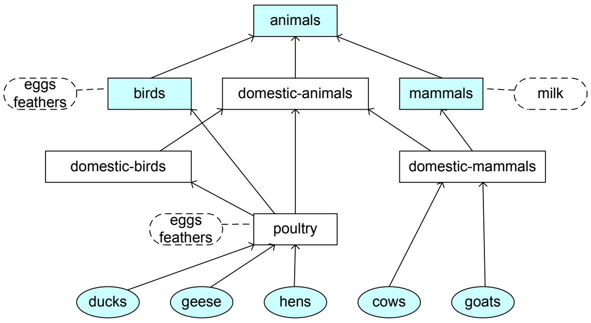

Fig. 1. The narrower and broader relations of key-terms on a hierarchical structure.

Notice that all terms given in the first and third columns of thesaurus $\mathcal{O}_{(k_i,k_j)}$ (i.e., on the hierarchical structure) are key-terms. In what follows, we will denote the set of all key-terms on the hierarchical structure as $hie(K)$. Obviously, $hie(K) \subseteq K$, and it may be that $K - hie(K) \neq \emptyset$. For instance, from our example, we can see that $K - hie(K) = \{eggs, feathers, milk\}$.

We denote the hierarchical structure itself as $H(K)$. A key-term k is said to be on structure $H(K)$, denoted by $k \vdash H(K)$, if there exists one node of $H(K)$, such that it is used to represent k. Clearly, $k \in hie(K)$ if $k \vdash H(K)$.

2.3 Treatment of Related Terms

We need to introduce another piece of notation, $\cap$.

☞ "$\cap$" is used to denote 'related-terms'.

"$t_1 \cap t_2, t_3, ..., t_l$" means that term t_1 has related terms $t_2, t_3, ..., t_l$. The *related* relation between terms has the following properties:

reflexive: $t \cap t$;

symmetric: if $t_1 \cap t_2$, then $t_2 \cap t_1$;

transitive: if $t_1 \cap t_2$, and $t_2 \cap t_3$, then $t_1 \cap t_3$ may not hold.

For the example thesaurus $\aleph$, deleting the tuples with relations $\Leftarrow$, $\Rightarrow$, $\supset$ and $\subset$, and then with symmetry, we can obtain Table 4.

Table 4. A related-normalized thesaurus $\mathcal{R}_{(k_i,k_j)}$

Key-Term	Relation	Key-Term(s)
birds	$\cap$	eggs, feathers
mammals	$\cap$	milk
poultry	$\cap$	eggs, feathers

The related relation of terms linked to the hierarchical structure (using dashed lines) is given in Fig. 1.

Notice that all terms listed in the first and third columns of thesaurus $\mathcal{R}_{(k_i,k_j)}$ are key-terms. For two related key-terms $k_i, k_j \in K$, we always have $k_i \cap k_j$ and $k_j \cap k_i$. However, we use the following rules to write their related relation:

- if $k_i \in hie(K)$ and $k_j \in K - hie(K)$, write $k_i \cap k_j$;
- if $k_i \in K - hie(K)$ and $k_j \in hie(K)$, write $k_j \cap k_i$;
- if $k_i, k_j \in hie(K)$, simply ignore them as their related relation has been implied by their narrower or broader relations (see Sect. 2.4);
- if $k_i, k_j \notin hie(K)$, simply ignore them as their related relation is not linked to the hierarchical structure.

That is, the right side of $\cap$ is always a key-term in $hie(K)$, i.e., in the first column of the related-normalized thesaurus; the left side of $\cap$ is always a key-term in $K - hie(K)$, i.e., in the third column of the related-normalized thesaurus.

In what follows, we will denote the set of the key-terms linked to the hierarchical structure (i.e., listed in the third column of thesaurus $\mathcal{R}_{(k_i,k_j)}$) as $rla(K)$. Obviously, $rla(K) \subseteq K$ and $hie(K) \cap rla(K) = \emptyset$. For our examples, we can see that $rla(K) = \{eggs, feathers, milk\}$.

Notice also that, for an arbitrary key-term k_j in $K - hie(K)$, we have three and only three cases:

➟ There exists at least one $k_i \vdash H(K)$, such that, $k_i \cap k_j$; in this case, $k_j \in rla(K)$.

- There exists one $k_{i_1} \vdash H(K)$ and $k_{i_2}, ..., k_{i_\lambda} \not\vdash H(K)$ $(\lambda \geq 2)$, such that, $k_{i_1} \cap k_{i_2}, k_{i_2} \cap k_{i_3}, ..., k_{\lambda-1} \cap k_\lambda, k_\lambda \cap k_j$; in this case, $k_j \notin rla(K)$, as the related relation does not satisfy transitivity.
- There exists no $k_i \vdash H(K)$, such that, $k_i \cap k_j$; in this case, $k_j \notin rla(K)$.

2.4 Superiority and Inferiority

Having discussed the properties of the ordering and related relations we can further discuss the properties between these two relations. The properties which are useful for an insight into the term semantic relations are:

superior: if $t_1 \cap t_2$ and $t_3 \supset t_2$, then $t_1 \cap t_3$;

inferior: if $t_1 \cap t_2$ and $t_2 \supset t_3$, then $t_1 \cap t_3$ may not hold.

From the superiority property we obtain the other two properties:

- If $t_1 \supset t_2$, then $t_1 \cap t_2$.

  In fact, from reflexivity we have $t_2 \cap t_2$. Now $t_2 \subset t_1$. So $t_2 \cap t_1$ by superiority, i.e., $t_1 \cap t_2$ by symmetry.
- If $t_1 \subset t_2$, then $t_1 \cap t_2$.

  In fact, from reflexivity we have $t_1 \cap t_1$. Now $t_1 \subset t_2$. So $t_1 \cap t_2$.

By the superiority property, we can further infer the related relations between key-terms. For instance, from $milk \cap mammals$ and $animals \supset mammals$, we have $milk \cap animals$.

However, the inferiority property may not always hold. For instance, from $milk \cap animals$ and $animals \supset birds$, an inferred result $milk \cap birds$ obviously makes no sense.

In what follows, we will call $\mathcal{S}_{(k,t)}$, $\mathcal{O}_{(k_i,k_j)}$, $\mathcal{R}_{(k_i,k_j)}$ together the *semantically normalized thesaurus*, and denote it by

$$\aleph_{\mathcal{SOR}} = \left[\aleph;\ \mathcal{S}_{(k,t)}|\mathcal{O}_{(k_i,k_j)}|\mathcal{R}_{(k_i,k_j)}\right].$$

Our aim is to formally express the semantic relations between terms, that is, to establish the frame of discernment, and then to express all key-terms as subsets of the frame. The next section attempts to discuss this core issue.

Before discussing the core issue, we first point out that the normalized thesaurus is a precondition for our method to be used. However, in the real world, it is very likely that a thesaurus will not satisfy the conditions of normalization. The problem of normalizing a thesaurus is a pressing one, and may necessitate much effort. It is beyond the scope of this paper to discuss such a problem, and will be treated as a significant subject for further study. Thus, in what follows, we always assume that thesaurus $\aleph$ has been normalized to thesaurus $\aleph_{\mathcal{SOR}}$.

3 Subset-Expressions of Key-Terms

As is known, in the application of evidential theory to IR, the frame of discernment, in which elements are exclusive and exhaustive, must first be established, and each key-term must be expressed as the subset of the frame. This section attempts to give a method to establish the frame and derive the subset-expressions of key-terms.

To do so, we need to introduce two further pieces of notation, $\rightleftharpoons$ and $\leftleftarrows$.

☞ "$\rightleftharpoons$" is used to denote 'expressed-by'.

"$k_1 \rightleftharpoons \{k_2, k_3, ..., k_l\}$", where $k_2, k_3,$ ..., k_l are atomic-terms, means that key-term k_1 is expressed by the set of these atomic-terms.

☞ "$\leftleftarrows$" is used to denote 'equivalent-to'.

"$k_1 \leftleftarrows k_2, k_3, ..., k_l$", where $k_1, k_2, ..., k_l$ have the same subset-expression, means that $k_1, k_2, k_3, ..., k_l$ are equivalent key-terms, and k_1 is their representative.

3.1 The Sub-Frame of Discernment Θ'

The derivation of all atomic-terms is the starting point for establishing the frame of discernment.

Atomic-terms can be derived from the hierarchical structure $H(K)$. In order to generate $H(K)$, we arrange key-terms from narrower to broader (or from specific to general) by using an *arrow* pointing to a 'parent' node (representing a key-term) k_i from a 'child' node k_j; the arrow denotes the relation $k_i \supset k_j$. The oldest node of $H(K)$, such as key-term *animals*, is called the root of the hierarchical structure (see Sec. 3.8).

The nodes to which no arrows point, are called *terminal nodes.* All terminal nodes are regarded as *atomic-terms.* The set of all atomic-terms, denoted by Θ', is called the *sub-frame of discernment.* Obviously, $\Theta' \subseteq hie(K)$. From Fig. 1, we can see that only five key-terms are atomic-terms in thesaurus $\aleph_{\mathcal{SOR}}$:

$$\Theta' = \{cows, ducks, geese, goats, hens\}.$$

Once all atomic-terms on $H(K)$ are derived, we are able to further express general key-terms using a subset of the sub-frame, called a *subset-expression.*

Each atomic-term can be expressed by itself. That is, for an arbitrary $a' \in \Theta'$, $a' \rightleftharpoons \{a'\}$. Thus, atomic-terms are always pairwise unrelated since the intersection of their subset-expressions is always empty.

3.2 Key-Terms in Set $hie(K)$

The subset-expressions of general key-terms can also be derived from the hierarchical structure $H(K)$.

First, consider all key-terms that are narrower or broader than at least one other key-term. For an arbitrary $k \in hie(K) \subseteq K$, a key point of the derivation is to find all possible narrower key-terms, and then traverse downward to atomic-terms. The subset of atomic-terms narrower than k is used as the subset-expression of k. For instance, in Fig. 1, we see that key-term *birds* has narrower atomic-terms *ducks*, *geese* and *hens*, we can thus express *birds* by a subset $\{ducks, geese, hens\}$. Further, with the symbol $\rightleftharpoons$, we can write subset-expressions for all key-terms in $hie(K)$ as shown in Table 5.

Table 5. Subset-expressions for key-terms in set $hie(K)$

Key-Term	Relation	Subset-Expression
animals	$\rightleftharpoons$	$\{cows, ducks, geese, goats, hens\}$
birds	$\rightleftharpoons$	$\{ducks, geese, hens\}$
cows	$\rightleftharpoons$	$\{cows\}$
domestic-animals	$\rightleftharpoons$	$\{cows, ducks, geese, goats, hens\}$
domestic-birds	$\rightleftharpoons$	$\{ducks, geese, hens\}$
domestic-mammals	$\rightleftharpoons$	$\{cows, goats\}$
ducks	$\rightleftharpoons$	$\{ducks\}$
geese	$\rightleftharpoons$	$\{geese\}$
goats	$\rightleftharpoons$	$\{goats\}$
hens	$\rightleftharpoons$	$\{hens\}$
mammals	$\rightleftharpoons$	$\{cows, goats\}$
poultry	$\rightleftharpoons$	$\{ducks, geese, hens\}$

3.3 Key-Terms in Set $rla(K)$

Next, consider all key-terms that are not narrower or broader than any other key-terms, but are directly related to at least one other key-term on $H(K)$. For an arbitrary $k \in rla(K) \subseteq K$, link k to the hierarchical structure using a *dashed line* between k and the key-terms to which k is related; check the *youngest* one among the key-terms; express k using the same subset-expression as the youngest one. For instance, in Fig. 1, we see key-term *eggs* linked to the hierarchical structure by dashed lines between it and two key-terms *birds* and *poultry*. The youngest of the two key-terms is *poultry* with subset-expression $\{ducks, geese, hens\}$, we can thus express *eggs* by the same subset *poultry* has. Subset-expressions for all key-terms in $rla(K)$ are shown in Table 6.

Notice that the choice of the youngest key-term among the key-terms related to a given key-term is essential: it ensures that the related relation is able to satisfy the superiority property (but does not ensure it satisfies the inferiority property).

Table 6. Subset-expressions for key-terms in set $rla(K)$

Key-Term	Relation	Subset-Expression
eggs	$\rightleftharpoons$	$\{ducks, geese, hens\}$
feathers	$\rightleftharpoons$	$\{ducks, geese, hens\}$
milk	$\rightleftharpoons$	$\{cows, goats\}$

3.4 Representatives in Set $rep(K)$

Often, while all key-terms in $hie(K) \cup rla(K)$ are expressed by subset-expressions, some key-terms have the same subset-expressions.

Two key-terms $k_i, k_j \in hie(K) \cup rla(K)$ are said to be *equivalent*, denoted by $k_i \leftleftarrows k_j$, if they have the same subset-expression. The set of equivalent key-terms is called an *equivalent class*, denoted by $equ(k)$, where k is a representative of the equivalent class. Denote $rep(K)$ as the set of all the representatives and, obviously, $rep(K) \subseteq \big(hie(K) \cup rla(K)\big)$.

We can choose a *representative* for each equivalent class by taking the *oldest* key-term of the class. For instance, from Fig. 1 and from Tables 5 and 6, we see that key-terms *birds*, *domestic-birds*, *eggs*, $feathers$ and *poultry* have the same subset-expression, and that *birds* is the oldest one, we can thus take *birds* as the representative of the equivalent class *birds*, *domestic-birds*, *eggs*, $feathers$ and *poultry*. Further, with the symbol $\leftleftarrows$, we can write all representatives for key-terms in $hie(K) \cup rla(K)$ as shown in Table 7.

Table 7. Equivalent classes' representatives in set $rep(K)$

Representative	Relation	Key-Term(s)
animals	$\leftleftarrows$	domestic-animals
birds	$\leftleftarrows$	domestic-birds, eggs, feathers, poultry
mammals	$\leftleftarrows$	domestic-mammals, milk

Notice that the choice of the oldest key-term as the representative of an equivalent class is immaterial: this is done only for the purpose that the root of the hierarchical structure can be chosen as a representative.

It is worth mentioning, similar to the synonym relation $\Leftarrow$ on the term set T, that relation $\leftleftarrows$ is an equivalent relation on the key-term set K with the following properties:

reflexive: $k \leftleftarrows k$ for an arbitrary $k \in K$;

symmetric: if $k_i \leftleftarrows k_j$ then $k_j \leftleftarrows k_i$, where $k_i, k_j \in K$;

transitive: if $k_i \leftleftarrows k_j$ and $k_j \leftleftarrows k_l$ then $k_i \leftleftarrows k_l$, where $k_i, k_j, k_l \in K$.

It may seem odd that we have relation $birds \mathrel{\overline{\leftarrow}} eggs$ and $mammals \mathrel{\overline{\leftarrow}} milk$, etc. Nevertheless, it is mathematically reasonable. Due to the limitations of thesaurus $\aleph_{\mathcal{SOR}}$, key-terms may not be entirely distinguished from each other. Semantically, it might be doubtful that key-terms *birds* and *eggs* are 'the same' according to some knowledge. Mathematically, key-terms *birds* and *eggs* are both equal to $\{ducks, geese, hens\}$ according to the sub-frame of discernment Θ', which has only five atomic-terms totally. Semantically, two key-terms are 'the same' if they are synonymous. Mathematically, two key-terms are equivalent if they contain the same atomic-terms.

3.5 Isolated-Terms in Set $iso(K)$

Also, we need consider some isolated key-terms. A key-term k' is said to be an *isolated-term* if there exists no key-term $k_i \in hie(K)$, such that, $k_i \supset k'$ or $k_i \cap k'$. That is, if key-term k' is isolated, then k' is neither on the hierarchical structure, nor (directly) related to any key-term which is on the hierarchical structure. Denote

$$iso(K) = K - hie(K) - rla(K).$$

Then, $k' \in iso(K)$ is an isolated-term. For instance, from Tables 2, 5 and 6, we can see that $iso(K) = \emptyset$.

Like atomic-terms, each isolated-term can be expressed by itself. That is, for an arbitrary $k' \in iso(K)$, $k' \rightleftharpoons \{k'\}$. Thus, isolated-terms are also pairwise unrelated as the intersection of their subset-expressions is always empty.

However, in practice, isolated-terms may themselves be semantically related to one another. Further study on how to treat the related relation of isolated-terms is needed.

3.6 The Frame of Discernment Θ

Generally, a *frame of discernment*, denoted by Θ, can immediately be established after the sub-frame of discernment and the set of isolated-terms are given:

$$\Theta = \Theta' \cup iso(K) = \{a_1, a_2, ..., a_{|\Theta|}\}.$$

Clearly, $|\Theta| = |\Theta'| + |iso(K)|$ as $\Theta' \cap iso(K) = \emptyset$.

3.7 The Evidence Sub-Space K'

Finally, we form an evidence sub-space K'. An *evidence sub-space*, denoted by K', is a subset of the power set of the frame of discernment Θ, over which the evidence functions can be defined:

$$K' = \Theta \cup rep(K) = \{k'_1, k'_2, ..., k'_m\},$$

where the dimensionality of the sub-space satisfies: $m = |K'| \leq |\Theta| + |rep(K)|$ as it may be that $\Theta \cap rep(K) \neq \emptyset$ [5].

The key-terms in K' are called *kernel-terms*. From the above discussion, it is clear that each kernel-term in $K' \subseteq K$ can be expressed by a subset of Θ. For thesaurus $\aleph_{\mathcal{SOR}}$, we can write subset-expressions for all kernel-terms as shown in Table 8.

Table 8. Subset-expressions for kernel-terms in set K'

Kernel-Term	Relation	Subset-Expression
animals	$\rightleftharpoons$	$\{cows, ducks, geese, goats, hens\}$
birds	$\rightleftharpoons$	$\{ducks, geese, hens\}$
cows	$\rightleftharpoons$	$\{cows\}$
ducks	$\rightleftharpoons$	$\{ducks\}$
geese	$\rightleftharpoons$	$\{geese\}$
goats	$\rightleftharpoons$	$\{goats\}$
hens	$\rightleftharpoons$	$\{hens\}$
mammals	$\rightleftharpoons$	$\{cows, goats\}$

It is important to understand that representatives (kernel-terms) in $rep(K)$ may be related to each other as the intersection of their subset-expressions over Θ' may not be empty. For instance, representatives *birds* and *mammals* are not related to one another, but both of them are related to representatives *animals*. Consequently, kernel-terms in $K' \supseteq rep(K)$ may be related to each other, and may not partition the frame of discernment $\Theta \supseteq \Theta'$.

3.8 Multiple Subset-Expressions

In the above discussion, we gained insight into the concept of term semantic relations by means of a normalized thesaurus, and a hierarchical structure generated from the normalized thesaurus. From thesaurus $\aleph_{\mathcal{SOR}}$, for instance, we generated a hierarchical structure $H(K)$ with a root *animals*. In a real world application, however, there may exist many roots with respect to a given normalized thesaurus.

A key-term k is called a *root*, if

- there exists no key-term k_i, such that, $k_i \supset k$;
- there exists at least one key-term k_j, such that, $k \supset k_j$.

Suppose there are s key-terms that are roots, and denote the set of roots as $R(K) = \{r_1, r_2, ..., r_s\}$, where $0 \leq s < n$. Each root will generate a hierarchical structure or, more precisely, a *hierarchical sub-structure*.

Denote $H_{r_i}(K)$ as the hierarchical sub-structure corresponding to root r_i, $hie_{r_i}(K)$ as the set of key-terms on $H_{r_i}(K)$ (including root r_i), Θ_{r_i} as the set of atomic-terms corresponding to root r_i. Clearly, $hie_{r_i}(K) \supset \Theta_{r_i}$ and so $|hie_{r_i}(K)| > |\Theta_{r_i}| \geq 1$.

For two arbitrary roots $r_i, r_j \in R(K)$, it is likely that $hie_{r_i}(K) \cap hie_{r_j}(K) \neq \emptyset$. That is, a key-term may be on, or linked to, both sub-structures $H_{r_i}(K)$ and $H_{r_j}(K)$. In particular, it may be that $\Theta_{r_i} \cap \Theta_{r_j} \neq \emptyset$, that is, an atomic-term may be on both $H_{r_i}(K)$ and $H_{r_j}(K)$.

In the context of IR, each root r_i should be regarded as referring to one specific topic, and other key-terms on $H_{r_i}(K)$ are its narrower key-terms under the same topic. Thus, all roots should be regarded as pairwise unrelated.

While key-term k is on, or linked to, more than one hierarchical sub-structure, it would have multiple subset-expressions corresponding to the individual roots. This usually happens when key-term k is polysemous (multiple-meaning). For instance, key-term 'phoenix' may have several meanings:

'the capital and largest city of Arizona', or

'a bird in Egyptian mythology', or

'a constellation in the Southern Hemisphere near Tucana and Sculptor'.

Thus, 'phoenix' may be on, or linked to, three sub-structures.

Since each root is considered to refer to only one topic, the multiple subset-expressions will be treated as different from root to root. In particular, when $a \in \Theta_{r_i} \cap \Theta_{r_j}$, we denote $a \rightleftharpoons \{a\}_{r_i}$ and $a \rightleftharpoons \{a\}_{r_j}$, and regard a as having different (semantic) meanings, corresponding to roots r_i and r_j, respectively.

In an extreme case, there is no root: $s = 0$. Thus, the sub-frame of discernment $\Theta' = \emptyset$, and $n = |K|$ key-terms are all isolated-terms. We can then immediately obtain $\Theta = iso(K) = K = \{k_1, k_2, ..., k_n\}$. That is, all the key-terms are merged into the frame of discernment (i.e., they are treated as 'atomic-terms'), and considered unrelated to each other. In many existing IR models, key-terms are dealt with in this way.

With the above notation, the hierarchical structure is an assembly of the individual hierarchical sub-structures: $H_R(K) = \big[H_{r_1}(K), H_{r_2}(K), ..., H_{r_s}(K)\big]$. The set of key-terms on at least one hierarchical sub-structure is denoted by $hie(K) = hie_{r_1}(K) \cup hie_{r_2}(K) \cup ... \cup hie_{r_s}(K)$. The sub-frame of discernment is denoted by $\Theta' = \Theta_{r_1} \cup \Theta_{r_2} \cup ... \cup \Theta_{r_s}$.

We give a detailed algorithm for establishing a frame of discernment and for expressing key-terms as subsets of the frame in [5].

3.9 Thesaurus Classes and Query Expansion

Our method reduces terms (i.e., $t \in T$) to their thesaurus classes: terms are replaced by representatives either synonymous (i.e., key-term $k \in K$), or equivalent (i.e., kernel-term $k' \in K'$). In order to clarify how thesaurus classes can be used to represent objects at a later stage, we introduce three further pieces of notation: $equ(k')$, $syn(k)$ and $cla(k')$.

For an arbitrary key-term $k \in K$, denote

$$syn(k) = \big\{t \;|k \Leftarrow t, t \in T\big\}$$

as the set of terms which are synonymous with key-term k.

For an arbitrary kernel-term $k' \in K'$, denote

$$equ(k') = \big\{k \;|k' \rightleftharpoons k, k \in K\big\}$$

as an *equivalent class* of key-terms which are equivalent to kernel-term k'.

Also, denote

$$\begin{aligned} cla(k') = \bigcup_{k \in equ(k')} syn(k) &= \bigcup_{k \in equ(k')} \big\{t \;|k \Leftarrow t, t \in T\big\} \\ &= \big\{t \;|k \Leftarrow t, t \in T; k' \rightleftharpoons k, k \in K\big\} \end{aligned}$$

as a *thesaurus class* of terms either synonymous with, or equivalent to, kernel-term k'.

Let us consider the following example.

Example 1. Suppose a user enters a query: q = 'birds'. Notice that term birds $\in K'$. Thus, from Table 7, we can write an equivalent class

$$equ(birds) = \{birds, domestic\text{-}birds, eggs, feathers, poultry\},$$

Also, from thesaurus $\mathcal{S}_{(k,t)}$ given in Table 2, we have

$$\begin{aligned} &syn(birds) = \{bird, birds\}, \\ &syn(domestic\text{-}birds) = \{domestic\text{-}birds\}, \\ &syn(eggs) = \{eggs\}, \\ &syn(feathers) = \{feathers\}, \\ &syn(poultry) = \{barnyard\text{-}birds, farmyard\text{-}birds, poultry\}. \end{aligned}$$

Hence, we can expand query q from a single term birds to a thesaurus class

$$\begin{aligned} cla(birds) = \{&bird, birds, domestic\text{-}birds, eggs, feathers, \\ &barnyard\text{-}birds, farmyard\text{-}birds, poultry\}. \quad \diamondsuit \end{aligned}$$

Query expansion, an important component in a retrieval system, has long been an effective technique to improve retrieval performance [4, 6, 8, 16, 19, 21, 31, 35]. Some good reviews of query expansion methods can be found in [10, 13].

In a practical IR environment, document collections are processed, and documents matching the user's query are displayed in real time. Documents that do not have any term matching the query are disregarded. Users of an IR system employing term matching as a basis for retrieval are faced with the challenge of expressing their queries with terms in the vocabulary of the documents they wish to retrieve. This difficulty is especially severe in extremely large, wide-ranging, full-text collections containing many different terms describing the same concept. The problem of term-mismatch has long been serious in IR.

The problem is more pronounced for short queries consisting of just a few terms related to the subject of interest: this can best be illustrated through the scenario of information search on the World Wide Web where users tend to enter very short queries. The shorter the query is, the less chance for important terms to co-occur in both relevant documents and the query. Hence a good way of matching terms is urgently needed. Query expansion is a process that modifies the original query representation so as to more precisely express the information needs.

Since our method can reduce different terms to their thesaurus classes, retrieval systems can achieve the effect of automatically expanding objects with thesaurus classes of the original query terms. The expansion may be expected to improve performance as it greatly increases matching between relevant document terms and query terms.

Consider several examples. If a user describes his information need as '*aviation school*', then relevant information indexed by terms *aeronautical engineering institute* might meet with retrieval failure—term mismatch arises from synonymous terms. In a retrieval based on a term '*cow*', the user might be also interested in the documents containing term *mammal*—term mismatch arises from narrower terms. A user enters a term '*planet*', he might be thinking of something like *Mercury* or *Venus*—term mismatch arises from broader terms. A user tries terms '*crime*' and '*murder*' when she desires to find some thrillers—term mismatch arises from the related terms. Thus, if the reader traces through all discussions given in this paper, it should become clear that the structure of the thesaurus classes embodies intuitive meaning, and this structure can be expected to resolve these term mismatching problems.

4 Retrieval Based on Evidential Theory

So far, we have concentrated on developing an effective method for tackling the problem of expressing key-terms as subsets of the frame of discernment. Before seeing how to apply our knowledge of expression to practical IR problems, we need to introduce evidential theory, which underpins the formal method proposed in this paper.

4.1 Evidential Theory

Evidential theory is by now a familiar one for many IR researchers. A detailed account of it can be found in [25]. Some general definitions applied in this study can be written as follows.

Let $\Theta = \{y_1, y_2, ..., y_{|\Theta|}\}$ denote a frame of discernment. Then the power set of Θ can be represented as $2^\Theta = \{Y_1, Y_2, ..., Y_{2^{|\Theta|}}\} = \{Y_i \mid Y_i =$

$\cup_{j=1}^{m_i}\{y_{i_j}\}$, $1 \leq m_i \leq |\Theta|, 1 \leq i \leq 2^{|\Theta|}\}$, that is, each element $Y_i \in 2^\Theta$ is a subset of Θ.

A function $m : 2^\Theta \rightarrow [0,1]$ is a *mass function* if there is a random subset variable Y over the *evidence space* 2^Θ, such that $m(Y)$ satisfies (1) $m(\emptyset) = 0$ and, (2) $\sum_{Y_i \subseteq \Theta} m(Y_i) = 1$. In evidential theory, masses are assigned to only those propositions (subsets) that are supported by evidence.

A function $bel: \ 2^\Theta \rightarrow [0,1]$ is a *belief function* if there is a random subset variable Y on Θ such that $bel(Y)$ satisfying: (1) $bel(\emptyset) = 0$, (2) $bel(\Theta) = 1$ and, (3) for any collection $A_1, A_2, ..., A_k$ ($k \geq 1$) of subsets of Θ, $bel(A_1 \cup A_2 \cup ... \cup A_k) \geq \sum_{I \subseteq \{1,2,...,k\}, I \neq \emptyset} (-1)^{|I|+1} bel(\cap_{i \in I} A_i)$. If we suppose $bel(Y) = \sum_{A \subseteq Y} m(A)$, then it is not difficult to verify that $bel(Y)$ is a belief function. We call $bel(Y)$ the belief function corresponding to mass function $m(Y)$.

Also, suppose $pls(Y) = \sum_{A \subseteq \Theta, A \cap Y \neq \emptyset} m(A)$, and call it the *plausibility function* corresponding to mass function $m(Y)$.

It can be verified that (1) $pls(Y) = 1 - bel(\Theta - Y)$, (2) $bel(Y) = 1 - pls(\Theta - Y)$ and, (3) $bel(Y) \leq pls(Y)$. Thus, $bel(Y)$ is also referred to as the *lower probability function*, and $pls(Y)$ as the *upper probability function*. The interval $[bel(Y), pls(Y)]$ is referred to as the belief interval. Here value $bel(Y)$ gives the degree to which the current evidence supports subset Y. The degree to which Y remains plausible is given by value $pls(Y) = 1 - bel(\Theta - Y)$. The difference $pls(Y) - bel(Y)$ represents the residual ignorance, $ign(Y) = pls(Y) - bel(Y)$, and is called the *ignorance function* corresponding to mass function $m(Y)$.

4.2 Object Representations and Mass Function

Having introduced the evidential functions, we move on to two other important issues—defining the representations of objects, and introducing agreement measures for ranking documents against a given query. We discuss the first in this subsection and the following; the second is discussed in Sec. 4.4.

In the IR context, we can simulate: (a) the frame of discernment by $\Theta = \{a_1, a_2, ..., a_{|\Theta|}\}$; (b) the evidence space by the evidence sub-space $K' = \{k'_1, k'_2, ..., k'_m\}$; (c) an evidence by an object (i.e., by the statistical information within the object, more precisely, by term weights obtained from the statistical information within the object); (d) a proposition by a statement "kernel-term k' appears". Thus, masses are assigned to only those kernel-terms that are supported by the object. By kernel-term k' supported by an object x (i.e., a document $x = d$ or query $x = q$), we mean here that it or, term(s) of thesaurus class $cla(k')$, appears in x.

Putting the above simulations together is equivalent to saying that each object x can be represented by a mass function $m_x(k')$ over sub-space K':

$$\left[m_x(k')\right]_{1\times m} = \left[m_x(k'_1), m_x(k'_2), ..., m_x(k'_m)\right],$$

where component $m_x(k')$ can be interpreted as indicating the strength of kernel-term k' or, a thesaurus class $cla(k')$, when supported by x. For instance, from Table 8, we have

$$\big[m_x(k')\big]_{1\times 8} = \big[m_x(animal), m_x(birds), m_x(cows), m_x(ducks),\\ m_x(geese), m_x(goats), m_x(hens), m_x(mammals)\big].$$

In order to estimate the strength of kernel-term k' supported by object x, suppose that term *weights*, $w_x(t)$, have been obtained, which are considered to reflect the importance of terms t ($\in V^x \subseteq T$) concerning x. Thus, the mass, $m_x(k')$, can be estimated from the weights of (i) k', (ii) synonymous terms of k', (iii) equivalent key-terms of k' and (iv) the synonymous terms of equivalent key-terms of k', in object x.

More specifically, for an arbitrary kernel-term $k' \in K'$, the mass is defined:

$$m_x(k') = \frac{\psi_x(k')}{N_x} = \frac{\psi_x(k')}{\sum_{k'\in(V^x\cap K')}\psi_x(k')},$$

where V^x is the set of terms appearing in object x; N_x, is the *normalization factor* of object x; function

$$\psi_x(k') = \sum_{t\in cla(k')} w_x(t) = \sum_{k\in equ(k')} \big(\sum_{t\in syn(k)} w_x(t)\big).$$

It can be seen that function $\psi_x(k')$ is the sum of weights, $w_x(t)$, of terms in the thesaurus class $cla(k')$. Thus, mass $m_x(k')$ is proportional to the sum. It is therefore evident that the design of weighting function $w_x(t)$ is crucial in determining retrieval performance. The effectiveness of the weighting function for reflecting the statistical importance of a term in respect to individual objects, has been investigated extensively in the literature [1, 4, 12, 14, 20, 22, 27, 28, 29, 31].

Let us see an example below, which may help to clarify the above idea and assist in understanding the computation involved in function $m_x(k')$.

Example 2. Let us return to Example 1. Suppose we are given the weights of terms in document d as follows.

$w_d(birds) = .646, \quad w_d(bird) = .421,$

$w_d(domestic\text{-}birds) = .0,$

$w_d(eggs) = .285,$

$w_d(feathers) = .17,$

$w_d(barnyard\text{-}birds) = .01, \quad w_d(farmyard\text{-}birds) = .0, \quad w_d(poultry) = .0,$

and so on. Suppose also that the normalization factor of d is $N_d = 6.52$. Then we arrive at the mass for kernel-term $k' = birds$ or, the thesaurus class $cla(birds)$:

$$m_d(birds) = \frac{.646 + .421 + .0 + .285 + .17 + .01 + .0 + .0}{6.52} \approx .235. \quad \diamondsuit$$

Notice that all terms in a given object x may not be partitioned into distinct thesaurus classes (that is, each term $t \in V^x$ may be classified into at least one thesaurus class), and that the normalization factor in the estimation of $m_x(k')$ is therefore given by $N_x = \sum_{k' \in (V^x \cap K')} \psi_x(k')$, rather than simply by $N_x = \sum_{t \in V^x} w_x(t)$.

Notice also that there is no necessity to design a weighting function in advance for estimating mass $m_x(k')$. The estimation can simply be made using occurrence frequencies of terms. In this case, $w_x(t) = f_x(t)$.

4.3 Object Representations and Other Evidential Functions

Objects can also be represented by the belief and plausibility functions. In order to compute functions *bel* and *pls*, the narrower relation and related relation between kernel-terms are involved, and the relations take their mathematical meanings: for two arbitrary kernel-terms $k'_i, k'_j \in K'$, we can consider their semantic relations $k'_i \subset k'_j$ and $k'_i \cap k'_j$ by the set relations and operations of their corresponding subset-expressions. We can clarify this idea by considering the example below.

Example 3. Suppose that the mass functions for documents d_1, d_2, d_3, d_4 and queries q_1, q_2 are obtained. These are given in Table 9.

Table 9. Mass functions

Functions	Kernel-Terms	d_1	d_2	d_3	d_4	q_1	q_2
mass	animals	0.125	0.000	0.133	0.300	0.000	0.200
functions	birds	0.560	0.445	0.867	0.000	1.000	0.600
	mammals	0.315	0.555	0.000	0.700	0.000	0.200

and $m_x(cows) = m_x(ducks) = m_x(geese) = m_x(goats) = m_x(hens) = 0$ for $x = d_1, d_2, d_3, d_4, q_1, q_2$.

Then, the corresponding belief and plausibility functions are calculated, and results are given in Table 10.
For instance, for kernel-term $k' = birds$ in document d_1, we have,

$$\begin{aligned} bel(birds) &= \sum_{k' \subseteq birds} m_{d_1}(k') = m_{d_1}(birds) = 0.560; \\ pls(birds) &= \sum_{k' \subseteq \Theta; k' \cap birds \neq \emptyset} m_{d_1}(k') = m_{d_1}(animals) + m_{d_1}(birds) \\ &= 0.125 + 0.560 = 0.685, \end{aligned}$$

where, from Table 5, all kernel-terms satisfying $k' \subseteq birds$ are

$birds \rightleftharpoons \{ducks, geese, hens\}$,

Table 10. Belief and plausibility functions

Functions	Kernel-Terms	d_1	d_2	d_3	d_4	q_1	q_2
belief	animals	1.000	1.000	1.000	1.000	1.000	1.000
functions	birds	0.560	0.445	0.867	0.000	1.000	0.600
	mammals	0.315	0.555	0.000	0.700	0.000	0.200
plausibility	animals	1.000	1.000	1.000	1.000	1.000	1.000
functions	birds	0.685	0.445	1.000	0.300	1.000	0.800
	mammals	0.440	0.555	0.133	1.000	0.000	0.400

$$ducks \rightleftharpoons \{ducks\}, \quad geese \rightleftharpoons \{geese\}, \quad hens \rightleftharpoons \{hens\};$$

and all kernel-terms satisfying $k' \cap birds \neq \emptyset$ *are*

$$animals \rightleftharpoons \{cows, ducks, geese, goats, hens\},$$
$$birds \rightleftharpoons \{ducks, geese, hens\},$$
$$ducks \rightleftharpoons \{ducks\}, \quad geese \rightleftharpoons \{geese\}, \quad hens \rightleftharpoons \{hens\}. \qquad \diamondsuit$$

4.4 Agreement Measures

In the foregoing, we discussed the definition of the representations of objects based on evidential functions. In this subsection, for ranking documents against a given query, we further discuss agreement (similarity) measures, over the evidential representations.

Suppose that document d and query q can be represented by $m_d(k')$ and $m_q(k')$, respectively. Our method involves computing belief $bel(m_d = m_q)$ and plausibility $pls(m_d = m_q)$. A study on the computation can be found in [3]. The *agreement measures* between $m_d(k')$ and $m_q(k')$ are defined by

$$\begin{aligned} bel(m_d = m_q) &= \sum_{k' \subseteq K'} m_d(k') m_q(k'), \\ pls(m_d = m_q) &= \sum_{k'_i, k'_j \subseteq K';\ k'_i \cap k'_j \neq \emptyset} m_d(k'_i) m_q(k'_j) \\ &= \sum_{k'_j \subseteq K'} m_q(k'_j) \Big(\sum_{k'_i \subseteq K';\ k'_i \cap k'_j \neq \emptyset} m_d(k'_i) \Big). \end{aligned}$$

Obviously, the *lower agreement*, $bel(m_d = m_q)$, measures the belief that document kernel-terms and query kernel-terms are equal (i.e., their subset-expressions are the same); the *upper agreement*, $pls(m_d = m_q)$, measures the plausibility that document kernel-terms and query kernel-terms are related (i.e., their subset-expressions have non-empty intersections).

Further, let us consider the belief interval

$$\big[bel(m_d = m_q),\ pls(m_d = m_q) \big].$$

As mentioned, $bel(m_d = m_q)$ gives the degree to which the current evidence supports $m_d(k') = m_q(k')$. The degree to which evidence $m_d(k') = m_q(k')$ remains plausible is given by $pls(m_d = m_q) = 1 - bel(m_d \neq m_q)$.

The following example illustrates the computation involved.

Example 4. Let us return to Example 3. We there gave mass functions for documents d_1, d_2, d_3, d_4 and queries q_1, q_2. Thus, for d_1 and q_1, we have

$$\begin{aligned}
bel(d_1 = q_1) =& m_{d_1}(animals)m_{q_1}(animals) + m_{d_1}(birds)m_{q_1}(birds) + \\
& m_{d_1}(mammals)m_{q_1}(mammals) \\
=& 0.125 \times 0 + 0.560 \times 1 + 0.315 \times 0 = 0.560; \\
pls(d_1 = q_1) =& m_{q_1}(birds)\big(m_{d_1}(animals) + m_{d_1}(birds)\big) \\
=& 1 \times \big(0.125 + 0.560\big) = 0.685,
\end{aligned}$$

also, we have

$$\begin{aligned}
bel(d_2 = q_1) = 0.445, \quad bel(d_3 = q_1) = 0.687, \quad bel(d_4 = q_1) = 0; \\
pls(d_2 = q_1) = 0.445, \quad pls(d_3 = q_1) = 1, \quad pls(d_4 = q_1) = 0.3.
\end{aligned}$$

Similarly, for d_1 and q_2, we have

$$\begin{aligned}
bel(d_1 = q_2) =& m_{d_1}(animals)m_{q_2}(animals) + m_{d_1}(birds)m_{q_2}(birds) + \\
& m_{d_1}(mammals)m_{q_2}(mammals) \\
=& 0.125 \times 0.2 + 0.560 \times 0.6 + 0.315 \times 0.2 = 0.424; \\
pls(d_1 = q_2) =& m_{q_2}(animals)\big(m_{d_1}(animals) + m_{d_1}(birds) + \\
& \qquad m_{d_1}(mammals)\big) + \\
& m_{q_2}(birds)\big(m_{d_1}(animals) + m_{d_1}(birds)\big) + \\
& m_{q_2}(mammals)\big(m_{d_1}(animals) + m_{d_1}(mammals)\big) \\
=& 0.2 \times \big(0.125 + 0.560 + 0.315\big) + 0.6 \times \big(0.125 + 0.560\big) + \\
& 0.2 \times \big(0.125 + 0.315\big) = 0.699,
\end{aligned}$$

also, we have

$$\begin{aligned}
bel(d_2 = q_2) = 0.378, \quad bel(d_3 = q_2) = 0.5468, \quad bel(d_4 = q_2) = 0.2; \\
pls(d_2 = q_2) = 0.578, \quad pls(d_3 = q_2) = 0.8266, \quad pls(d_4 = q_2) = 0.58.
\end{aligned}$$

Finally, ranking documents by the belief interval, we respond to users' queries as follows.

For query q_1, the response is

$$d_3[0.687, 1.0] \succ d_1[0.560, 0.685] \succ d_2[0.445, 0.445] \succ d_4[0, 0.3].$$

For query q_2, the response is

$$d_3[0.5468, 0.8266] \succ d_1[0.424, 0.699] \succ d_2[0.378, 0.578] \succ d_4[0.2, 0.58],$$

where $d_i[bel_i, pls_i] \succ d_j[bel_j, pls_j]$ is explained as "document d_i is more in agreement with the query than document d_j". ♢

Conclusion and Further Work

The ability to formally express term semantic relations is a core issue in IR. The problems for the expression are how to establish the frame of discernment, and how to express key-terms as subsets of the frame. The problems lead to many other IR problems as pointed out repeatedly in the literature. Solution of the problems is a technical barrier to applying mathematical tools, especially evidential theory, to IR. In this study, we focus on the problems, and present a method for establishing the frame of discernment and for deriving subset-expressions of key-terms. Then, we propose a novel method for representing documents and queries based on evidential functions, and for ranking documents against a given query. A central aim of this study is to treat the semantic relations between terms and incorporate the relations into the retrieval strategies for more effective retrieval.

A key-term, if polysemous, may be expressed by different frame subsets corresponding to different roots. In IR, it is difficult to automatically determine which meaning is being used in the context. Almost all existing IR methods suffer from the same problem. Thus, it is hard to determine into which thesaurus class a polysemous term should be placed. This paper does not deal with how the class is determined; it is left as a significant subject for further study.

Thesaurus class methods can be regarded as *recall*[1] improving devices. The thesaurus classes of query terms may be expected to retrieve more relevant documents because extra 'related' terms are added to the query when the thesaurus classes are assigned to the query instead of single terms. However, if terms included in a thesaurus class have high *document frequencies*[2], then the addition of these terms would be likely to lead to unacceptable losses in *precision*[3]. For this reason, some studies suggest that thesaurus classes should be formed only from those terms which have low document frequencies, [36] for instance. This interesting issue needs to be investigated in further work.

We intend to develop an experimental investigation into the performance of our method.

Acknowledgements

This research was supported in part by both EPSRC and Microsoft.

[1] The proportion of relevant documents actually retrieved in answer to a query.

[2] The number of documents in a collection in which a term appears.

[3] The proportion of retrieved documents actually relevant to the query.

References

1. G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, 20(4):357–389, 2002.
2. A. Bookstein. Relevance. *Journal of the American Society for Information Science*, 30:269–273, 1979.
3. D. Cai. Extensions and applications of evidence theory. In *Soft Computing for Risk Evaluation and Management: Applications in Technology, Environment and Finance*, volume 76, pages 73–93, New York, 2001. Physica-Verlag, Heidelberg.
4. D. Cai. *$\mathcal{I}f\mathcal{D}$—Information for Discrimination.* PhD thesis, University of Glasgow, Glasgow, Scotland, 2004.
5. D. Cai and C. J. Van Rijsbergen. An algorithm for modelling terms. Technical Report TR-2005-190, Department of Computing Science, University of Glasgow, 2005.
6. D. Carmel, E. Farchi, Y. Petruschka, and A. Soffer. Automatic query refinement using lexical affinities with maximal information gain. In *Proceedings of the 25th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 283–290, 2002.
7. C. Carpineto, R. Mori, and G. Romano. Informative term selection for automatic query expansion. In *The 7th Text REtrieval Conference (TREC-7)*, pages 363–369. NIST Special Publication, 1998.
8. C. Carpineto, R. D. Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001.
9. W. S. Cooper. A definition of relevance for information retrieval. *Information Storage and Retrieval*, 7:19–37, 1971.
10. E. N. Efthimiadis. Query expansion. *Annual Review of Information Systems and Technology*, 31:121–187, 1996.
11. N. Fuhr. Optimum polynomial retrieval functions based on the probability ranking principle. *ACM Transactions on Information Systems*, 7(3):183–204, 1989.
12. N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992.
13. S. Gauch, J. Wang, and S. M. Rachakonda. A corpus analysis approach for automatic query expansion and its extension to multiple databases. *ACM Transactions on Information Systems*, 17(3):250–269, 1999.
14. D. Harman. An experimental study of factors important in document ranking. In *Proceedings of the 9th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 186–193, 1986.
15. S. P. Harter. Psychological relevance and information science. *Journal of the American Society for Information Science*, 43(9):602–651, 1992.
16. M. A. Hearst. Improving full-text precision on short queries using simple constraints. In *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval*, 1996.
17. J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 111–119, 2001.

18. M. Lalmas. Dempster-Shafer's theory of evidence applied to structured documents: modelling uncertainty. In *Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 110–118, 1997.
19. M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 206–214, 1998.
20. S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.
21. G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.
22. G. Salton and C. S. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, 29(4):351–372, 1973.
23. T. Saracevic. Relevance reconsidered '96. In *Proceedings of the 2nd International Conference on Conceptions of Library and Information Science*, pages 201–218, 1996.
24. S. Schocken and Hummel. On the use of Dempster-Shafer model in information indexing and retrieval applications. *International Journal of Man-Machine Studies*, 39:843–879, 1993.
25. G. Shafer. *A Mathematical Theory of Evidence.* NJ: Princeton University, Princeton, 1976.
26. W. T. Silva and R. L. Milidiú. Belief function model for information retrieval. *Journal of the American Society for Information Science*, 44(1):10–18, 1993.
27. K. Sparck Jones. A statistical interpretation of term specificity and its application to retrieval. *Journal of Documentation*, 28(1):11–21, 1972.
28. H. Turtle and W. B. Croft. Inference networks for document retrieval. In *Proceedings of the 13th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 1–24, 1990.
29. C. J. Van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33(2):106–119, 1977.
30. C. J. Van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 39:481–485, 1986.
31. C. J. Van Rijsbergen, D. J. Harper, and M. F. Porter. The selection of good search terms. *Information Processing & Management*, 17:77–91, 1981.
32. S. K. M. Wong and Y. Y. Yao. A probability distribution model for information retrieval. *Information Processing & Management*, 25(1):39–53, 1989.
33. S. K. M. Wong and Y. Y. Yao. A probabilistic inference model for information retrieval. *Information Systems*, 16(3):301–321, 1991.
34. S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalised vector space model in information retrieval. In *Proceedings of the 8th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25, 1985.
35. J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.
36. C. T. Yu and G. Salton. Effective information retrieval using term accuracy. *Journal of the Association for Computing Machinery*, 20(3):135–142, 1977.