A Semantic Similarity Method for Products and Processes

June, ２０１３

ＤＯＣＴＯＲ　ＯＦ　ＥＮＧＩＮＥＥＲＩＮＧ

Suriati Akmal

ＴＯＹＯＨＡＳＨＩ　ＵＮＩＶＥＲＳＩＴＹ　ＯＦ　ＴＥＣＨＮＯＬＯＧＹ

Abstract

Due to the complexity of customer requirements coupled with drastic technological changes, development of products and processes is becoming increasingly knowledge intensive. Specifically, retrieving product and process information and making effective use of it requires similarity measures.

Similarity measures are concerned with quantifying of the likeness of the things that are compared. Similarity measures have been practically applied in a wide variety of fields ranging from data mining, case-based reasoning system, image interpretation and pattern recognition. Several researchers have proposed similarity measures that evaluate the likeness between values of numeric properties. However, in many applications some attributes are non-numeric. One solution is to use syntactic similarity measures that calculate the similarity between two words. However, syntactic approaches are limited as they fail to produce good matches when confronted with the meaning associated to the words they compare.

To overcome the above drawbacks semantic similarity measures are been investigated. A semantic similarity measure is a function that quantifies the degree of likeness between two things based on the meaning associated to each thing being compared. This research contributes to the field of semantic similarity measures for products and processes. A novel approach has been proposed in this research, based on Formal Concept Analysis (FCA) and a set of criteria for the characterization of products and processes called Formal Attribute Specification Template (FAST).

This research focuses on countable objects that are represented in terms of their physical aspects and processes in which they are involved. Processes can be intentional or unintentional. In an intentional process, a

particular objective is accomplished. Unintentional processes include natural phenomena and undesired processes such as harmful explosions or fires.

The proposed approach is composed of semantic similarity measures that compare classes in a taxonomy obtained with FCA and a template for the specification of formal attributes (FAST).

The semantic similarity measures of the proposed approach compare classes of products or processes. The comparison is based on the assumption that the more common attributes that are shared by two classes the more similar they are. Therefore, a class is 100% similar to another class if both classes have exactly the same attributes. In particular, the attributes are the formal attributes from the FCA. For this purpose, several similarity equations are investigated in this research by using formal attributes as the sets they compare.

Class taxonomies are defined by means of the subclass relation. A class is a subclass of another class if every member of the subclass is also a member of the super class. Formal Concept Analysis (FCA), which is a method based on applied lattice and order theory, is selected as the taxonomy generator.

FAST helps to describe the formal attributes common to all members of a given class that distinguish them from members of another class. The product formal attributes are expressed in terms of its mereological and topological structure and its involvement with one or more processes. The process formal attributes are expressed in terms of: (1) objects that are always changed by the process (a.k.a inputs); (2) objects that are always produced by the process (a.k.a outputs); (3) participating physical objects (including locations, agents, and performer) other than inputs and outputs; (4) sub-activities that compose the process (a.k.a sub-activities).

The proposed approach was evaluated against edge-counting and information-based similarity measures. In order to quantify the efficacy of each similarity measure, the degree of correlation with human judgment was used. The results of the evaluation show that the proposed approach performed better than existing similarity measures.

The proposed approach is illustrated with two case studies. The first case study demonstrates the use of FAST for the construction of an ontology for machining processes. The resulting machining processes ontology was evaluated and compared against a third-party ontology. The degree of correlation with Internet-search engine using the value of the Normalized Google distance evaluated the accuracy of each ontology. The results of evaluation show that the ontology obtained with FAST is slightly better than the existing ontology. It was also found that FAST can provide the design rationale of the ontology.

The second case study focused on the application of the proposed semantic similarities for selecting the service strategy for Product-Service systems (PSS) at the early stage of design. It is often the case that the PSS designer is faced with limited amount of knowledge at the early stage of design. One solution is to use the case-based reasoning (CBR) system to facilitate the service strategy selection in which PSS design problems are solved by using or adapting previously obtained design solutions. Existing CBR-systems use numerical similarity measures to search the relevant solution to the problem to be solved. In this case study, a semantic CBR-system was developed by incorporating product-class-comparison based on the proposed semantic similarities. The results of evaluation show that the proposed approach proved useful when some details of information are not available.

Acknowledgement

During my stay in Toyohashi, Japan I was facilitated by many people and I would like to express my gratitude for their support. The first and foremost thanks to God who blessed me with such a great opportunity to learn and granted me success in this long-time endeavor “*Alhamdulillah*”.

I owe a debt of gratitude to my research advisor, Associate Professor Dr. Rafael Batres, for providing me with a wealth of knowledge, sharing with the times of struggles, and his consistently guiding and inspiring me throughout this endeavor.

I am thankful to Prof. Yoshiaki Shimizu for his valuable input and time as well as accepting and inducting me in Industrial System Engineering Lab. I would like to thank to Prof. Masaki Aono and Associate Prof. Tetsuo Miyake for serving on my doctoral committee. I greatly appreciate for their assistance and input. I also want to thank to all members of Industrial System Engineering Lab for sharing their knowledge, wisdom and inspiration.

I am very obliged to the Ministry of Higher Education of Malaysia and Universiti Teknikal Malaysia Melaka for awarding me with the scholarship to study at Toyohashi University of Technology.

I owe the biggest acknowledgement to my parents, siblings, husband and sons for their understanding, encouragements, patience, love and supports while I am occupied in my research work.

Glossary

Case-based reasoning system (CBR)	is one way to solve problem in which problems are solved by using or adapting previously design solutions to old problems.
Edge-based measure	similarity measure that relies on the use of subclass links (edges) between classes.
False positive	is the errors of retrieving results that are not fulfill the condition.
Feature-based measure	similarity measure that take into account the features that are common to two classes.
Formal Concept Analysis (FCA)	is a method based on applied lattice and order theory that can be used to generate lattice
Information-based measure	similarity measure that depends on information content.
Mereology	expresses the part-whole relations of an object
Mean Absolute Percentage Error (MAPE)	is a measure to determine the accuracy of a series in statistics.
Ontologies	describes a shared understanding about the meanings of objects by means of classes of objects, taxonomy, relation between classes, properties of objects in each class and axioms.
OWL	is a language for processing web information. (http://www.w3schools.com)
Process	is "an operation or a series of operations" that "cause a physical or chemical change in a substance or mixture of substances".

Product	is a something that is the result of a process.
Product-service system	is a mix of both products and services aimed at better sustainability of both production and consumption.
Root Mean Squared Error (RMSE)	is a measure that determine the differences between predicted and observed value.
Similarity	is a term to enclose whether two things, or two situations are similar or dissimilar.
Semantic similarity	is a term to quantify the degree of likeness between two things based on the meaning.
Synset	is a collection of one or more words and phrases ("collocations") collectively referred to as "word forms" that can all share the similar meaning (synonym). (http://lyle.smu.edu/~tspell/jaws/doc /edu/smu/tspell/wordnet/Synset.html)
Topology	refers to the connectivity between objects

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1. Similarity measures

Generally, "similarity" is a term to enclose whether two things, or two situations are similar or dissimilar. According to [1], similarity plays an important role in studies of theories of cognition and how people make comparisons. According to [2], "similarity is a core element for learning, knowledge and thought, for only our sense of similarity allows us to order things into kinds so that these can function as stimulus meanings reasonable expectation depends on the similarity of circumstances and on our tendency to expect that similar cause will have similar effects".

According to Holt [3], similarity is important for humans to understand the existence of objects, structure and actions together with their connections in reality. The degree to which we determine if two things are similar is both intuitive and based on our knowledge. For example, when an individual plans to use a toaster on the dining table as shown in Fig. 1-1, he or she will imagine the result of using the toaster, which is related to the function performed by the product. The memory, which has some prior knowledge, organizes the information and somehow translates it into associations such as bread and toaster, toasted bread and toaster. Based on memory of the past, a toaster is always used to toast the bread. When comparing a toaster and let's say a pizza oven, we are inclined to look at common aspects such as the use of heat to produce warm and somehow crispy bread.

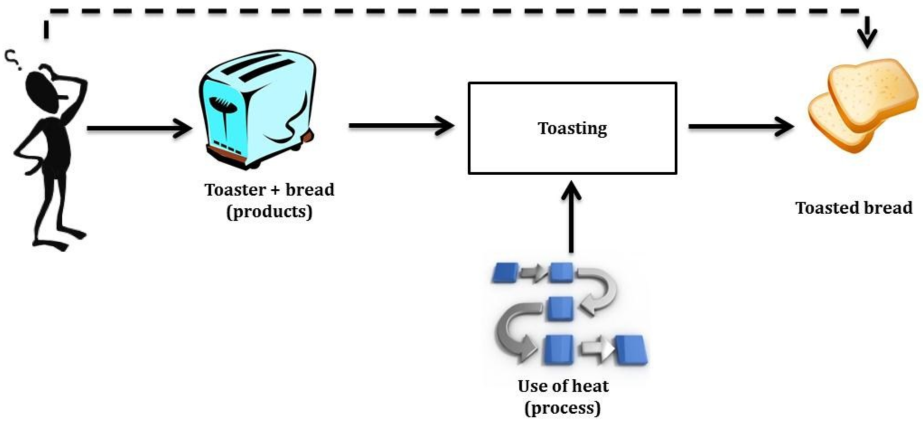


Fig. 1-1 The function of a product are the desired behavior of a product

In addition, if only a few objects are given, it is easy for a human to identify how close two objects are by finding their common aspects. However, it becomes more complex for a large numbers of objects.

Therefore many practical applications require computational similarity measures. As a matter of fact, the computational approaches for measuring similarity that emphasize imitate the human ability of assessing similarity between two things date back to [4].

The past decade has seen the development of computational similarity measure that are based on geometric models that assume objects are represented by points in some coordinate space. The similarity of these approaches is calculated by the metric distance between respective points. However, one of major problems with this approach is the inappropriateness to represent the dimensional representation for qualitative properties of thing being compared [5].

In recent years, there has been an increasing interest in using feature comparison to quantify the degree of likeness of the things that are compared. Tversky and Gati [5] identified similarity as a function that quantifies the degree to which two sets of features match each other. They proposed a similarity that considers both common and distinct features which are known as the contrast model. Their contrast model explained that the similarity should not be viewed as a symmetric relation such as a is similar to b than b is similar to a. For example, people say “the son resembles the father” rather than “the father resembles the son”; “the portrait resembles the person” and not “the person resembles the portrait”. Russel and Norvig [6] defined similarity as an evaluation of the common intrinsic features shared by two things. The intrinsic features are the important features that belong to a thing. If the thing is described without this feature, the meaning of the thing is incomplete.

Similarity measures play an important role in information retrieval process, information extraction, information integration and other applications involving comparison two things. In an information retrieval system, determining the optimal match between a queries and stored information is the fundamental operation that highly depends on similarity measures. In such systems, the retrieved information is sorted in order of their decreasing similarity. High-ranked information is likely to have similar properties to the query.

Also, similarity measures can be used for problem solving. For example the case-based reasoning systems use reasoning that draw conclusion by similarity. It imitates human reasoning for solving a problem by making use of the previous experiences.

Similarity measures in pattern recognition are used for classifying sets of objects into classes. Similar objects are grouped within the same cluster and dissimilar objects in different cluster.

In numerous multimedia processing systems and applications, assessment of image similarity is important for image copy detection, retrieval and recognition problem. Similarity measures are used to interpret the characteristics of an image that compared against its variations versions such as contrast/brightness-variation.

Although numerous concept of similarity measures have been applied in many scientific fields and presented in many forms and interpretations, they all have in common of comparing two objects, two situations, for various reasons including knowledge, biases and goals [7].

Most similarity measures evaluate differences between values of numeric attributes such as in the numerical difference between two given diameter values. However, many applications require non-numeric similarities as well. For example, case-based reasoning systems for the conceptual design of products and processes must be developed to work with a limited knowledge about the products and processes.

Nearly all of non-numeric similarity measures are based on syntactic grounds. For example, the Levenshtein distance [16], [17] can be used to calculate the similarity between two words, in terms of the minimum number of operations that are needed to transform one of the words into the other. However, from the point of view of the meaning of the words that are compared, existing syntactic similarity-measures often result in incorrect matches.

Semantic similarity measures can be used in order to overcome the limitations of syntactic approaches. A semantic similarity is a function that assigns a numeric value to the similarity between two classes of objects based on the meaning associated to each of the objects [18]. For a review of semantic similarity metrics, the reader is referred to the paper of Cross and Hu [19].

Recently, the use of ontologies for evaluating similarity has been reported in the literature [20], [21]. Ontologies are formal models that use mathematical logic to disambiguate and define classes of things [22]. Specifically, ontologies describe a shared and common understanding of a domain in terms of classes, possible relations between things, and axioms that constrain the meaning of classes and relations [23]. A class represents a set of things that share the same attributes. A relation is used to represent a relationship among two or more things. Examples of relations are less than, connected to, and part of. Class taxonomies are defined by means of the subclass relation. A class is a subclass of another class if every member of the subclass is also a member of the super class. Axioms are typically represented as logic constructions that formally define a given class or relation.

Most semantic similarities are defined in terms of the number of edges between the classes that they compare. The research to date has tended to focus semantic similarities that are defined in terms of features but uses synsets for the comparison between words rather than classes. Most of the existing similarities measures use a large database such as WordNet for general purpose and Mesh for medical purpose for evaluating the word comparison.

In this thesis, a comprehensive approach towards the similarity measures for products and processes information that can deal with non-attribute information is developed.

1.2. Why are similarity measures necessary for products and processes?

A product is defined as something that is the result of a process. On the other hand, typical chemical engineering textbooks define a process as “an operation or a series of operations” that “cause a physical or chemical change in a substance or mixture of substances” [8]. Textbooks also explain that processes commonly have several steps, each of which represents a specific physical or chemical change. Such definitions assume that during the realization of a process, a particular objective is accomplished. In other words, according to these definitions, a process has a design intention.

However, unintentional phenomena are also of concern to engineers. For example, explosions (such as those that result in property damage) may happen as a result of an abnormal situation rather than a well-designed series of steps. Despite differences related to whether an objective is involved or not, both intentional and unintentional processes share the ability to transform material or energy through one or more changes. This research addresses both kinds of processes.

Due to the complexity of customer requirements coupled with drastic technological changes, development of products and processes is becoming increasingly knowledge intensive. It brings about change in the way industries organize products and processes. The market demands industries to effectively manage the know-how about products and processes as a means to differentiate the business competitions. Information about products and processes has to be considered as a rather special resource [9]: it does not get lost when it is used, and the costs for generating and procuring information are high compared to the costs for its storage and dissemination.

Product development which is a multi-disciplinary in nature requires a variety of product life-cycle knowledge [10]. Specifically, design teams face a considerable challenge in making effective use of increasing amounts of information that is stored in several information systems. Also, it is often the case that product designers can reuse past designs rather than designing from scratch [11]. Thus it would be very important to have the ability to retrieve product data.

As mentioned above, information retrieval consists of translating and matching a query against a set of information objects. The information retrieval system responds to the query using a given algorithm and a similarity measure. Particularly, information retrieval plays an important role in areas such as product family design [12], product embodiment, and detailed design [13]. Shah et al. [14] present a combination framework that consists of software engineering, data engineering and knowledge engineering and design theory.

In order to support product and process information retrieval and reuse, some authors suggest the use of case-based reasoning (CBR) in which design problems are solved by using or adapting previous design solutions [13], [15].

A CBR system is composed of domain knowledge, a case base and a search mechanism based on a similarity measure. Domain knowledge refers to knowledge about the features of the different objects or entities that a case is about. A case base contains a set of cases, each of which describes a problem and a solution to the problem. The problem is typically defined in terms of specific features of objects. Finally, a similarity measure quantifies the differences that exist between objects [7]. CBR uses similarity measures to identify cases which are more relevant to the problem to be solved.

1.3. Overview of the proposed approach

The objective of this thesis is to develop a more effective semantic similarity method for products and processes. The proposed approach is composed of semantic similarity measures that compare classes in a taxonomy obtained with Formal Concept Analysis (FCA) and a template for the specification of formal attributes.

The proposed approach is based on two main pillars. One is a semantic similarity measure based on Formal Concept Analysis (FCA). The semantic similarity measure of the proposed approach compare classes of products and processes. The semantic similarity measure is emphasized on the common formal attributes that are obtained from FCA. It is a method based on applied lattice and order theory, is selected as the taxonomy generator. The underlying principle in this research is that if a class represents a set of things that share the same attributes (such as a class in a taxonomy), we can state that a class is equivalent to another class if both classes have exactly the same attributes. This implies that the more common attributes that are shared by two classes the more similar they are. For this purpose, several similarity equations are investigated in this research by using formal attributes as the sets they compare. It became clear that the sets of features could be replaced with sets of formal attributes from the FCA.

The second pillar is a new way to specify the formal attributes required by FCA. This method is referred to as Formal Attribute Specification Template (FAST). FAST identifies the product formal attributes by considering its mereological and topological structure and its involvement with one or more processes. FAST also identifies the formal attributes of processes.

The proposed semantic similarity method consists of two steps: taxonomy generation and similarity calculation.

FAST is used in the taxonomy generation for formal attribute identification which is later used in FCA to generate a lattice. The resulting lattice and formal attribute information obtained with FCA are later used to create a class hierarchy.

In the second step, similarity between two classes of this taxonomy is calculated using a semantic similarity measure , in which the taxonomy structure and formal attribute information are used as input. For this purpose, the edge-counting and information-based similarity measures were used to evaluate and compare against the proposed approach. In order to quantify the efficacy of each similarity measure, the degree of correlation with human judgment and NGD similarity were used. The results of the evaluation show that the proposed approach performed better than existing similarity measures.

1.4. Thesis outline

The remainder of this thesis consists of six chapters followed by bibliography. Topics discussed in every chapter are as follows:
Chapter 2 contains a comprehensive description on concept of semantic similarity and presents an overview of common semantic similarity measures.

Chapter 3 describes the contribution of this research. This chapter introduces the semantic similarity equation and the Formal Attribute Specification Template (FAST) of the proposed approach.

In chapter 4, the proposed approach is evaluated and compared against the existing similarity measures. The correlation of each similarity score is compared against the human similarity ratings.

Chapter 5 describes the application of the proposed approach for constructing machining process ontology. The resulted machining ontology was evaluated and compared against a third-party ontology. The degree of correlation with Internet-search engine using the value of the Normalized Google distance evaluated the accuracy of each ontology.

Chapter 6 demonstrates a real-world application in product-service system. In this research, the existing CBR systems that use numerical similarity measures for service strategy selection for product-service system (PSS) was modified by incorporating the product-class-comparison based on the proposed semantic similarities. The results of evaluation show that the proposed approach proved useful when some details of information are not available.

Chapter 7: summarizes the main contribution of this thesis and draws conclusions on the conducted research. Finally, some possible improvements are discussed.

Chapter 2

LITERATURE REVIEW

This chapter presents a brief review on semantic similarity, and describes the existing approaches to determine the similarity of two things in a hierarchical structure. Finally, the role of FCA for building the class hierarchy is explained.

2.1. Semantic similarity measures

Semantic similarity is used for providing necessary semantic context information for information retrieval applications and in a variety of applications including word sense disambiguation, classification and ranking, detection of redundancy, and detection of malapropisms [24], [25]. To date, the existing similarity measures in the literature proposed for measuring similarity in a taxonomy between words. Some of researchers take the advantage of combination of taxonomy with corpus to measure the similarity.

In this thesis, we use the term semantic similarity measures to denote the quantifying of the degree of likeness between two things based on the meaning associated to each thing being compared.

In the literature, the term similarity and relatedness are very often used interchangeably. However, there is a difference between them. The term similarity is concerned about likeness, while relatedness seeks to determine the relationship between two things.

Resnik [30] defined similarity as a special case of relatedness. For example, the bicycle and cyclist appear to be more closely related than the terms car and bicycle, even though car and bicycle are more similar. In this example, "bicycle is related to cyclist" is based on the functional relationship such as cyclist rides bicycle. The notion of relatedness emphasizes on the various kind of lexical relationship such as meronymy (bicycle-wheel) and antonymy (large-small), or functional relationship or frequent association (camel-desert) [31].

Another common term is distance. Distance is inversely proportional to similarity. Cross defined distance is the inverse of both similarity and relatedness. The less distance between two things, increase the similarity between two things.

The computational approach regarding similarity requires a consistent type of relation between things being compared such as the hierarchical relation (i.e. is-a, part-whole), associative relation (i.e. cause-effect) and equivalence relation (i.e. synonymy) [26]. Among these approaches, the hierarchical relation is a well-studied technique and has been widely applied in computing the similarity between two things. Using this relation, it shows how well the computational models imitate the human cognitive view of classification [26].

When the things that are compared correspond to classes in a taxonomy, a semantic similarity is a function that assigns a numeric value to the similarity between two classes of objects [32]. The classes in a taxonomy are related by means of a subclass relation also known as *is-a* relation or *subsumption* relation. A class C_1 is said to be a subclass of C_2 (C_2 is a superclass of C_1) if all the members of C_1 are also members ofC_2. It is worth mentioning that when talking about 'the similarity between two classes C_1 andC_2,' in reality the comparison is about two generic members of those classes $x \in C_1$ and $y \in C_2$. Thus the similarity between two classes is based on how closely they are related in the taxonomy.

According to the type of knowledge sources that used to assess the semantic similarity assessment, the semantic similarity measures can be divided into three families of functions: [1] those based on the taxonomical structure (Section 2.1.1); [2] those relying on the information content (Section 2.2.1); [3] those based on the set of features (Section 2.1.3).

2.1.1. Edge-based measures

Edge counting measures are based on the distance between two classes. The most primitive edge-based similarity measure is that which computes the distance of the shortest path length between two classes [33]. The distance can be measured by the number of edges that links the two classes via *is-a* links in the taxonomy. The shorter the path from one node to the other, the more similar they are. For example, the length of the shortest path between node K and L in Fig. 2-1 is 4. The path length between E and B is 4. The similarity between these two cases is the same according to the path-length measures. However, in a more realistic scenario, similarities between any two adjacent nodes are not necessarily equal. To address this limitation some authors assign weights to each edge that connects two classes.

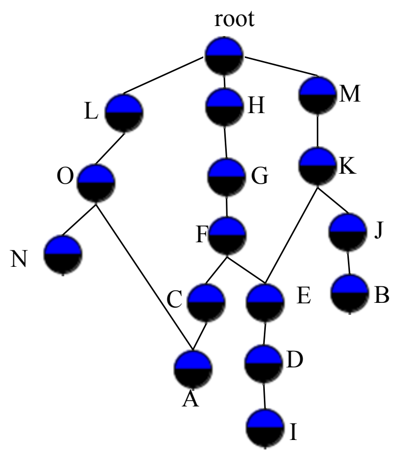


Fig. 2-1 A sample taxonomy lattice

Leacock and Chodorow [34] introduced the maximum depth of classes hierarchy.

$$sim_{LC}(C_1, C_2) = \log \frac{length}{2D} \qquad \text{(Equation 1)}$$

where length is the shortest path length between two synsets and D is the maximum depth of the taxonomy in which a lowest common ancestor (LCA) is found.

Wu and Palmer [35] proposed the following similarity measure that relies on the use of subclass links (edges) between classes. In the above-mentioned examples, the similarity between K and L is less than the similarity between E and B as the latter two classes are in a lower level in the hierarchy structure. They are scaling the proposed method to the relative position of the word in the taxonomy.

$$sim_{WuandPalmer}(C_1, C_2) = \frac{2N_3}{N_1 + N_2 + N_3} \qquad \text{(Equation 2)}$$

where N_1 and N_2 are the number of subclass edges from C_1 and C_2 to their closest common superclass; N_3 is the number of subclass edges from the closest common superclass of C_1 and C_2 to the root class in the taxonomy. For example, the similarity between classes A and E in Fig. 2-1 is calculated as follows. As their closest common superclass are F, N_1 and N_2 are 2 and 1 respectively, and N_3 is 3. Note that the similarity measure of Wu and Palmer is not defined for the case in which the closest common superclass happens to be the root class. For example, the calculation of the similarity for classes A and B in Fig. 2-1 returns 0.

The advantage of edge-based measures is their simplicity. These measures also involve a low computational cost as no corpus is required during the similarity evaluation.

However, edge-based approaches highly depend on the degree of completeness, homogeneity and coverage of the semantic links represented in the taxonomy [36]. Moreover, this approach exclusively uses the shortest path between two classes to their common ancestor. For example, when they are applied in ontologies with multiple inheritances, only the shortest path is taken into consideration. Consequently, large amounts of knowledge available in the taxonomy maybe ignored.

Another problem is that many edge-counting approaches take only "is-a" into account although other relationship types may represent a substantial fraction of the total number of edges. In other words, these approaches rely on the notion that all links in the taxonomy represent a uniform distance [37].

2.1.2. Information-based measures

Similarity measures based on information content rely on functions that determine the degree of specificity of a class. This approach was originally introduced by Resnik [38] who stated that the concept of similarity depends on the amount of information shared between two classes. Resnik [38] emphasized that the more specific a class that subsumes the class being compared (lowest common subsume), the more similar they are.

$$sim_{Resnik}(C_1, C_2) = \max_{S(C_1,C_2)}\left(IC_{corpus}(C)\right) \qquad \text{(Equation 3)}$$

where $S(c_1, c_2)$ are the set of concepts that subsume c_1 and c_2 and IC_{corpus} is the corpus-based information content for a concept C.

This approach has successively been refined by Lin [32]. Lin states that the similarity between two concepts is measured by the ratio between the amount of information needed to state the commonality between the two concepts being compared and the information needed to fully describe what the two concepts are. Lin’s similarity measure is defined as

$$sim_{Lin}(C_1, C_2) = \frac{2IC(C_3)}{IC(C_1) + (C_2)} \qquad \text{(Equation 4)}$$

where IC is the information content and C_3 is the closest common superclass (lowest common ancestor in edge-based measure).

In this thesis for the calculation for Lin's similarity measure, the approach proposed by Seco et al. [41] for the estimation of the IC of a concept is used . Therefore, equation (4) becomes

$$sim_{Lin}(C_1, C_2) = \frac{2g(C_3)}{g(C_1) + g(C_2)} \qquad \text{(Equation 5)}$$

where $g(C)$ is a function that depends on the structure of the ontology and is defined as

$$g(C) = 1 - \frac{\log(h(C) + 1)}{\log(C_{\max})} \qquad \text{(Equation 6)}$$

where $h(C)$ is the number of subclasses of C and $C_{\max}$ is the total number of classes in the taxonomy. For example, the calculation for the similarity between classes 1 and 5 in Fig 2-2 is calculated as follows. Both classes 1 and 5 do not have any subclasses, $h(C)$ is 0 and subsequently the $g(C_1)$ and $g(C_2)$ is 1. Class 3 is their closest common superclass in which the $g(C_3)$ is 0.102 with $h(C)$ is 4. Thus, the similarity measure for classes 1 and 5 is 0.102. Note that, the similarity measure of Lin is influenced by the number of subclass ($h(C)$) of a class. Let takes another example such as the similarity between classes 2 and 5. The $h(C)$ values are 0.613 and 1 for class 2 and class 5 respectively. Therefore, the similarity measure between classes 2 and 5 is 0.127.

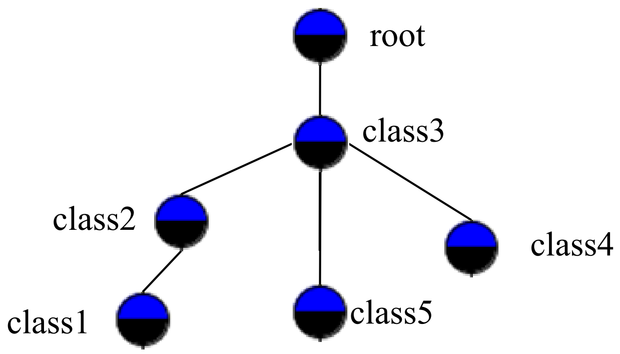


Fig. 2-2 A sample taxonomy for information-based similarity

Jiang and Conrath [26] proposed a distance measure that is computed by subtracting the sum of IC of each term from the IC of their LCA.

$$dist_{JC}(C_1, C_2) = IC(C_1) + IC(C_2) - 2 \times IC(C_3) \quad \text{(Equation 7)}$$

The information-based approaches allow us to compute the similarity using the corpus [39]. Using the available corpus data, these measures outperform the shortest-path measures [26].

Some authors proposed a similarity measure that relies on the whole hierarchical structure and applied it to a WordNet. In this measure, the assumption is that the WordNet is organized in a meaningful way based on the principle of cognitive saliency [40]. They argue that the more hyponyms a concept has the less information it provides, otherwise there would be no need to further differentiate. Likewise, concept at the leaf nodes, are the most specified and provides maximal information. Therefore, the function of this similarity is determined by the number of hyponyms and/or their relative depth in the taxonomy. For example, Seco et al. [41] proposed an IC calculation based on the number of hyponyms.

$$sim_{Seco}(C) = 1 - \frac{\log(hypo(C) + 1)}{\log(\max nodes)} \quad \text{(Equation 8)}$$

where hypo(C) is the number of hyponyms of a class C and *max_nodes* is a constant that is set to the maximum number of concepts that exist in the taxonomy.

The disadvantage of this approach is whenever changes in the taxonomy or in the corpus, re-calculation of the affected branches are required [42]. Moreover, the structure of the taxonomy has a great influence on the similarity scores. Therefore, this approach requires the taxonomy must be as complete as possible. In other words, the taxonomy should include most of the specializations of a specific class in order to provide reliable results. As a result, partial taxonomies with a limited scope may not be suitable for this purpose [37].

2.1.3. Feature-based measures

The feature-based measures are introduced to overcome the limitation of uniform distance assumption in edge-based measures and corpus dependent approaches in information-based measures. In fact, the taxonomical links in an ontology do not necessary represent uniform distance. Feature-based similarities have their origin in the work of Tversky [43] whose similarity measure is based on set theory. Feature-based approach takes into account the features that are common to two classes being compared and also the specific differentiating features of each class. Tversky's similarity measure is defined as

$$sim_{Tversky}(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + \alpha|C_1 \setminus C_2| + \beta|C_2 \setminus C_1|} \qquad \text{(Equation 9)}$$

where C_1 and $C_2$are sets of features, $|C_1 \setminus C_2|$ is set of features in C_1 but not in C_2 and $|C_2 \setminus C_1|$ is set of features in C_2 not in C_1. The α and β are parameters that account for the relative importance of the non-common features. Rodriguez and Egenhofer [45] defined α and β as function:

$$\alpha = \begin{cases} \dfrac{depth(C_1)}{depth(C_1)+depth(C_2)}; if\_depth(C_1) \leq (C_2) \\ 1-\dfrac{depth(C_1)}{depth(C_1)+depth(C_2)}; if\_depth(C_1) > (C_2) \end{cases} \quad \text{(Equation 10)}$$

$$\beta = 1-\alpha \quad \text{(Equation 11)}$$

This similarity returns a score within the range of [0, 1]. The score increases if two classes have more common attributes and decreases with the high number of asymmetrical attributes between the two concepts.

Some recent feature-based approaches rely on information that is available in ontologies. Petrakis et al. [44], proposed the X-similarity, that relies on the matching between synsets and a concept's glosses extracted from WordNet. The two terms are said to be similar if their synsets and glosses of their concepts and those of the concepts in their neighborhood (terms that a connected with semantic relation) are lexically similar. Their proposed similarity function is represented as

$$sim_{x-sim} = \begin{cases} 1; if\_S_{synsets}(a,b) > 0 \\ \max\{S_{neighborhods}(a,b), S_{glosses}(a,b)\}; if\_S_{synset}(a,b) = 0 \end{cases} \quad \text{(Equation 12)}$$

where the similarity for glosses and synsets as well as similarity for semantic neighbors, $S_{neighborhoods}$ are calculated as

$$s(a,b) = \frac{|A \cap B|}{|A \cup B|} \quad \text{(Equation 13)}$$

$$sim_{neighborhods}(a,b) = \max \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \quad \text{(Equation 14)}$$

where A and B denote the set of synsets or glosess for term a and b. The similarity between term neighborhoods is computed differently based on their semantic relationship (is-a and part-of in WordNet) and the maximum (the union of the synsets of all terms up to the root each term hierarchy) is taken.

Also, Rodriguez and Egenhofer [45] proposed a similarity measure by computing the weighted sum of similarities between synsets, features and neighbor concepts.

$$sim_{RE}(C_1, C_2) = w_w \cdot S_{synset}(C_1, C_2) + w_u \cdot S_{features}(C_1, C_2) + w_v \cdot S_{neighborhods}(C_1, C_2)$$

(Equation 15)

where w_w, w_u and w_v are the weight of each component and the summation of weight is equal to 1.

Rodriguez and Egenhofer's similarity measure is only applicable to the noun and a verb category in WordNet whereas a term can be represented by others features such as attributes associated to the terminology.

2.2. Semantic similarity measures using multiple ontologies

The semantic similarity methods presented so far assume that the classes being compared are from the same ontology. However, the numbers of ontologies are increasing due to the advent of semantic web in which the developed ontology is used to formalize the conceptualization behind the idea of semantic web [46]. Although the topic is out of scope of this research, in this section, we provide a brief discussion on how the similarity methods can be used to compare classes from different ontologies which is referred to as cross-ontology similarity methods in the literature.

According to Cross and Hu [19], a cross-ontology similarity method is an approach that is based on establishing association links between the classes have been proposed. The foundation for many existing approaches is the use of Tversky's model of similarity with various features of classes [19].

The more properties of two classes share in common, the more links there are between the classes and the more closely related they are [47]. Recently, the cross-ontology similarity methods have been proposed in very promising research area of the matchers of ontology alignment system to support the semantic interoperability. Ontology alignment (OA) systems focus on finding a set of mapping pairs between source ontology, O_S and target ontology, O_T with each pair having a similarity degree in the range of 0 and 1 [48]. Many proposed methods use background knowledge sources, WordNet, UMLS or both as a reference ontology with semantic similarity measure. There are several systems have been introduced to facilitate the ontology alignment process such as OLA, ASMOV, CIDER, Anchor-Flood [46] and AgreementMaker [48].

Cross, Silwal and Morell [48] show a very recent experiment using reference ontologies (it is also known as mediating ontologies) to improve the ontology alignment process. They incorporated semantic similarity in reference ontologies to determine indirect mappings where source and target classes map to different concepts in mediating ontology. Their work extends the AgreementMaker's mediating matcher (MM) by incorporating the semantic similarity measures within the reference ontolgy and it is called mediating matcher semantic similarity measurement (MMSS). For this purpose, the Adult Mouse Ontology (MA) and Human Anatomy (HA) were used for the evaluation of the proposed approach.

The first step is to determine the mapping set between source and target classes on the same class in the reference ontology. In this step, the base similarity matcher with lexicon (BSM^{lex}) is used to compose mapping from the source and target classes to produce an exact match on the bridge classes in the mediating ontology, M_{ST}.

Also, they consider the sets of unmapped source classes, U_S in the mapping set from source to mediating ontology and the sets of unmapped target classes, U_T in the mapping set from target to mediating ontology. For each pair (s, t) in U_S x U_T, the semantic similarity measure is used to compute the similarity between all bridge classes for s and all bridge classes for t. The standard Lin semantic similarity measure is used with IC is defined by [41] was used in their experiment. They use maximum aggregation operator to determine the enhanced mapping set, E_{ST}. The final mapping is $M_{ST} \cup E_{ST}$. The results of the experiment show that the MMSS discovers more correct mapping than the MM.

2.3. FCA and class hierarchy generation

To emulate the human ability in assessing similarity between things, computational models require a support from knowledge sources. Knowledge sources represent the concepts of the real world domain that are defined formally with relationships they share with the other concepts of the same domain. Some of the knowledge sources are taxonomy (class hierarchy), ontology, thesaurus and domain corpora.

The proposed approach requires a taxonomy of classes of products or processes. Typically, however class hierarchies are developed in an ad-hoc fashion, lacking the rational of their structure. To resolve this issue, this thesis proposes a class hierarchy development based on Formal Concept Analysis (FCA).

FCA is an analysis technique for knowledge processing based on applied lattice and order theory [27].

Several efforts have been reported on the use of FCA in products and processes. For example, Fu and Cohn [28] suggest the use of FCA to support the development of municipal utility domain to overcome the limitation of current mapping information. In another related effort, Nanda et al. [18] proposed the use of FCA for providing a systematic guideline for constructing product families domain. Stumme [29] described the use of FCA to manage the knowledge related to business processes across department and company boundaries.

On the other hand, several works have been proposed to measure the similarity of classes obtained with FCA such as in Formica [30], Tadrat et al. [31], Alqadah and Bhatnagar [32], Zhao and Halang [33], Saquer and Deogun [34] and Souza et al. [35]. Formica [30] proposes a similarity based on the information-based approach to calculate the classes with a weight [0,1] which is user-defined. Alqadah and Bhatnagar [32] improve the Jaccard coefficient, Sorenesen coefficient (or Dice coefficient) and Symmetric difference based on set theory where the zero-induced is incorporated. In addition, Zhao and Halang [33] develop a similarity measure for FCA by modifying the Tversky's feature-based similarities. They replaced the sets of features with a rough lower approximation which is represented only with the sets of objects of the two concepts. Tadrat et al. [31] propose a similarity measure that characterizes by a vector of frequencies of the object and attributes between two concepts in FCA. Their approach was based on vector model of information retrieval.

FAST is used to define the formal attributes that can later be used in the FCA. This research uses FCA to generate lattice in which, FCA requires information to be organized in a formal context. For this purpose, the list of potential classes (formal object) and formal attributes are added to the context table. Context table represents the object and attribute information and their relation in FCA that are organized in incidence matrix. If a formal object has a formal attribute, a checkmark is inserted in the corresponding cell. Subsequently, a lattice is generated. The next step is an iterative process for analyzing the resulting lattice and resolving inconsistencies. All concept-subconcept relation in lattice is analyzed. If an inconsistency is found, the context table is revised by adding or removing attributes. A new lattice is generated if the context table is modified. The resulting lattice and formal attribute-information are used to create a class hierarchy and convert it into a computer-processable form. A taxonomy structure and formal attribute information came as the results of taxonomy generation step. Appendix A provides the description of FCA in which the proposed approach is based on.

Chapter 3

THE PROPOSED APPROACH

3.1. Introduction

Theoretical frameworks for products and processes refer to the world view with which products and processes can be represented. Such a theoretical framework is useful for determining formal attributes. For this purpose several existing theoretic frameworks for products and processes were studied.

Chandrasekaran's extensive work on Functional representation (FR) [50] defined FR is a device-centered description of the product that is organized in structure (what it is), function (what the device is intended to do) and behavior (how the artifact does what it does). FR is a top-down approach in which the function of the device is specified first and the behavior of device components is specified in terms of how they contribute to the individual functions.

In order to achieve the function of interest, a function is represented by describing its application, the initiating conditions and the predicates that the product has to satisfy. How a product achieves its functions is described by using Casual process description (CPD) or by using passive function characterizes the structural properties of a device. A CPD is represented as a directed graph in which the nodes represent the states (process variables and device states) while the arcs represent state transitions.

On the other hand, behavior-structure representation distinguishes structural and behavioral aspects of the artifact based on general systems theory [51], which identifies structure and behavior descriptions of complex systems. In VEDA [52], [53] information models describe the artifact in terms of material (e.g. pipes, tanks) and phenomenological entities. The behaviors of individual structure subsystems together with their structural interrelations generate the behavior of the whole system. The structure systems are classified into devices and connections. In this representation, behavior refers not to the behavior of the device but to physicochemical phenomena that takes place in a device.

Several efforts have been made to find a reusable representation of processes. Sowa [62] describes a *process* according to time points that mark the beginning and ending of the process and the changes that take place in between. To Sowa, a process can be caused by one or more *agents* over some time interval. Here, an *agent* is an animate entity that is capable of doing something to fulfill a specific intention.

A *process* is defined in the SUMO Ontology [63] as "the class of things that happen and have temporal parts or stages." A *process* may have participants which are *objects*, such as the machine, circuit boards, components, and solder in a soldering process. In SUMO, an *object* can denote a physical object or a geographical region. *Agent*, *instrument*, *resource*, and *result* are objects that participate in the *process*. An *agent* is defined as an active determinant (either animate or inanimate) of the *process*, with or without voluntary intention. A *resource* is something that is present at the beginning of a *process*, is used by the *process*, and as a consequence is changed by the *process*. An *instrument* is used by an *agent* to perform a *process* and is not affected by that *process*. A *resource* differs from an *instrument* in that its internal or physical properties are altered in some way by the *process*.

A process in IDEFØ [64] is described in terms of *activity* building blocks. Fig. 3-1 shows an *activity* is characterized by its *inputs*, *outputs*, *constraints*, and *mechanisms*. *Input* is the information, material or energy that is converted to the *output* of an *activity*. An *output* is the information, material or energy produced by or resulting from the *activity*. A *constraint* or *control* is the information, material or energy that constrains and regulates an *activity*. A *mechanism* represents the resources, such as people, equipment, or software tools that perform an *activity*. Furthermore, an *activity* can be composed of other activities (mereology).

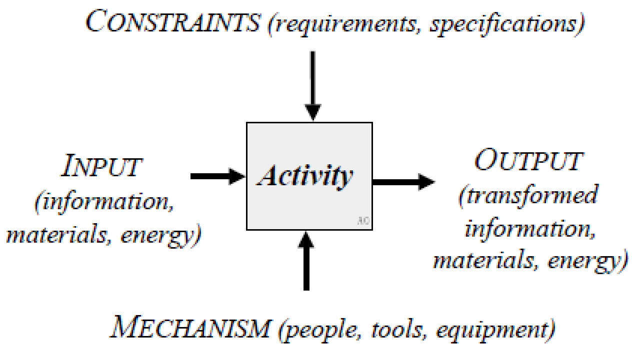


Fig. 3-1 Activity representation in IDEF0

ISO 15926 defines *activity* as a *possible individual* that has its life cycle bounded by *beginning* and *ending* events [65] as shown in Fig. 3-2. In addition, an *activity* brings about change by causing an *event* (an *event* occurs at an instant in time). A *participation* relation is used to express that a possible individual is involved in an *activity*. Because ISO 15926 uses a four-dimensional view of the world, an *activity* consists of temporal parts of those members of possible individuals that participate in the *activity*. For example, in creating a blind hole on a metal piece using a hand drill, the drilling activity shares the temporal parts of the worker and the hand drill that participates to change the shape of the piece. In this example, the drilling activity causes the hole to come into existence.

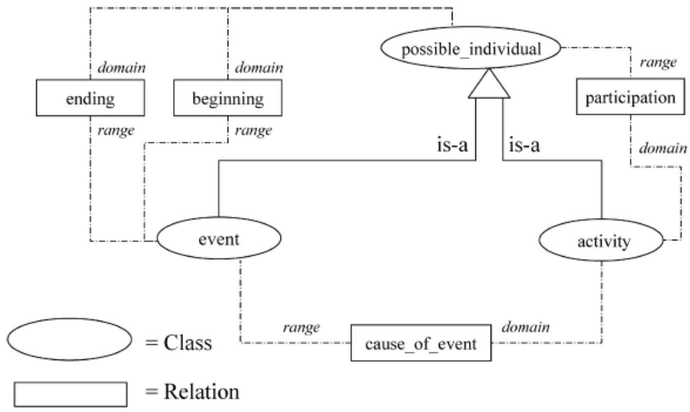


Fig. 3-2 Activity in Upper ontology based on ISO 15926

WPML is an ontology-based language designed to represent work processes [66], [67]. WPML is based on OntoCAPE [68], which was originally developed as a comprehensive ontology for the chemical process engineering domain. WPML defines an *action* as a building block that describes a step in a work process. *Actions* are characterized by their causal and temporal aspects. On the other hand, the changing nature of the *action* is described by means of the so-called *OperationalFunction*. Therefore, valve_opening, drilling, material_charging can all be defined as subclasses of *OperationalFunction*.

Gero and Kannengieser [69] propose the use of the structure-behavior-function (SBF) world-view to characterize a process. The notion of function of a process is related to the goal of providing a given process, which assumes that processes can be *designed*. Behavior attributes refer to those attributes of a process that allow comparison on a performance level. Examples of behavior attributes of processes are speed, rate of convergence, cost, amount of space required, and accuracy. The structure of a process is described in terms of its inputs, outputs, and subprocesses.

One common denominator in all these approaches is the existence of an elementary element to define the process that is used together with relations that associate the process with other objects. The most common relations are those for identifying the objects that are transformed by the process (the input), those for representing the objects that are produced by the process (the output), those for identifying the tools or the actors that participate in the process, the relations for indicating the location of the process, part-whole relations for describing the process structure, and time duration. Table 3-1 summarizes these common elements.

Table 3-1 Comparison of process representations

	Process building block	object that is changed by the activity	object that is produced by the activity	performer of activity	location of the activity	composition of the activity	time duration
IDEFØ	activity	input	output	control	--	yes	--
Sowa	activity	--	--	agent	situation	--	starting and stop-ping
SUMO	process	resource	output	agent, instrument	region	--	--
ISO 15926	activity	possible_indi vidual (by means of the participation relation)	possible_individua l (by means of the participation rela-tion)	possible_individ ual (by means of the participation relation)	possible_indivi dual (by means of the relative_locatio n relation)	yes	points in time (by means of the beginning and ending relations)
Onto-CAPE	action	hasInputState , actsOn	hasOutputState	actor, tool	actsOn	yes	TimeInterval
SBF	transformation	input	output	--	--	yes	--

3.2. Product representation

In this thesis, the theoretical framework for representing a product is based on the ISO 15926 standard which specifies an upper ontology for long-term data integration, access and exchange [60]. It was developed in ISO TC184/SC4-Industrial Data by the EPISTLE consortium (1993-2003) and designed to support the evolution of data through time. The upper ontology was developed as a conceptual data model for the representation of technical information of process plants including oil and gas production facilities but it was designed to be generic enough for any engineering domain [61]. The theoretical framework is illustrated in Fig. 3-3.

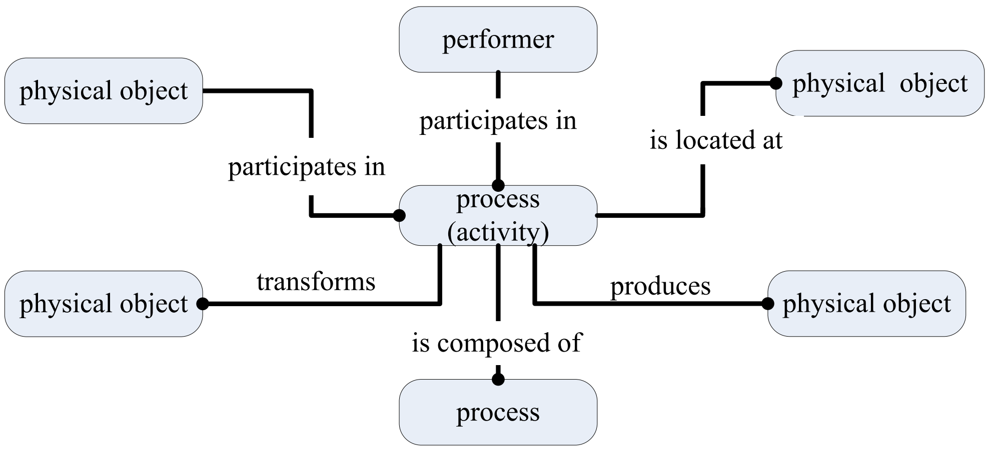


Fig. 3-3 Composition of device and its relation to processes.

In this theoretical framework, the physical object is represented in terms of its physical parts as well as in terms of its relation to some process (activity).

The physical part of a product is represented by physical object that is defined in terms of a distribution of matter, energy, or both. A physical object can be described in terms of its parts (Fig. 3-4). This is possible through a mereological relation that refers to the relationship that a part has in regards to the whole of an object. Mereological relations are reflexive, antisymmetric, and transitive.

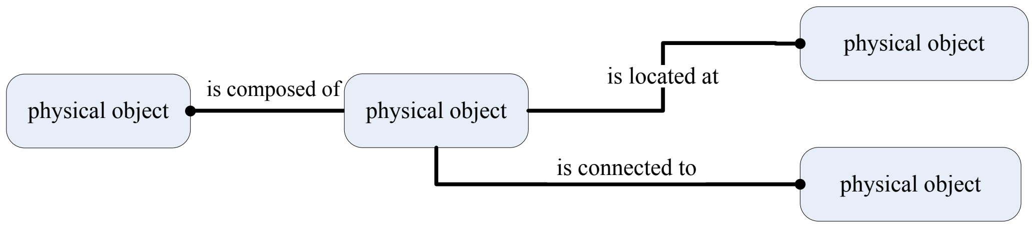


Fig. 3-4 Composition of device presentation

Physical objects exist in reference to a specific place. The location relation (relative location in ISO 15926) is a kind of mereological relation that is used to locate objects in a particular place.

The function of a product can be defined as an intended process associated to the device. For example, the function associated to a sofa is represented as the process of seating in which the sofa is involved along with a person that sits on it.

Similarly, the function of an electric fan is to generate cool air. In this case, the description of the device includes information about the home appliance and the cooling process. The cooling process is in turn composed of other processes such as conversion of electricity into rotary movement, convection, diffusion and heat transfer. Therefore information about the process or processes associated to the device is an indispensable element to complete the description of the product.

Different objects can participate in a process. Participating physical objects include those objects that are transformed by the process, those objects that are produced by the process, those objects that are not affected by the process (the device itself, other tools or instruments), as well as agents (such as a person or a control system) that participate or execute the process.

As in with a physical object, a process is also described in terms of its relative location and its mereology.

3.3. Process representation

The theoretical framework for processes is the same as that for products (Fig. 3-3). In general, a process changes an object that exists before the execution of the process to produce another object. In a four-dimensional view, these objects correspond to the temporal parts of the object before and after the process. In addition, among the objects that participate in a process we can distinguish those entities that are not intended to be affected by the activity but that are used by the activity. Therefore, four types of objects that participate in a process can be identified: the objects that are transformed by the process (the inputs), the objects that are produced by the process (the outputs), the objects that are used for the execution of the process (the performers) and the objects that accommodate the process (the location of the process).

For example, a drilling process always transforms a solid object (the so-called blank or work piece) and produces a solid object that has at least one hole. A performer in this case is a cutting tool that is pressed against the solid object and rotated in a given way so as to produce the hole. In this example, the location of the process is the machine that holds the cutting tool that is also perpendicular to the work piece. One can argue that both the performer and the location may be affected by the process (e.g. deteriorated) but they are not intended to be modified, which makes them different from the other two types of objects.

The Performer corresponds to the concept of instrument in SUMO. It indicates an object that is used by the process but that is not intended to be changed by the process.

In addition, a process can be composed of other subprocesses. For example, a given hole-making process can include a cooling sub-process in order to reduce the wear of the cutting tool as a result of friction force.

3.4. Formal Attribute Specification Template (FAST)

The Formal Attribute Specification Template has been developed for identifying the formal attributes of a given class. FAST is a systematic guideline to characterize the classes of products and processes and represents the relationship between the products and processes.

In FAST, a product has the following kinds of formal attributes:

- the classes of objects that compose the product (the product parts)
- the classes of places where the product is required to be
- the classes of process in which the product participates

Fig. 3-5 shows the steps for the selection of formal attributes of a given class of product.

Similarly, FAST identifies five kinds of formal attributes required for describing a process:

- the classes of objects that are always transformed by the process (the input of the process)
- the classes of objects that are always produced by the process (the output of the process)
- the classes of performers that are always used by the process
- the classes of locations that always accommodate the process
- the classes of process composition (the parts of the process)

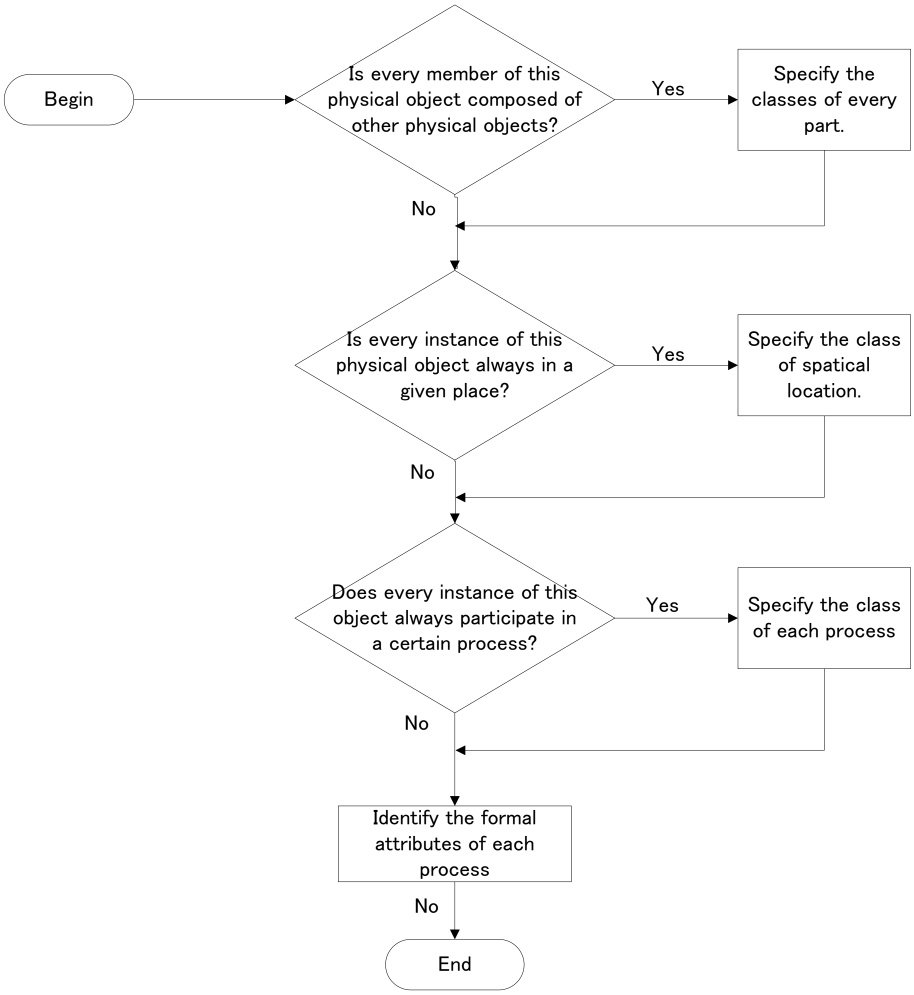


Fig. 3-5 Flow diagram for the formal attribute selection of a given class of product

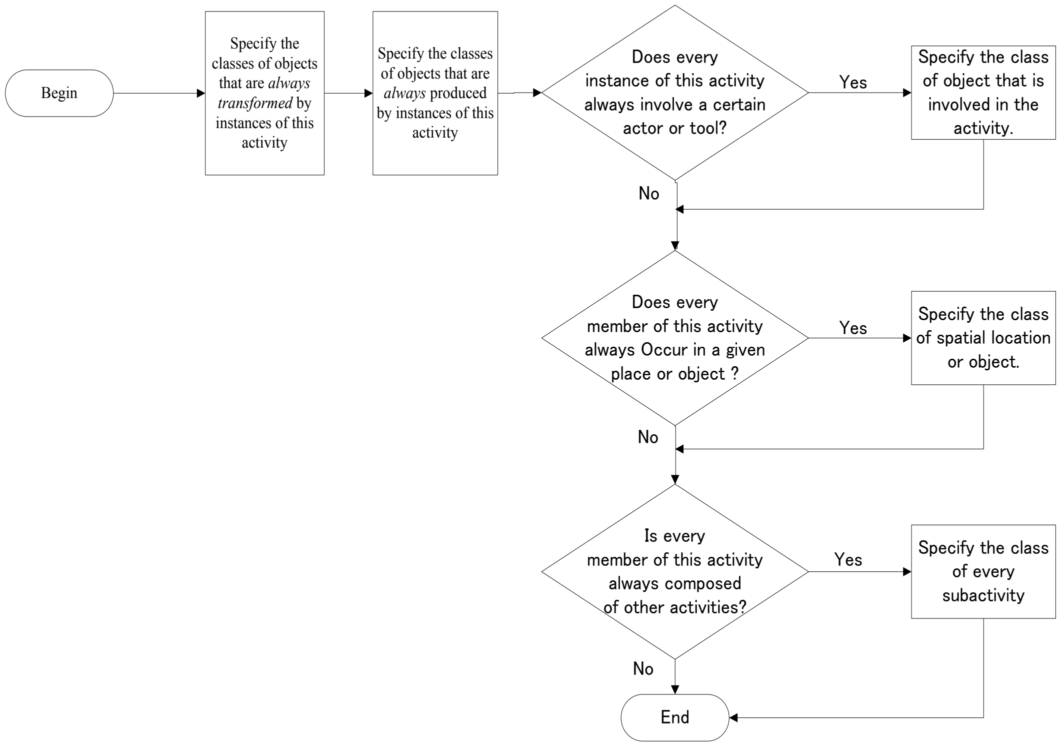


Fig. 3-6 Flow diagram for the attribute identification of a given candidate class

Based on these five characteristics the formal attributes of a given class of product or process can be identified . Fig. 3-6 shows the steps for the selection of formal attributes of a given class of process. For example, to characterize a fusion welding process, the objects that are transformed by the activity are solid physical objects. The object produced by any member of this class of activity is a physical object that is made of the welded parts. As heating is always involved in a fusion welding, it is a part of the activity. Therefore, the attributes of the welding process become: “transforms solid physical objects,” “produces a physical object,” and “composed of heating.”

On the other hand, if we are given a class of product such as printer that is involved in printing. The objects that are transformed by the particular printing process of a printer are: data, paper and electricity. The object that is produced is printed paper. Injecting is always involved as a part in the printing process in which the printer is involved in using inkjet as performer. Thus, the attributes of the printer become: consumes data, consumes paper, converts electricity, involved in using inkjet and produces printed paper.

Each formal attribute in the FCA context table is seen as a constraint about the meaning of a particular class of product or process and it is not an attribute in the sense of a property of a specific instance.

3.5. Procedure for taxonomy construction with FCA

This thesis follows the general steps proposed by Stevens et al. [70] that include, identification of purpose and scope, knowledge acquisition, conceptualization, integration, encoding, documentation, and evaluation but we use FAST to guide the knowledge acquisition and conceptualization stages.

The proposed methodology aims at facilitating the developing of taxonomy in such a way that the developer can justify the rationale behind the involved decisions. The procedure for taxonomy construction consists of the following steps:

Step 1. Identification of the purpose and scope of the project.

The purpose and scope are necessary to identify the domain of interest that the taxonomy will cover. For example, developing a taxonomy for electric home appliances.

Step 2. Identification of the potential classes to be defined under the scope of the project.

This step refers to the identification of candidate classes that may or may not appear in the final taxonomy and the object column of a FCA context table is populated with these classes.

Step 3(a). Compile and organize definitions of each class.

Information sources such as scientific papers, technical reports, and

Internet resources are consulted to define each class in natural language. When several definitions are found preference is given to those that explicitly describe participating objects, objects transformed by the process (inputs), objects produced by the process (outputs) and/or subactivities. When contradictions among several definitions of a given class occur experts can be consulted to disambiguate.

Step 3(b). Identification of formal attributes.

Formal attributes are identified using the FAST.

Step 4. Add the attributes and incidence information to the context table.

The formal attributes are added to the context table created in Step 2. If a class has always an attribute, a checkmark is inserted in the corresponding cell.

Step 5. Use the FCA to generate a concept lattice.

After adding the formal attributes, the context table is completed and a lattice is generated. Lattices in this paper were generated by means of the Grail algorithm [71] (a simpler algorithm is illustrated in Appendix B). Finally, the lattice is used to create the ontology. The naming of each class is done based on object or attributes labels from the nodes in the lattice.

Step 6. Analyze the lattice and resolve inconsistencies.

The first thing to be done is to check the concept-subconcept relation. Analysis of the lattice is done using object exploration [72]. The ontology designer analyzes the consistency of formal objects by tracing all paths in the lattice. The tracing starts from the root node, then to the next lower node and continuing until reaching the bottom node. If the relation between objects in a concept and objects in its subconcept is found to be inconsistent, then inconsistency is resolved by adding or removing attributes. In case of new attributes, the context table is revised. If the context table is modified then a new concept lattice is generated. This procedure is repeated until all concept-subconcept relations have been explored.

Step 7. Create a class hierarchy and convert it into a computer-processable form.

In this step, the resulting lattice and formal-attribute information obtained in the previous step are used to create a class hierarchy of an ontology. The naming of each class is done after the names of object and attributes that correspond to the concept on which the class is derived. An ontology editor such as the Protégé ontology editor [27] can be used for carrying out this and the remaining steps.

Step 8. Connect the class hierarchy into an upper ontology

Integration is carried out by means of aligning the resulting ontology with an upper ontology that defines domain-independent classes such as physical objects, activities, mereological and topological relations.

The results of all these steps are a taxonomy structure and formal attributes information. These results can be used in equation 17-25 to evaluate the proposed approach.

3.6. The semantic similarity measures of the proposed approach

The semantic similarity measures of the proposed approach compute the similarity of classes of products or processes in a taxonomy by taking into account the formal attributes from FCA. In a given class hierarchy, the formal attributes play a crucial role to distinguish one class from another. The similarity between two classes is a function of the number of formal attributes they share in common. The more common formal attributes shared by the two classes the more similar they are. This means that attribute information can be used to justify the design of class hierarchies (i.e. taxonomies). Subsequently, similarity measures can be developed based on the number of common attributes that are shared between two classes. Also, the semantic similarity measures of the proposed approach follow a similar principle as proposed in the Tversky's model (Equation 9) which considers that the similarity between two classes in a taxonomy can be measured as a function of their common and differential features. For this purpose, several similarity equations in data mining literature are investigated in this research by using formal attributes as the sets they compare.

- **Formal attributes**

The proposed approach emphasizes on the common attributes shared by two classes in a taxonomy. The approach assumes that two classes that share formal attributes are considered more similar than classes not having common attributes. That is, for a given classes, this research considers the degree of overlap (common attributes shared by two classes) as a function for similarity.

The attributes in the semantic similarity of the proposed approach refer to the formal attributes which obtained using a systematic method by using FAST.

- **Taxonomical relationships**

In a taxonomy that based on FCA, a class $A\langle O_i, A_i\rangle$ is said to be subclass of another class $B\langle O_j, A_j\rangle$ provided that $\left(A_i \subseteq A_j\right)$. In other words, in a given hierarchical structure, a class is equivalent to another class if both classes have exactly the same attributes[1]. The class B is the superclass of A which defined an order written $A \leq B$. The relation $\leq$ is known as hierarchical order of the classes.

The semantic similarity measure of the proposed approach also considers the uses of the multiple-inheritance (a class is subsumed by several superclasses). Differently from previous edge-based measures that considers only the shortest path-length between two classes, the proposed measure allows measuring the similarity between classes by considering the multiple taxonomic superclasses belonging to all possible taxonomical paths connecting the classes being compared. For measuring the similarity of multiple-inheritance, classes are connected through the subsumption (is-a) relation. As the subclass-superclass relation is transitive, a subclass inherits all the attributes from all its superclasses. Therefore, the semantic similarity measures of the proposed approach emphasized on the sets of formal attributes associated to the classes includes all those inheritance attributes from its superclasses that found traversely going through all the upper taxonomical paths modeled in the ontology for that concept.

- **Similarity measure based on formal attributes**

The similarity measure used in the proposed approach is represented by equation (16)

$$sim(C_i, C_j) = a\frac{|A_i \cap A_j|}{f(A_i, A_j)} + (1-a)\frac{|A'_i \cap A'_j|}{g(A'_i, A'_j)} \qquad \text{(Equation 16)}$$

where C_i *and* C_j are classes in the taxonomy, $|A_i \cap A_j|$ is the number of common attributes shared by classes C_i *and* C_j, A_i *and* A_j are the sets of attributes of classes C_i *and* C_j respectively and a takes the values of 0 or 1.

[1] The attributes of a class also include those attributes inherited from its parent classes.

$sim(C_i, C_j)$is a function whose values are in between the range of 0 and 1. Value of 1 denotes the two objects are highly similar, while the two objects are said to be dissimilar if similarity value is equal to 0.

$f(A_i, A_j)$ is a function of the attributes of classes C_i *and* C_j. For example, when *a=1* and $f(A_i, A_j) = |A_i \cap A_j| + \alpha|A_i \setminus A_j| + \beta|A_j \setminus A_i|$ equation (16) becomes Tversky's similarity

$$sim_{Tversky}(C_i, C_j) = \frac{|A_i \cap A_j|}{|A_i \cap A_j| + \alpha|A_i \setminus A_j| + \beta|A_j \setminus A_i|} \qquad \text{(Equation 17)}$$

where $|A_i \setminus A_j|$ is the relative complement of A_i *and* A_j. Following the work of Rodriguez and Egenhoffer [55], parameters α and β are calculated as Equation 10 and 11.

When $\alpha = 0.5$ and $\beta = 0.5$, Equation (17) becomes the Dice's coefficient [56] which quantifies the overlap of two sets of attributes in relation to an estimate of their average size. In other words, the Dice coefficient is the number of attributes in common to both classes C_i *and* C_j relative to the average size of the total number of attributes present in C_i *and* C_j.

$$sim_{Dice}(C_i, C_j) = \frac{|A_i \cap A_j|}{1/2(|A_i| + |A_j|)} \qquad \text{(Equation 18)}$$

Suppose we are given two classes of scanner and fax modem as shown in Fig. 3-3:

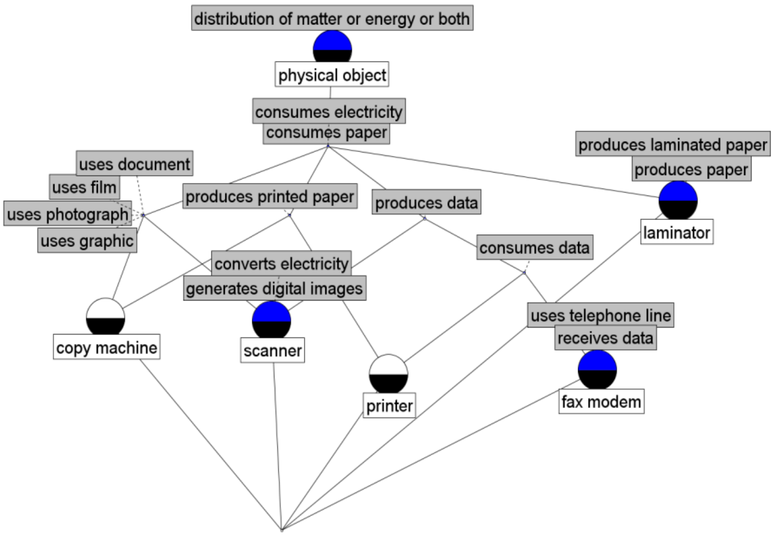


Fig. 3-7 A sample for similarity calculation

Scanner, A_i= {distribution of matter or energy or both, consumes electricity, consumes paper, uses document, uses film, uses photograph, uses graphic, converts electricity, generates digital images, produces data}

fax modem,A_j = {distribution of matter or energy or both, consumes electricity, consumes paper, produces data, consumes data, uses telephone line, receives data}.

The cardinality of set of attributes of scanner, $|A_i|$ is 10, while the cardinality of set of attributes of fax modem $|A_j|$is 7. The common attributes are= {distribution of matter or energy or both, consumes electricity, consumes paper, produces data}, then $|A_i \cap A_j| = 4$. By equation 16, the similarity between scanner and fax modem is (2(4)) / (10 +7) = 0.471.

When $\alpha = \beta = 1.0$, equation (17) becomes the Jaccard's coefficient [57], in which $f(A_i, A_j)$ is the cardinality of the union sets of sets A_i *and* A_j

$$sim_{Jaccard}(C_1, C_2) = \frac{|A_i \cap A_j|}{|A_i \cap A_j| + |A_i \cap A_j|} = \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \qquad \text{(Equation 19)}$$

Consider again the set of attributes of scanner, $|A_i|$ and set of attributes of fax modem $|A_j|$ used for in the previous example. The similarity between scanner and fax modem using the Jaccard coefficient is 4/13= 0.308.

Jaccard's coefficient is related to the Dice's similarity through Equation 20:

$$sim_{Jaccard} = sim_{Dice}(A_i, A_j)/(2 - sim_{Dice}(A_i, A_j)) \qquad \text{(Equation 20)}$$

When a=1 and $f(A_i, A_j) = \min(|A_i|, |A_j|)$ equation (16) becomes the overlap coefficient [58] given by Equation 21. The overlap between two set of attributes of classes C_i and C_j is equal to the intersection between the two set of attributes normalized by the size of the minimum number of attributes.

$$sim_{Overlap}(C_1, C_2) = \frac{|A_i \cap A_j|}{\min(|A_i|, |A_j|)} \qquad \text{(Equation 21)}$$

Another variation is the all-confidence similarity [58]. It differs from Equation 21 where the two set of common attributes are divided by the maximum number of attributes between classes A and B.

$$sim_{All\ confidence}(C_1, C_2) = \frac{|A_i \cap A_j|}{\max(|A_i|, |A_j|)} \qquad \text{(Equation 22)}$$

When $f(A_i, A_j) = \sqrt{A_i}\,\sqrt{A_j}$ equation 16 becomes a cosine similarity with attributes sets instead of vectors.

$$sim_{Cosine}(C_i, C_j) = \frac{|A_i \cap A_j|}{\sqrt{A_i} \times \sqrt{A_j}} \qquad \text{(Equation 23)}$$

When $a=0$ and $g(A_i', A_j') = min(|A_i'|, |A_j'|)$ equation 16 becomes equations 24 which is similar to that of van der Weken et al. [59] but using formal attribute sets instead of fuzzy sets.

$$sim_{vanDerWeken1}(C_i, C_j) = \frac{|A_i' \cap A_j'|}{\min(|A_i'|, |A_j'|)} \quad \text{(Equation 24)}$$

Another variation is equation 25:

$$sim_{vanDerWeken2}(C_i, C_j) = \frac{|A_i' \cap A_j'|}{\max(|A_i'|, |A_j'|)} \quad \text{(Equation 25)}$$

where A_1' and A_2' are the complements of sets of attributes A_1 and A_2. Values of *a*, $f(A_i, A_j)$ and $g(A_i', A_j')$ are summarized in Table 3-2.

Table 3-2 Association among sets of attributes of classes being compared

Equation	a	$f(A_i, A_j)$	$g(A_i', A_j')$
sim$_{Tversky}$	1	$\lvert A_i \cup A_j\rvert + \alpha\lvert A_i \setminus A_j\rvert + (1-\alpha)\lvert A_j \setminus A_i\rvert$	
sim$_{Dice}$	1	$1/2\ (\lvert A_i\rvert + \lvert A_j\rvert)$	
sim$_{Jaccard}$	1	$(\lvert A_i\rvert + \lvert A_j\rvert) + (A_i \cap A_j)$	
sim$_{Overlap}$	1	$\min(\lvert A_i\rvert, \lvert A_j\rvert)$	
sim$_{All\ confidence}$	1	$\max(\lvert A_i\rvert, \lvert A_j\rvert)$	
sim$_{Cosine}$	1	$\sqrt{A_i} \times \sqrt{A_j}$	
sim$_{van\ der\ Weken\ 1}$	0		$\min(\lvert A_1'\rvert, \lvert A_2'\rvert)$
sim$_{van\ der\ Weken\ 2}$	0		$\max(\lvert A_1'\rvert, \lvert A_2'\rvert)$

For the evaluation of the semantic similarity measures of the proposed approach, we also investigate a composite similarity obtained by combining semantic similarities:

$$sim_{composite}(C_i, C_j) = w_1 sim_1 + w_2 sim_2 \quad \text{(Equation 26)}$$

where w_1 and w_2 are weights and sim_1 and sim_2 represent two different semantic similarity measures of the proposed approach.

Chapter 4 and 5 demonstrate the evaluation of the proposed approach against edge-counting and information-based similarity measures. In order to quantify the efficacy of each similarity measure, the degree of correlation with human judgment and NGD similarity will be used.

Chapter 4

EVALUATION OF THE PROPOSED APPROACH

4.1. A taxonomy for home electric appliances

This chapter discusses the evaluation on the effectiveness of the proposed approach (Chapter 3) and its comparison with respect to edge-counting and information-based measures. We provide an example for evaluating of the proposed approach in the domain of home electric appliances. The characteristics of home electric appliances are described in terms of processes and participating objects as outlined in Sections 3.3 to 3.6.

In order to enable fair comparisons, several researches use human judgment for evaluating the similarity between word pairs [32]. As a result, the degree of correlation obtained against human judgments and the results of the computerized similarity measures (i.e. the semantic similarity measures of the proposed approach, edge-counting and information-based measures) can be used to quantify the likeness of two classes being compared. If the degree of correlation of the proposed approach is close to 1, the proposed approach properly approximates the judgments of human subjects.

4.1.1. Taxonomy construction

This section describes the development of an electric home appliance taxonomy, which is based on the method described in Chapter 3. The list of potential classes was extracted from product categories in Amazon.com and the formal attributes information were obtained using FAST, using expert consultations and brainstorming. In the development of the taxonomy, we focused on the process or processes in which the given appliance participates or is involved. Therefore, formal attributes include a reference to the process or a description of the process in terms of the objects that are transformed by the process and the objects that are produced by the process. For example, the formal attribute identification of an electric kettle starts by the analyzing its main process associated to it, which is a process that produces hot water. Heating is a part of that process. In order to produce hot water, the electric kettle consumes electricity that is converted into thermal energy that is used to heat water. Therefore, the formal attributes of an electric kettle become *heats; produces hot water; heats water;* and *consumes electricity*.

With formal attributes information obtained this way, a context table was created (Fig. 4-1). Subsequently, the Grail algorithm [71] was used to generate the concept lattice shown in Fig. 4-2. After analyzing and correcting the lattice, the final lattice and formal-attribute information were used to develop taxonomy using the Protégé ontology editor [74]. Subsequently, the resulting class hierarchy was saved in OWL format [75].

Strictly speaking, formal attribute information must be in the form of axioms as in the following example.

```
Class filtration:
    SubClassOf:
        heating_device
    SubClassOf:
        produces some hot_water
```

However, for simplicity in the similarity calculation, formal attributes were added as OWL properties. For example, the formal attribute for "produces hot water" is declared as follows:

```
Declaration( ObjectProperty( :produces_hot_water) )
ObjectPropertyDomain(:produces_hot_water :water_heater)
```

This resulted in an OWL file with 33 classes, 39 properties, and 5 levels in the class hierarchy.

	home electric appliances	room electric heater	hair dryer	electric blanket	washing machine	electric clothes dryer	refrigerator	room air-conditioner	electric dish washer	microwave oven	toaster	electric kettle	television set	conventional electric fan	blender	bread machine	electric oven	water heater	vacuum cleaner
consumes electricity	x																		
generates infrared radiation	x										x						x		
generates high energy frequency waves	x									x									
heats	x	x	x	x		x		x		x	x	x				x	x	x	
heats room	x	x						x											
heats water	x											x						x	
heats body	x			x															
heats food	x									x	x					x	x		
removes heat	x						x	x						x					
removes heat from room	x							x						x					
removes heat from food	x						x												
removes water	x		x		x	x			x										
removes water from hair	x		x																
removes water from clothes	x				x	x													
removes water from dishes	x								x										
removes dirt	x				x				x										x
removes dirt from clothes	x				x														
removes dirt from dishes	x								x										
removes dirt from surfaces	x																		x
produces cooked food	x									x							x		
bakes	x															x	x		
produces baked food	x																x		
bakes bread	x															x			
sucks up dirt	x																		x
produces toasted bread	x										x								
receives tv-signals	x												x						
mixes food	x														x				
chops food	x														x				
delays bacteria growth	x						x												
delays mold growth	x						x												
produces air circulation	x							x						x					
produces food	x									x	x				x	x	x		
produces hot water	x											x						x	

Fig. 4-1 Context table for an ontology of home electric appliances

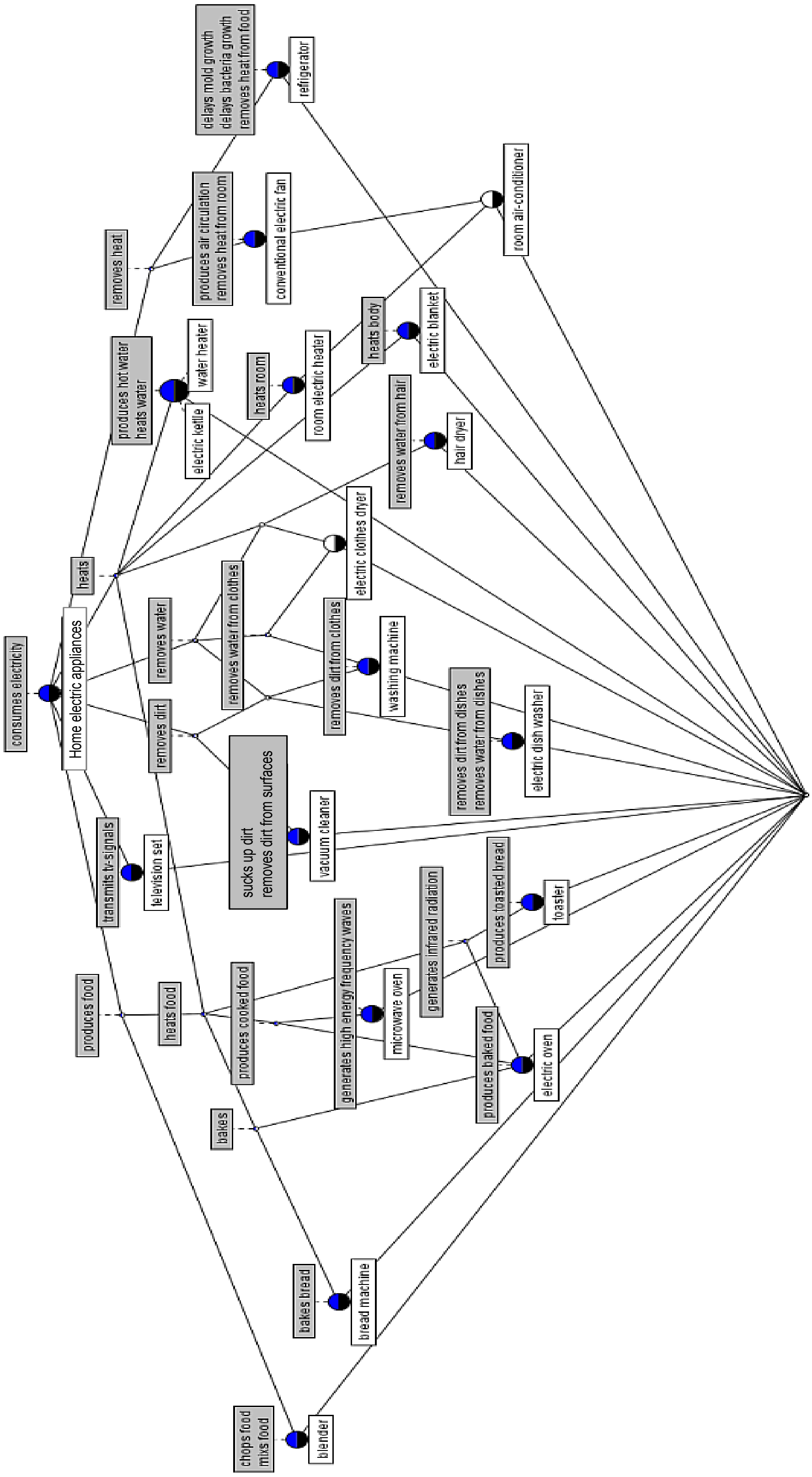


Fig. 4-2 Concept lattice obtained with the context table of Fig. 4-1

4.2. Evaluation of the proposed approach

The evaluation is carried out by measuring the degree of correlation between the calculated similarity scores and scores obtained by human judgments. For this purpose, a questionnaire was administered to 30 respondents. The questionnaire asked each respondent to rank the likeness between 'electric kettle' and each of 17 home electric appliances. Respondents then rated the similarity of the pairs on a 1-17 scale, with lower numbers indicating higher similarity.

The comparison was carried out by calculating the correlation coefficient and the sum of squared errors.

The level of inconsistency of each questionnaire was calculated with the following formula.

$$d_i = \sum_j \left| q_{ij} - \mu_{ij} \right| \qquad \text{(Equation 27)}$$

Where q_{ij} is the value of the score that participant *i* submitted for pair *j* and μ_{ij} is the mean of the scores of all the users except that of user *i* for pair *j*.

Using Equation 27, questionnaires with values of d_i above two standard deviations from the mean $\bar{d}_i$ were excluded from the analysis. The inconsistency value per respondent (per each set of questionnaire) is shown in Fig. 4-3. It is obvious that respondent id 16, 17 and 19 are unreliable because they far away from the others in the curve. Their evaluation was not taken into consideration for this experiment. Refer to Appendix C for the questionnaire and their results.

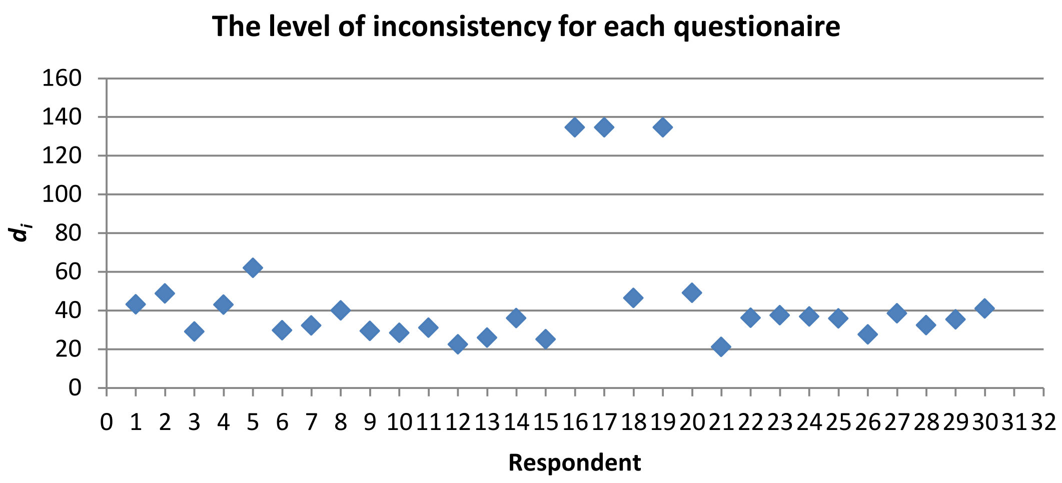


Fig. 4-3 The level of inconsistency for each questionnaire

The average standard deviations of the scores across respondents were also evaluated to identify inconsistencies. Since one of the questionnaires had a standard deviation lower than average, it was not taken into account. With this last change, the sample size was reduced from 30 to 27.

Finally, individual pair scores with one standard deviation below or above the pair mean were eliminated, which accounted for 4% of the total data. Fig. 4-4 shows the terms pair integrity and it is observed that all pairs are taken into account for this experiment.

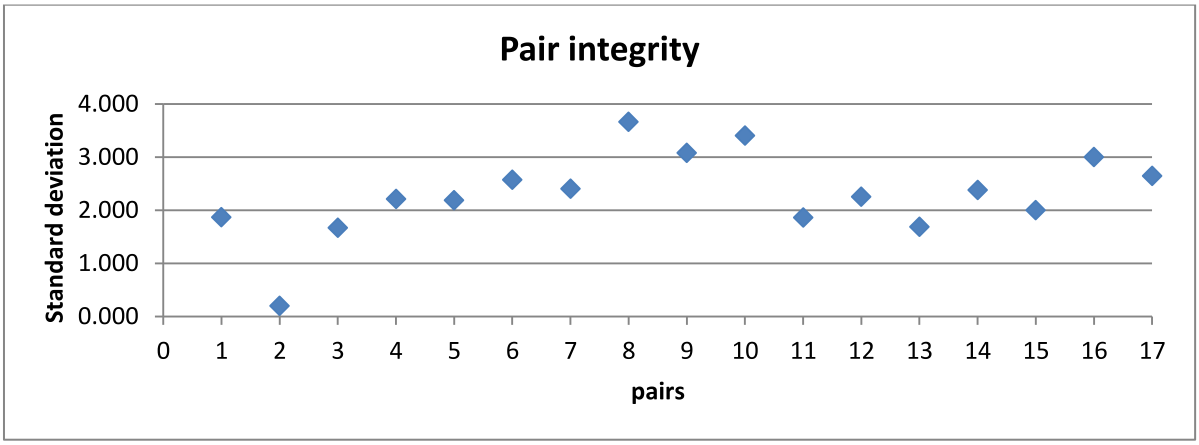


Fig. 4-4 Terms pair integrity

Subsequently, the average scores were normalized using the following transformation:

$$s_j = \frac{\overline{q}_j - q^{\min}}{q^{\max} - q^{\min}} \qquad \text{(Equation 28)}$$

where represents the similarity of pair j, $q^{\max} = 17$ and $q^{\min} = 1$. Values of s_j are shown in the first column of Table 4-1.

4.2.1. Similarity calculation

A program was developed in Java using the ontology library Jena [76]. The program reads the ontology and the names of the two classes to be compared. Firstly, it extracts the formal attribute information of each class in the ontology. Then, the program proceeds to calculate the cardinalities for each set of attributes, the minimum and maximum values, the number of common attributes, etc. Attributes of a class include those inherited from all of its parent classes. Similarity calculations are then carried out using the semantic similarity measures of the proposed approach as explained (Section 3.6). Then the Wu-Palmer's and Lin's similarities are calculated by edge counting, using the taxonomy structure of the ontology.

4.2.2. Experiment results

Table 4-1 summarizes the calculation results of the investigated similarities rating between 17 class comparisons.

Initially, the root node in the Wu-Palmer's similarity was set to 'home electric appliance'. For the reason explained in Chapter 2, N_3 becomes 0 for several pairs for which their common superclass happens to be the root node. Since these pairs clearly contain different classes, the result is incorrect. As a workaround we introduced 'device' as subclass of physical object (defined in ISO 15926) and made 'home electric appliance' a subclass of 'device'. From Table 4-1, it can be seen that the Overlap coefficient ($sim_{Overlap}$) with R=0.795 followed by the Wu-Palmer similarity with R=0.782, the Cosine similarity (sim_{Cosine}) with R=0.781, and Dice with (sim_{Dice}) with R=0.777.

After considering every possible combination of the similarity equation of Table 3.1 in the composite similarity equation (Equation 29), the best two combinations were:

$$sim_{Cosine+Jaccard}(C_1, C_2) = 1.887 sim_{Cosine} - 0.887 sim_{Jaccard} \qquad \text{(Equation 29)}$$

with a correlation of R=0.817 and

$$sim_{Dice+Jaccard}(C_i, C_j) = 1.966 sim_{Dice} - 0..966 sim_{Jaccard} \qquad \text{(Equation 30)}$$

with a correlation of R=0.816.

The weights of 1.887 and -0.887 and 1.966 and -0.966 for Equation 32 and 33, respectively were obtained by numeric optimization so as to minimize the residual sum of squares between the composite similarity and s_j of Equation 23.

4.2.3. Analysis of the results

To eliminate biases in the analysis of the results, we removed those pairs that produced squared errors greater than two times the standard deviation. The pairs (electric kettle, television set) and (electric kettle, electric oven) produced the biggest squared error. After removing both pairs, the correlation value of the Overlap coefficient increased to R=0.947. Again, sim_{Cosine}(R=0.922) and sim_{Dice} (R=0.919) were second and third in performance, respectively. For the combined similarities, $sim_{Cosine+Jaccard}$

increased to R=0.950 and increased to $sim_{Dice+Jaccard}$ R=0.947.

A hierarchical cluster analysis was also conducted in order to compare relatively homogeneous groups of results. The cluster analysis was equally applied to both the human assessment results and the results obtained with $sim_{Overlap}$. Clustering was carried out using Ward's minimum variance algorithm.

A comparison of the clusters indicates that most of the object pairs that belong to one cluster with $sim_{Overlap}$ also belong to a cluster in the results of human judgment. As shown in Fig. 4-5, only (electric kettle, television set), (electric kettle, air conditioner), and (electric kettle, bread machine) were grouped into another cluster. This is probably due to missing attributes in the FCA context table. Although another possible reason is that these two pairs were particularly difficult to judge during the answering of the questionnaire.

Table 4-1 Comparison between similarity measures

Electric kettle with:	$\mathrm{sim}_{Tversky}$	sim_{Cosine}	$\mathrm{sim}_{Jaccard}$	$\mathrm{sim}_{van\ der\ Weken\ 2}$	$\mathrm{sim}_{van\ der\ Weken\ 1}$	$\mathrm{sim}_{Overlap}$	$\mathrm{sim}_{All\ Confidence}$	sim_{Dice}	sim_{Lin}	$\mathrm{sim}_{Wu\ and\ Palmer}$ (with device class)	$\mathrm{sim}_{Wu\ and\ Palmer}$	Human judgment rank
Electric dish washer	0.22	0.22	0.13	0.87	0.9	0.25	0.2	0.22	0.01	0.29	-	0.29
Washing machine	0.22	0.22	0.13	0.87	0.9	0.25	0.2	0.22	0.01	0.29	-	0.32
Electric clothes dryer	0.5	0.5	0.33	0.93	0.93	0.5	0.5	0.5	0.18	0.57	0.4	0.62
Hair dryer	0.5	0.5	0.33	0.93	0.93	0.5	0.5	0.5	0.18	0.57	0.4	0.76
Water heater	1	1	1	1	1	1	1	1	1	1	1	1
Electric blanket	0.58	0.58	0.4	0.94	0.97	0.67	0.5	0.57	0.18	0.67	0.5	0.67
Toaster	0.41	0.41	0.25	0.87	0.93	0.5	0.33	0.4	0.18	0.5	0.33	0.73
Bread machine	0.41	0.41	0.25	0.87	0.93	0.5	0.33	0.4	0.18	0.5	0.33	0.52
Electric oven	0.35	0.35	0.2	0.8	0.92	0.5	0.25	0.33	0.18	0.5	0.33	0.8
Microwave oven	0.41	0.41	0.25	0.87	0.93	0.5	0.33	0.4	0.18	0.5	0.33	0.72
Vacuum cleaner	0.25	0.25	0.14	0.9	0.9	0.25	0.25	0.25	0.01	0.33	-	0.18
Television set	0.35	0.35	0.2	0.91	0.97	0.5	0.25	0.33	0.01	0.5	-	0.08
Room electric heater	0.58	0.58	0.4	0.94	0.97	0.67	0.5	0.57	0.2	0.67	0.5	0.81
Room air-conditioner	0.41	0.41	0.25	0.87	0.93	0.5	0.33	0.4	0.18	0.57	0.4	0.49
Conventional electric fan	0.25	0.25	0.14	0.9	0.9	0.25	0.25	0.25	0.01	0.33	-	0.25
Refrigerator	0.22	0.22	0.13	0.87	0.9	0.25	0.2	0.22	0.01	0.33	-	0.34
Blender	0.25	0.25	0.14	0.9	0.9	0.25	0.25	0.25	0.01	0.33	-	0.21
Correlation with human judgment	0.726	0.781	0.708	0.272	0.608	0.795	0.729	0.78	0.73	0.782		1
Sum of squared errors	0.93	0.662	1.576	3.532	3.921	0.484	0.941	0.69	2.71	0.474		

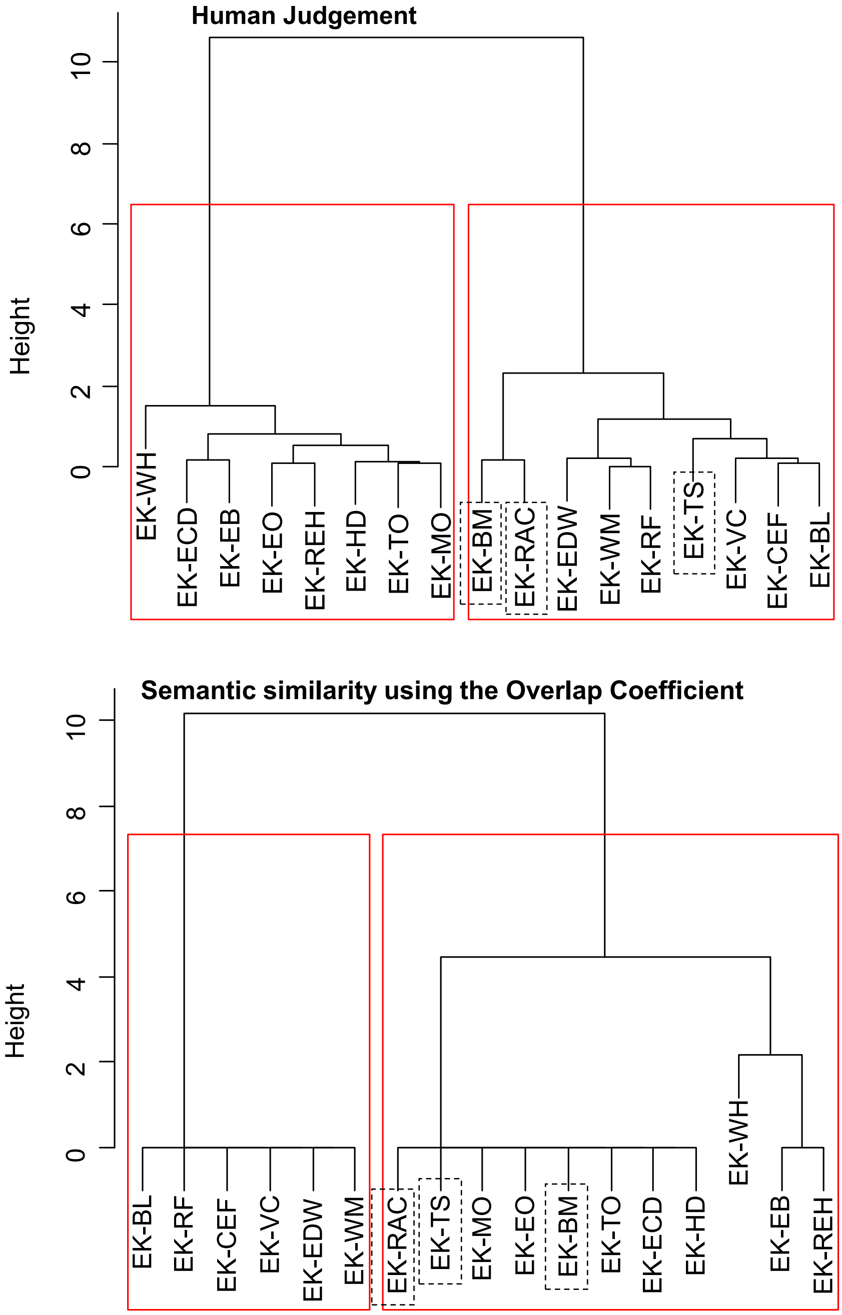


Fig. 4-5 Results of the cluster analysis.

4.3. Conclusions

This chapter presented semantic similarity measures of the proposed approach based on taxonomy that developed using FCA to determine the degree of similarity of two classes in the application of home electric appliance. The results of the experiment show that our approach performs better when compared against the Wu-Palmer similarity measure. In addition, while Wu-Palmer's similarity is only defined for trees, our approach can be applied to taxonomies containing a class with multiple direct-superclasses (multiple-inheritance). For example, similarity between two classes, N and B as shown in Chapter 2 (Fig.2-1) without taking any mathematical calculation we can see that the similarity is 0 due to both classes shares identical root class and the common superclass.

The proposed similarity measures are not only based on the taxonomy but also on the formal attributes that obtained using FAST in characterizing each class in the taxonomy. Consequently, formal attributes information can be used to calculate similarities in trees and lattices. Results of the numeric experiments showed that in all cases, the proposed semantic measures performed better than the similarities of Wu Palmer and Lin similarity measures.

In the electric appliance experiment, after removing the least performing pairs (electric kettle, television set) and (electric kettle, electric oven), the correlation saw an increase of approximately 25%. The reason might be that both television set and electric oven were characterized by processes which are unfamiliar to the common user. For example, toaster was characterized as a device that uses infrared radiation. In this case, infrared radiation was considered as a part of heating, which is directly related to toasting bread. Similarly, TV set was defined as a device that receives television signals.

When other devices were characterized in terms of processes and participating objects that were more familiar to the common user, the calculated similarities were close to the human judgments. However, albeit important to the designers, from a user point of view, subprocesses that are not directly perceived by the users (i.e. the mechanism with which a product achieves its given function) are probably not taken into account. This could be a limitation of the questionnaire approach for evaluating the similarities.

The use of formal concept analysis to develop taxonomy provides a degree of flexibility to a designer that is interested in developing something new. Formal concept analysis can provide the designer with not only the most similar product but also with a set of attributes that characterize it. Those attributes can provide an insight of the kind of solution (s)he is searching for. For example in the conceptual design of a plant, a designer might be interested in a device for heating. While specific technologies such as a microwave oven, an electric kettle or a water heater could potentially be useful, the designer might find it more useful to know about the characteristics of those technologies. As a result this extra knowledge could provide the designer the opportunity to think ‘outside the box‘.

Chapter 5

APPLICATION OF THE PROPOSED APPROACH TO PROCESS ONTOLOGY DEVELOPMENT

5.1. An ontology for machining processes

A manufacturing process aims to fulfill given requirements by transforming materials into objects that have specific shapes, structures, and other properties [77]. Several kinds of processes are commonly utilized, including mass-change, phase-change, structure-change, deformation, and consolidation processes.

A computer representation of manufacturing processes presents a range of potential benefits in areas such as product design and process planning [78], [79], [80], [81].

One approach to the computer representation of processes is by means of ontologies, which capture the semantics of things represented in a specific domain [82]. Ontologies are useful for knowledge representation and sharing, automated reasoning, and human-machine interfaces [83], [84].

In general, a domain ontology is composed of classes, relations and axioms [65]. A class represents a set of things that share the same attributes. For example, all the members of the class drilling use a drill to remove material and create a hole. A relation is a tuple that indicates a relationship between two or more things. Examples of relations are less than, connected to, and part of. In particular, the subclass relation is defined for organizing classes in the form of a class hierarchy. Axioms are typically represented as logical constructions that serve as formal definitions of a given class.

Several ontologies have been developed for generic knowledge representation in the domains of product and manufacturing including PRONTO [85], MASON [86], and ADACOR [87]. In addition, ontologies have been developed for specific manufacturing processes. For example, Grüninger and Delaval [88] developed a cutting process ontology that can be used in sheet-metal cutting design. There are a number of methodologies to develop ontologies including Uschold and King's method [89], Grüninger and Fox's method [90], Noy and McGuiness's method [82], the METHONTOLOGY framework [91], the Cyc methodology, KACTUS, SENSUS, and the On-To-Knowledge Methodology [92]. Some of these methodologies are briefly described in Appendix D.

One of the difficulties in ontology development is the lack of systematic methods for the design of the class hierarchy. This is caveat because an adequate class hierarchy is a key element in accurate and consistent ontologies [14]. At present, however, it is the current practice to develop class hierarchies in an ad-hoc manner, without the reasons and justifications of the class structure. Another technical challenge is how to define the axioms that constrain the meaning of the definitions in the ontology.

This chapter demonstrates the proposed semantic similarity method for the construction of an ontology for machining processes. The resulting machining processes ontology was evaluated and compared against MAnufacturing's Semantics ONtology (MASON) [86].

5.1.1. Ontology construction

Machining processes are commonly used to remove material and to modify the surfaces of objects that have usually been produced by other means. Several kinds of machining processes exist, including mechanical, electrical, chemical, laser, thermal, and hydrodynamic processes [93], [94]. For illustration purposes, the scope of this case study is limited to mechanical machining (i.e. those that use mechanical means to remove material). In order to develop the ontology, several common textbooks [94], [95], [96] and Internet sources were consulted. The potential classes are listed in the first column of Table 5-1.

For the preparation of the Formal Concept Analysis, attributes were selected based on FAST. Drilling is a hole-making process that produces a holed physical object by using a drill. The object that is transformed by a given drilling is a solid physical object. The object that is produced is also a solid physical object but with a hole in it. Next, constraints on performers and location are identified. For example, a drill is always involved in a drilling. Therefore, the formal attributes for drilling are: *changes a physical object*; *produces a holed object*; *involves a cutting tool to remove material*; and *uses a drill*.

Boring, reaming, taping, counterboring, spot facing, and countersinking also change a solid physical object and generate a solid physical object with a hole (a holed object). However, these four machining processes differ from drilling in that the work piece to be machined has already a hole. More differences can be found when we focus on the object that is produced by each of these processes: boring gives place to a physical object with a concentric axis; tapping produces a physical object with a threaded hole; counterboring, spot facing, and countersinking produce a physical object in which only a portion of the hole is enlarged. However, in counterboring the enlarged portion is also a hole in which the bottom part is flat and square. Therefore, the formal attributes of counterboring become: *consumes a physical object*; *changes a holed object*; *produces a holed object in which a portion of the hole is enlarged*; *enlarges a portion of an existing hole to a larger diameter*; *produces a holed object with an enlarged portion that is cylindrical*; *enlarges the end portion of the hole*; *produces a physical object in which the bottom part of the enlarged portion is flat and square*; and *involves a cutting tool to remove material*.

Table 5-1 summarizes the formal attributes for each potential class. For the location criterion, we could have referred to the machine where a given kind of process takes place. However, in the mechanical machining domain, there are different types of machines that range from manual lathes to computer numerical control machines. Because none of the machining processes always takes place in a given machine, the corresponding formal attributes are absent (for the same reason the machines are not considered as performers either). Based on the formal attributes of Table 5-1, a context table was created (Table 5-2). Subsequently, Concept Explorer [97] was used to generate the concept lattice. The resulting lattice is shown in Fig. 5-1.

After generating the lattice, object exploration was conducted to verify the completeness of the lattice. In object exploration, the modeler focuses at the relations between objects associated to a concept its subconcepts to see if they make sense.

Therefore, all paths in the lattice of Fig. 5-1are traced starting from the root node until reaching the bottom node.

During the object exploration, it was noticed that the lattice ignores the difference between reaming and boring despite the fact that textbooks and machining experts differentiate between them (Fig. 5-1). Another possible inconsistency is that counterboring is presented as a subclass of reaming.

To resolve these inconsistencies, we consulted the textbooks once again to disambiguate with more differences. Some textbooks pointed to differences on the surface finish of the product which was difficult to account for, particularly because tolerances differ among the different sources. A clear consistent difference was found in the tool (the performer) employed in reaming and boring. Reaming employs a multiple-tooth cutting tool called a reamer. On the other hand, boring always uses a single-point cutter (boring bar) [94], [98].

Table 5-1 List of potential classes and formal attributes for machining processes

	Object that is changed by the activity	Object that is produced by the activity	Performer	Composition
drilling	physical object	a holed object	involves a cutting tool to remove material, uses a drill	
boring	physical object, a holed object	a holed object , enlarged portion is cylindrical	involves a cutting tool to remove material	enlarges the end portion of the hole, enlarges a portion of an existing hole to a larger diameter
reaming	physical object, a holed object	a holed object, enlarged portion is cylindrical	involves a cutting tool to remove material	enlarges the end portion of the hole, enlarges a portion of an existing hole to a larger diameter
counterboring	physical object, a holed object	a holed object, enlarged portion is cylindrical, physical object in which the bottom part of the enlarged portion is flat and square	involves a cutting tool to remove material	enlarges the end portion of the hole, enlarges a portion of an existing hole to a larger diameter
milling			involves a rotating cutting tool to remove material	
blasting	physical object		involves an abrasive particles to remove material	
grinding	physical object		involves an abrasive particles to remove material	
taping	physical object, a holed object	enlarged portion is cylindrical, an internal thread hole	involves a cutting tool to remove material	enlarges a portion of an existing hole to a larger diameter
turning	physical object		involves a cutting tool to remove material	changed object is rotated
spot facing	physical object, a holed object	physical object in which the bottom part of the enlarged portion is flat and square, physical object in which the enlarged portion provides seat for a washer	involves a cutting tool to remove material	a holed object in which a portion of the hole is enlarged
lapping	physical object		involves an abrasive particles to remove material	
countersinking	physical object, a holed object	a holed object, physical object in which the enlarged portion provides a recess for a countersunk flat heat screw or countersunk rivet, produces a physical object in which the bottom part of the enlarged portion is cone-shaped	involves a cutting tool to remove material	a holed object in which a portion of the hole is enlarged, enlarged the end portion

Table 5-2 Preliminary context table

	Object that is changed by the activity					Object that is produced by the activity					Composition				Performer	
	changes a physical object	changes a holed object	produces a holed object	produces a threaded hole	enlarged portion is cylindrical	produces a holed object in which the bottom part of the enlarged portion is flat and square	produces a holed object in which the enlarged portion provides seat for a washer	produces a holed object in which the enlarged portion provides a recess for a countersunk flat heat screw or countersunk rivet	produces a holed object in which the bottom part of the enlarged portion is cone-shaped	changed object is rotated	enlarges the end portion of the hole	enlarges a portion of an existing hole to a larger diameter	uses a drill	involves a cutting tool to remove material	involves a rotating cutting tool to remove material	involves an abrasive particles to remove material
drilling	x		x										x	x		
boring	x	x	x		x							x		x		
reaming	x	x	x		x							x		x		
tapping	x	x	x	x	x							x		x		
counterboring	x	x	x		x	x					x	x		x		
spot facing	x		x			x	x							x		
coutersinking	x	x	x					x	x		x	x		x		
turning	x									x				x		
milling	x													x	x	
blasting	x															x
grinding	x															x
lapping	x															x

Similarly, grinding, lapping, and blasting were also shown as equivalent classes in the lattice. To verify this conclusion, textbooks were consulted focusing on these three classes, and it was found out that once again the difference was in the performer. Grinding is carried out with a tool called grinding wheel, which is a circular object made of abrasive materials bonded together. Lapping is a process that uses the so-called lap plate upon which abrasive slurry is placed. Blasting is characterized by the use of a high-pressure stream of abrasive particles which in some cases can be replaced with another fluid such as air or water [99].

Consequently, the inconsistencies can be corrected by adding the corresponding attributes which are shown at the dotted box of context table in Table 5-3. The revised lattice is shown in Fig. 5-3.

Note there are eight unnamed nodes (A, B, C, D, E, F, G, and H) in the lattice of Fig. 5-3. These are considered as newly discovered classes that can be identified based on the individual formal attributes and the parent nodes. These are named “machining process”, “machining that uses cutting tool”, “machining that produces a holed object”, “machining that changes a portion of an existing hole to a larger diameter”, “machining that produces an enlarged portion that is flat and square”, “machining that enlarges the end portion of the hole”, “machining that produces an enlarged portion that is cylindrical”, “machining that uses abrasive particles” respectively.

After analyzing and correcting the lattice, the resulting lattice and attribute information served as the basis to develop a computer-processable ontology using the Protégé ontology editor [100]. Protégé has a graphical user interface that facilitates the specification of classes, relations, and axioms. After editing the ontology, the user can save the ontologies in the OWL language, which is useful for automatic reasoning and integration. The resulting classes in the ontology are shown in Fig. 5-4.

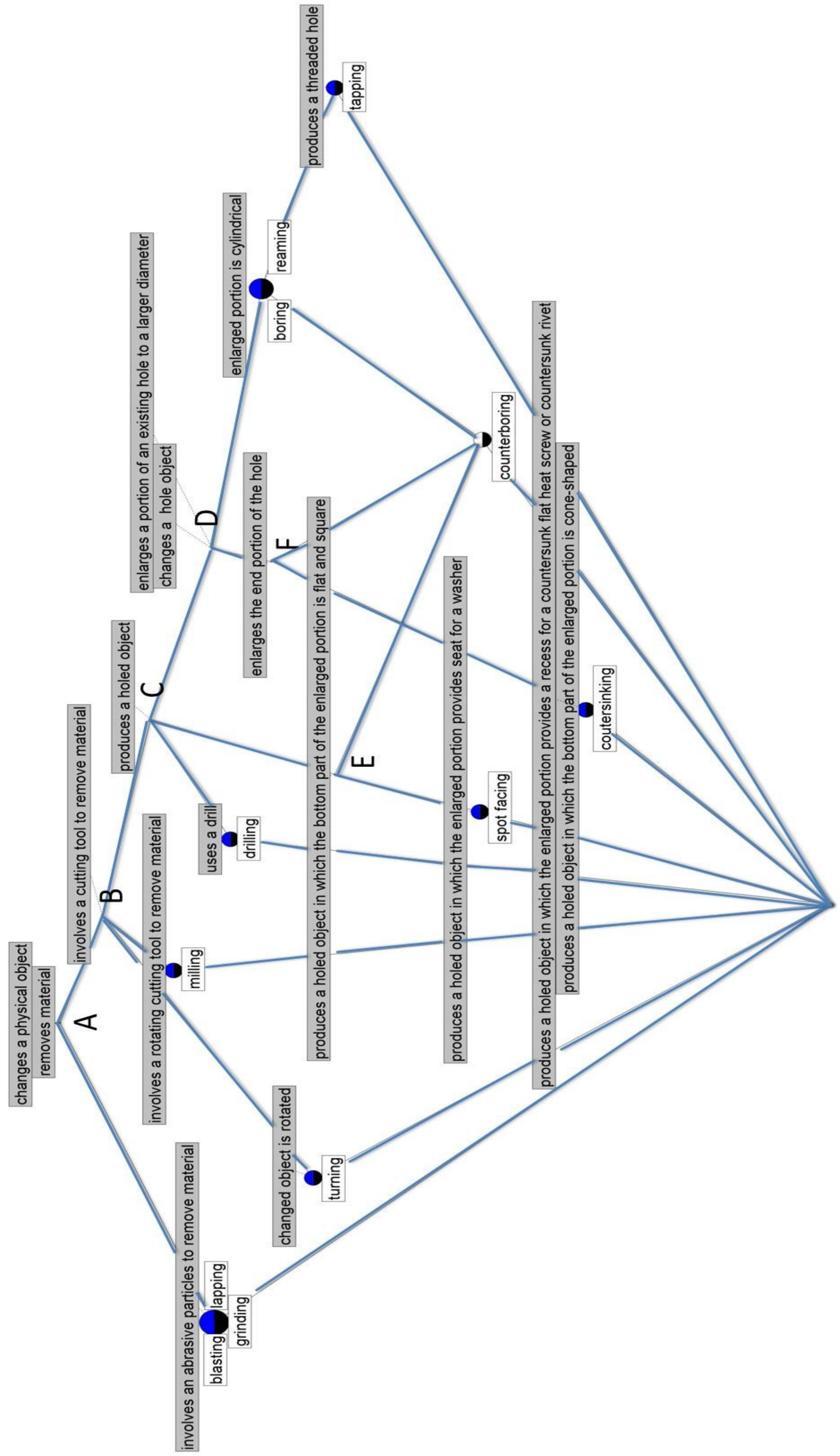
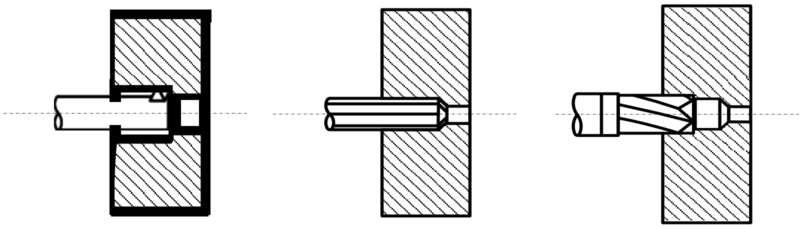


Fig. 5-1 Lattice obtained with the preliminary context table

The top node of the class hierarchy (machining_process) was made a subclass of activity in the upper ontology. This paper uses ISO 15926 but other upper ontologies can also be used.



(a) (b) (c)

Fig. 5-2 (a) boring enlarges a hole; (b) reaming produces a slightly enlarged a hole that has a more accurate diameter; (c) counterboring enlarges a part of the hole so that the bottom part of the enlarged portion of the hole is flat and square

Table 5-3 Modified context table.

	Object that is changed by the activity		Object that is produced by the activity							Composition			Performer								
	changes a physical object	changes a holed object	produces a holed object	produces a threaded hole	enlarged portion is cylindrical	produces a holed object in which the bottom part of the enlarged portion is flat and square	produces a holed object in which the enlarged portion provides seat for a washer	produces a holed object in which the enlarged portion provides a recess for a countersunk flat heat screw or countersunk rivet	produces a holed object in which the bottom part of the enlarged portion is cone-shaped	changed object is rotated	enlarges the end portion of the hole	enlarges a portion of an existing hole to a larger diameter	use of a high-pressure stream composed of abrasive particles and in some cases another fluid such as air or water	uses abrasive slurry	uses grinding wheel	uses a single-point cutting tool called a boring bar	uses a multiple-tooth cutting tool called a reamer	uses a drill	involves a cutting tool to remove material	involves a rotating cutting tool to remove material	involves an abrasive particles to remove material
drilling	x		x															x	x		
boring	x	x	x		x							x				x			x		
reaming	x	x	x		x							x					x		x		
tapping	x	x	x	x	x							x							x		
counterboring	x	x	x		x	x					x	x							x		
spot facing	x		x			x	x												x		
coutersinking	x	x	x					x	x		x	x							x		
turning	x									x									x		
milling	x																		x	x	
blasting	x												x								x
grinding	x														x						x

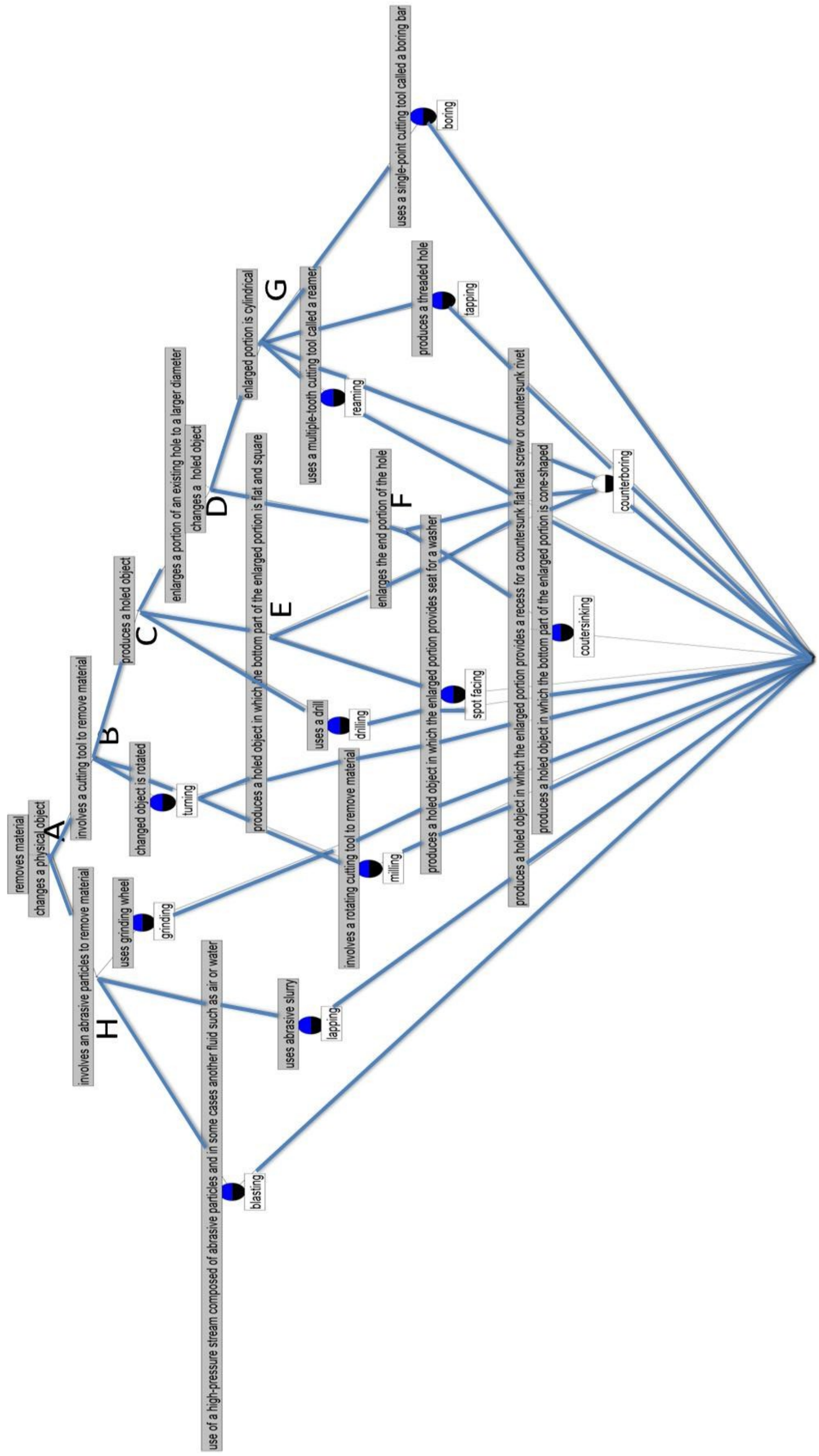


Fig. 5-3 The revised concept lattice

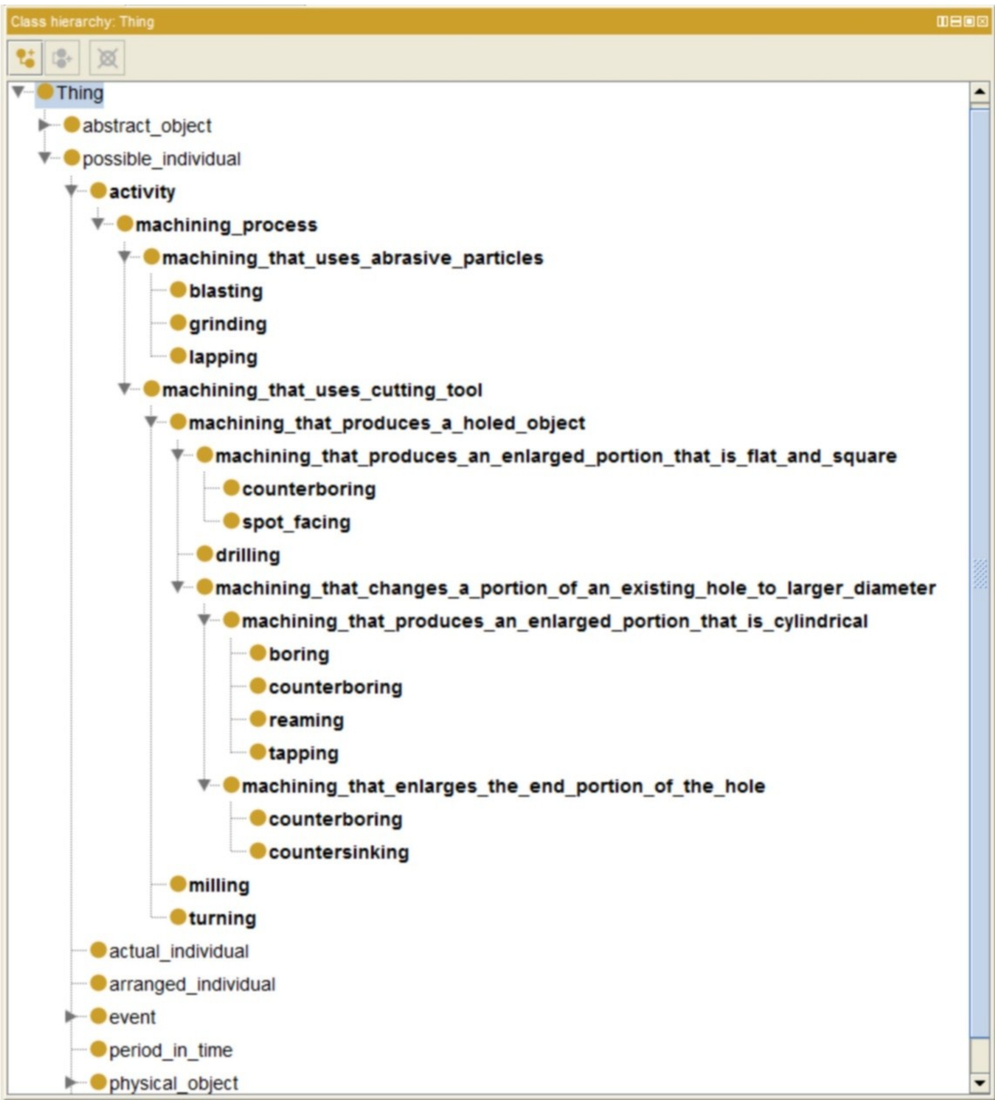


Fig. 5-4 Class hierarchy of machining processes

Note that all the machining operations presented so far share one thing in common: the involvement of phenomena such as plastic deformation, frictional forces, thermo-mechanical coupling and chip-and-burr formation [101]. These phenomena are also processes which correspond to parts of each of the machining processes (composition). Should the scope of the project be extended to include advanced machining processes, information about the physico-chemical phenomena will be necessary to emphasize some important differences, such as between a turning operation and a chemical machining.

5.1.2. Ontology evaluation and remediation

The machining ontology was evaluated and compared against the MAnufacturing's Semantics ONtology (MASON) [86]. The purpose of this evaluation was to determine the advantages of the proposed methodology.

Both ontologies distinguish between those processes based on abrasion and those processes that use a cutting tool (cutting in MASON). These two classes are grouped together as machining_process in our ontology and as Shearing_Operation in MASON. In both ontologies, drilling, milling and turning were grouped under the same class. However, our ontology differentiates between drilling, milling, and turning.

A numeric evaluation of the accuracy of each ontology was carried out using semantic similarity measures. For this purpose, in each ontology, we measure the similarity between two classes using the Wu-Palmer similarity measure (equation 2) [35].

Afterward, for each pair of classes, we compare the value of the Wu-Palmer similarity against the value of the NGD similarity (Equation 36) which is based on the normalized Google distance [102].

$$sim_{NGD} = 1 - \frac{\max(\log f(t_1), \log f(t_2)) - \log f(t_1, t_2)}{\log M - \min(\log f(t_1), \log f(t_2))} \qquad \text{(Equation 31)}$$

where $f(t_1)$, $f(t_2)$ and $f(t_1, t_2)$ give the number of hits for the terms t_1, t_2 and (t_1, t_2) respectively, each of which is obtained with a Web search engine. In this evaluation, t_1, t_2 are terms that correspond to the names of classes C_1 and C_2. M corresponds to the amount of indexed documents in a given Web search engine. For the Web search, we use Google Scholar, for which we assume M=5.8x10^8 based on an earlier estimate [103] and by assuming a growth rate of 2.7% based on the world-wide average annual increase of academic papers.

In order to restrict the Web search to the domain of study, keywords in both search engines were formulated with the inclusion of the term "machining" and search was carried out using double quotes.

For example, for the similarity between counterboring and spot facing, search with Scholar for "machining" "counterboring" results with Scholar was f(counterboring)=1019 hits; search for "machining" "spot facing" produces f(spot facing)=620 hits; and search for "machining" "counterboring" "spot facing" results in f(counterboring, spotfacing) = 56 hits. Substituting these values in Equation 31 we obtain v(counterboring, spot facing) = 0.7854.

The evaluation was carried out by groups of n classes each of which was compared against it and the remaining n-1 classes. Table 5-4 shows the result of the first group in the machining ontology, which corresponds to the pair comparisons for C_1=counterboring. Since there are 12 target classes in the machining ontology (n=12) and 17 target classes in MASON (n=17), the total number of calculated similarities were 12^2 and 17^2, respectively. The complete sets of results are shown in the Appendix F.

We assess and compare the ontologies by their performance against the NGD similarity, measured by the correlation coefficient (R), the Root Mean Squared Error ($RMSE$), and the Mean Absolute Percentage Error ($MAPE$) of each of the pairs (C_i, C_j) $\forall i = 1..n, j = 1..n$.

Table 5-4 Evaluation of C_1=counterboring using the class hierarchy of the machining ontology

C_2	sim_{NGD}	$sim_{Wu\ and\ Palmer}$
counterboring	1.00	1.00
milling	0.55	0.50
countersinking	0.86	0.83
drilling	0.61	0.67
spot facing	0.80	0.80
boring	0.69	0.83
reaming	0.77	0.83
turning	0.54	0.67
tapping	0.70	0.83
grinding	0.53	0.25
blasting	0.55	0.25
lapping	0.59	0.25
RMSE		0.17
MAPE		0.13
R		0.79

Then, the average *RMSE* of each group was calculated by summing the individual *RMSE* for each pair (C_i, C_j) and then dividing the total by n. Also considered were the minimum and maximum values of *RMSE*. Similar calculations were carried out for *MAPE* and *R*. Table 5-5 summarizes the results for each class in the machining ontology.

It was noticed that the group that corresponds to the class of tapping (C_1=tapping) had a correlation coefficient of 0.06 which is less than the 1/10 of the average correlation in all the groups. Using a sample of 30 search results obtained with Scholar we verified that the result was not due to false positives. False positive is the errors of retrieving results that are not fulfill the condition. Therefore the result suggests that the position of the class in the class hierarchy is inadequate and can be improved.

Table 5-5 Performance of each class in the machining ontology

C_1	*RMSE*	*MAPE*	*R*
counterboring	0.17	0.13	0.79
milling	0.24	0.20	0.67
countersinking	0.20	0.15	0.86
drilling	0.24	0.20	0.45
spot facing	0.18	0.14	0.87
boring	0.28	0.22	0.61
reaming	0.25	0.20	0.86
turning	0.26	0.22	0.60
tapping	0.39	0.30	0.06
grinding	0.35	0.30	0.70
blasting	0.32	0.28	0.81
lapping	0.34	0.30	0.84
Average	0.27	0.22	0.68
Minimum	0.39	0.30	0.87
Maximum	0.17	0.13	0.06

Table 5-6 shows the average, minimum, and maximum values of all groups for both ontologies. The values obtained after removing the group of the class tapping are also included.

Table 5-6 Average, minimum, and maximum values of *RMSE*, *MAPE* and *R*

	Machining ontology developed with the proposed method						MASON		
	All classes			Group of tapping removed					
	Average	Max	Min	Average	Max	Min	Average	Max	Min
RMSE	0.27	0.39	0.17	0.26	0.35	0.17	0.28	0.45	0.23
MAPE	0.22	0.30	0.13	0.21	0.30	0.13	0.22	0.41	0.16
R	0.67	0.87	0.06	0.73	0.87	0.45	0.52	0.88	0.29

Small differences in *RMSE* and *MAPE* were found between both ontologies. However, the correlation coefficient of the machining ontology presented an improvement of 29-40% with respect to that of MASON.

5.2. Web-based Evaluation of Semantic Similarity Measures

The evaluation of different similarity measure is similar to the experiments conducted in section 4.2. However, this evaluation was made by comparing against the NGD similarity. The similarity measure presented in Chapter 3 was applied for pairs of classes in the machining process ontology. These similarity results were compared with similarity calculations obtained on internet search engine using Google's Scholar and Elsevier's Scirus.

The calculation was carried out for pairwise similarities between all the pairs of machining processes, resulting in 79 comparisons. The resulting similarity scores were compared against the Web-based similarity denoted by Equation 25. The same keyword condition as described in Section 5.1.2 was repeated where the term machining was included and the hit counts were used to evaluate all pair's comparison.

The correlation coefficient R, the sum of squared error SSE and standard deviation 2σ were used to assess and compare the performance of each similarity measures.

Table 5-7 summarizes the results of calculations. Interestingly, the Jaccard coefficient has high correlation value of R=0.751 and R=0.779 for both comparison against Scirus and Google Scholar, respectively, as can be seen from the Table 5-7. Notice that, Lin similarity has the largest value of sum of squared error because most of the classes which have been compared in the machining ontology are lacking of subclasses. This situation refers to $(h(C)$ in Equation 6. Largest value of sum of squared error shows the less accuracy of information-based similarity measure.

Then, the results of each similarity measure are evaluated using the studentized residual analysis. The purpose of this evaluation is to identify unusual observation that produce residual outside the 95% confidence limits. This unusual observation is considered as outlier. Here, an outlier is a point that far away from the pattern described by the other points does not comply with the general behavior of data [58].

An influence (or bubble) plot is useful to observe studentized residuals ($studres_i$), hat values (h_i) and Cook's distance (D_i) on a single plot for each semantic similarity measures. For example, Fig. 5-5 shows an influence plot Wu Palmer similarity measure in comparison against the Google Scholar. The horizontal axis represents the hat-values, the vertical axis represents the studentized residuals. The area of the circle represents the leverage and the residual information. The larger the size of the circle, the larger is the impact of an unusual observation.

Table 5-7 The results for experiments of material removal process

	Pairs	Scirus	Scholar	$sim_{Wu\ and\ Palmer}$	sim_{Lin}	sim_{Dice}	$sim_{All\ Confidence}$	$sim_{Overlap}$	$sim_{van\ der\ Weken\ 1}$	$sim_{van\ der\ Weken\ 2}$	$sim_{Jaccard}$	sim_{Cosine}	$sim_{Tversky}$
Counterboring with	counterboring	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	milling	0.543	0.546	0.500	0.111	0.533	0.400	0.800	0.929	0.684	0.364	0.566	0.400
	countersinking	0.862	0.857	0.545	0.639	0.800	0.800	0.800	0.857	0.857	0.667	0.800	0.800
	drilling	0.607	0.605	0.667	0.184	0.625	0.500	0.833	0.929	0.722	0.455	0.645	0.500
	spotfacing	0.785	0.797	0.800	0.639	0.706	0.600	0.857	0.929	0.765	0.545	0.717	0.600
	boring	0.514	0.687	0.833	0.471	0.842	0.800	0.889	0.929	0.867	0.727	0.843	0.889
	reaming	0.763	0.766	0.833	0.471	0.842	0.800	0.889	0.929	0.867	0.727	0.843	0.889
	turning	0.548	0.538	0.500	0.111	0.533	0.400	0.800	0.929	0.684	0.364	0.566	0.400
	tapping	0.695	0.703	0.667	0.244	0.842	0.800	0.889	0.929	0.867	0.727	0.843	0.889
	grinding	0.545	0.534	0.250	0.016	0.400	0.300	0.600	0.857	0.632	0.250	0.424	0.300
	blasting	0.531	0.547	0.250	0.016	0.400	0.300	0.600	0.857	0.632	0.250	0.424	0.300
	lapping	0.564	0.593	0.250	0.016	0.400	0.300	0.600	0.857	0.632	0.250	0.424	0.300
Milling with	milling	1.000	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	countersinking	0.581	0.566	0.444	0.111	0.533	0.400	0.800	0.929	0.684	0.364	0.566	0.400
	drilling	0.836	0.798	0.571	0.111	0.727	0.667	0.800	0.944	0.895	0.571	0.730	0.667
	spotfacing	0.515	0.526	0.500	0.111	0.667	0.571	0.800	0.941	0.842	0.500	0.676	0.571
	boring	0.686	0.729	0.444	0.111	0.571	0.444	0.800	0.933	0.737	0.400	0.596	0.444
	reaming	0.674	0.657	0.444	0.111	0.571	0.444	0.800	0.933	0.737	0.400	0.596	0.444
	turning	0.883	0.805	0.667	0.111	0.800	0.800	0.800	0.947	0.947	0.667	0.800	0.800
	tapping	0.706	0.669	0.444	0.111	0.571	0.444	0.800	0.933	0.737	0.400	0.596	0.444
	grinding	0.886	0.800	0.333	0.016	0.600	0.600	0.600	0.895	0.895	0.429	0.600	0.600
	blasting	0.690	0.635	0.333	0.016	0.600	0.600	0.600	0.895	0.895	0.429	0.600	0.600
	lapping	0.668	0.651	0.333	0.016	0.400	0.300	0.600	0.857	0.632	0.250	0.424	0.300
Countersinking with	countersinking	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	drilling	0.636	0.628	0.600	0.184	0.625	0.500	0.833	0.929	0.722	0.455	0.645	0.500
	spotfacing	0.771	0.798	0.545	0.184	0.588	0.500	0.714	0.857	0.706	0.417	0.598	0.500
	boring	0.509	0.674	0.667	0.317	0.737	0.700	0.778	0.857	0.800	0.583	0.738	0.700
	reaming	0.789	0.775	0.667	0.317	0.737	0.700	0.778	0.857	0.800	0.583	0.738	0.700
	turning	0.567	0.542	0.444	0.111	0.533	0.400	0.800	0.929	0.684	0.364	0.566	0.400
	tapping	0.709	0.712	0.667	0.317	0.737	0.700	0.778	0.857	0.800	0.583	0.738	0.700
	grinding	0.556	0.540	0.222	0.016	0.400	0.300	0.600	0.857	0.632	0.250	0.424	0.300
	blasting	0.580	0.598	0.222	0.016	0.400	0.300	0.600	0.857	0.632	0.250	0.424	0.300
	lapping	0.598	0.595	0.222	0.016	0.400	0.300	0.600	0.857	0.632	0.250	0.424	0.300
Drilling with	drilling	1.000	0.988	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	spotfacing	0.566	0.573	0.667	0.184	0.769	0.769	0.833	0.941	0.889	0.625	0.772	0.714
	boring	0.697	0.773	0.600	0.184	0.667	0.667	0.833	0.933	0.778	0.500	0.680	0.556
	reaming	0.729	0.723	0.600	0.184	0.667	0.667	0.833	0.933	0.778	0.500	0.680	0.556
	turning	0.808	0.767	0.571	0.111	0.727	0.727	0.800	0.944	0.895	0.571	0.730	0.667
	tapping	0.762	0.736	0.600	0.184	0.667	0.667	0.833	0.933	0.778	0.500	0.680	0.556
	grinding	0.814	0.763	0.286	0.016	0.545	0.545	0.600	0.889	0.842	0.375	0.548	0.500
	blasting	0.727	0.663	0.286	0.016	0.545	0.545	0.600	0.889	0.842	0.375	0.548	0.500
	lapping	0.670	0.654	0.286	0.016	0.545	0.545	0.600	0.889	0.842	0.375	0.548	0.500
Spotfacig with	spotfacing	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	boring	0.474	0.645	0.800	0.184	0.625	0.556	0.714	0.867	0.765	0.455	0.630	0.556
	reaming	0.694	0.714	0.800	0.184	0.625	0.556	0.714	0.867	0.765	0.455	0.630	0.556
	turning	0.502	0.506	0.500	0.111	0.667	0.571	0.800	0.941	0.842	0.500	0.676	0.571
	tapping	0.638	0.668	0.800	0.184	0.625	0.556	0.714	0.867	0.765	0.455	0.630	0.556
	grinding	0.494	0.494	0.250	0.016	0.500	0.429	0.600	0.882	0.789	0.333	0.507	0.429

	blasting	0.446	0.513	0.250	0.016	0.500	0.429	0.600	0.882	0.789	0.333	0.507	0.429
	lapping	0.536	0.572	0.250	0.016	0.500	0.429	0.600	0.882	0.789	0.333	0.507	0.429
Boring with	boring	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	reaming	0.601	0.794	0.833	0.471	0.889	0.889	0.889	0.933	0.933	0.800	0.889	0.889
	turning	0.669	0.723	0.444	0.111	0.571	0.444	0.800	0.933	0.737	0.400	0.596	0.444
	tapping	0.602	0.783	0.833	0.471	0.889	0.889	0.889	0.933	0.933	0.800	0.889	0.889
	grinding	0.671	0.698	0.222	0.016	0.429	0.333	0.600	0.867	0.684	0.273	0.447	0.333
	blasting	0.570	0.675	0.222	0.016	0.429	0.333	0.600	0.867	0.684	0.273	0.447	0.333
	lapping	0.551	0.707	0.222	0.016	0.429	0.333	0.600	0.867	0.684	0.273	0.447	0.333
Reaming with	reaming	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	turning	0.601	0.639	0.444	0.111	0.571	0.444	0.800	0.933	0.737	0.400	0.596	0.444
	tapping	0.661	0.805	0.833	0.471	0.889	0.889	0.889	0.933	0.933	0.800	0.889	0.889
	grinding	0.805	0.636	0.222	0.016	0.429	0.333	0.600	0.867	0.684	0.273	0.447	0.333
	blasting	0.658	0.660	0.222	0.016	0.429	0.333	0.600	0.867	0.684	0.273	0.447	0.333
	lapping	0.662	0.691	0.222	0.016	0.429	0.333	0.600	0.867	0.684	0.273	0.447	0.333
Turning with	turning	1.000	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	tapping	0.707	0.709	0.444	0.111	0.571	0.444	0.800	0.933	0.737	0.400	0.596	0.444
	grinding	0.866	0.855	0.333	0.016	0.600	0.600	0.600	0.895	0.895	0.429	0.600	0.600
	blasting	0.643	0.651	0.333	0.016	0.600	0.600	0.600	0.895	0.895	0.429	0.600	0.600
	lapping	0.662	0.658	0.333	0.016	0.600	0.600	0.600	0.895	0.895	0.429	0.600	0.600
Tapping with	tapping	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	grinding	0.693	0.839	0.222	0.016	0.429	0.333	0.600	0.867	0.684	0.273	0.447	0.333
	blasting	0.727	0.847	0.222	0.016	0.429	0.333	0.600	0.867	0.684	0.273	0.447	0.333
	lapping	0.716	0.877	0.222	0.016	0.429	0.333	0.600	0.867	0.684	0.273	0.447	0.333
Grinding with	grinding	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	blasting	0.730	0.679	0.667	0.545	0.800	0.800	0.800	0.947	0.947	0.667	0.800	0.800
	lapping	0.745	0.732	0.667	0.545	0.800	0.800	0.800	0.947	0.947	0.667	0.800	0.800
Blasting with	blasting	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	lapping	0.754	0.718	0.667	0.545	0.800	0.800	0.800	0.947	0.947	0.667	0.800	0.800
Lapping with	lapping	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Correlation with Scirus	1	0.938	0.628	0.625	0.714	0.731	0.611	0.661	0.707	0.751	0.717	0.713
	Sum of squared errors (Scirus)			5.431	18.898	1.793	2.766	1.570	4.505	1.776	4.635	1.475	2.997
	2σ (Scirus)			0.161	0.393	0.057	0.081	0.059	0.107	0.071	0.118	0.052	0.083
	Correlation with Scholar	0.938	1	0.678	0.736	0.738	0.744	0.632	0.611	0.680	0.779	0.737	0.735
	Sum of squared errors (Scholar)			5.582	19.871	1.662	3.006	1.280	3.865	1.498	5.007	1.478	3.198
	2σ (Scholar)			0.196	0.400	0.076	0.119	0.047	0.095	0.054	0.154	0.069	0.118

Observations No. 71, 72 and 73 in the regression analysis for Wu and Palmer as shown in Fig. 5-5 represents the unusual observation. Table 5-8 summarizes the observations that have studentized residual that are outside the ±2 range (95% confidence limits) for each similarity measure. However, out of 10 semantic similarity measures only Wu and Palmer similarity, Lin similarity, both van der Weken similarity measures and Jaccard coefficient have the unusual observation. It can be seen that, the Wu and Palmer similarity and Lin similarity share common observations No. of 71, 72 and 73 that considered as the unusual observation. Appendix G shows the results of regression analysis for other similarity measures.

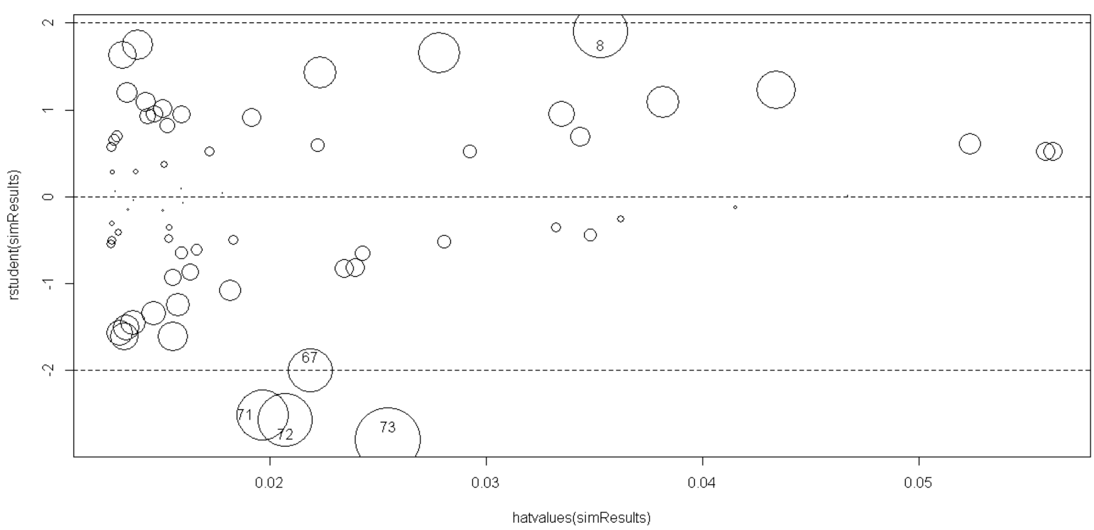


Fig. 5-5 Regression analysis for Wu Palmer similarity

It was noticed that the class of tapping appears three times in the unusual observations: (71(tapping, grinding), 72(tapping, blasting), 73(tapping, lapping)). Therefore, we identified class of tapping as the outlier. Hence the results encounters with suspicion are the observation which resulted either from a mistake or other irrelevant effects. It is suggested that by modifying the outlier, it changes the coefficient substantially.

It has been verified that the result of an unusual observation was not due to false positives. Therefore the result suggests that the position of the class of tapping in the class hierarchy is inadequate and can be improved.

Table 5-8 The studentized residual for each similarity measure in comparison against the Google Scholar

Similarity measure	observation	pair	residual
$sim_{Wu\ and\ Palmer}$	67	(turning, grinding)	-2.000
	71	(tapping, grinding)	-2.508
	72	(tapping, blasting)	-2.569
	73	(tapping, lapping)	-2.803
sim_{Lin}	67	(turning, grinding)	-2.543
	71	(tapping, grinding)	-2.377
	72	(tapping, blasting)	-2.460
	73	(tapping, lapping)	-2.778
$sim_{van\ der\ weken1}$	1	(counterboring, counterboring)	2.570
$sim_{van\ der\ weken2}$	1	(counterboring, counterboring)	2.185
$sim_{Jaccard}$	1	(counterboring, counterboring)	2.210

To resolve this inadequacy, several textbooks were being referred to identify the suitable position of tapping in the class hierarchy. Tapping is the process of cutting an internal thread [96], [94]. The textbooks pointed out that, other methods to produce internal thread can be performed by milling and turning. Therefore the class of tapping is removed from the subclass of machining that produces enlarged portion is cylindrical to the subclass of involves of cutting tool to removes material which is shared with milling and turning. Fig. 5-6 shows the concept lattice after repositioning class of tapping in the machining process ontology. After repositioning, the correlation coefficient of Jaccard coefficient against Scirus and Scholar is increased to 0.758 (with p-value of 5.97×10^{-16}) and 0.824 (with p-value of 1.15×10^{-20}), respectively. Table 5-9 shows the improved correlation coefficient of all similarity measures for refined machining ontology.

Table 5-9 The correlation coefficient, sum of squared errors and standard deviation for machining ontology

	Scirus	Scholar	$\mathrm{sim}_{Wu\ and\ Palmer}$	sim_{Lin}	sim_{Dice}	$\mathrm{sim}_{All\ Confidence}$	$\mathrm{sim}_{Overlap}$	$\mathrm{sim}_{van\ der\ Weken\ 1}$	$\mathrm{sim}_{van\ der\ Weken\ 2}$	$\mathrm{sim}_{Jaccard}$	sim_{Cosine}	$\mathrm{sim}_{Tversky}$
Correlation with Scirus	1	0.73	0.671	0.612	0.73	0.739	0.601	0.666	0.694	0.758	0.729	0.726
Sum of squared errors (Scirus)			5.092	20.17	1.48	2.102	1.728	3.338	1.725	3.253	1.406	2.176
2σ (Scirus)			0.15	0.397	0.05	0.06	0.059	0.09	0.058	0.09	0.05	0.063
Correlation with Scholar	0.73	1	0.726	0.754	0.79	0.793	0.672	0.663	0.703	0.824	0.794	0.784
Sum of squared errors (Scholar)			5.063	20703	1.3	2.666	1.171	4.051	1.579	4.595	1.134	2.708
2σ (Scholar)			0.151	0.382	0.04	0.08	0.044	0.093	0.056	0.114	0.036	0.08

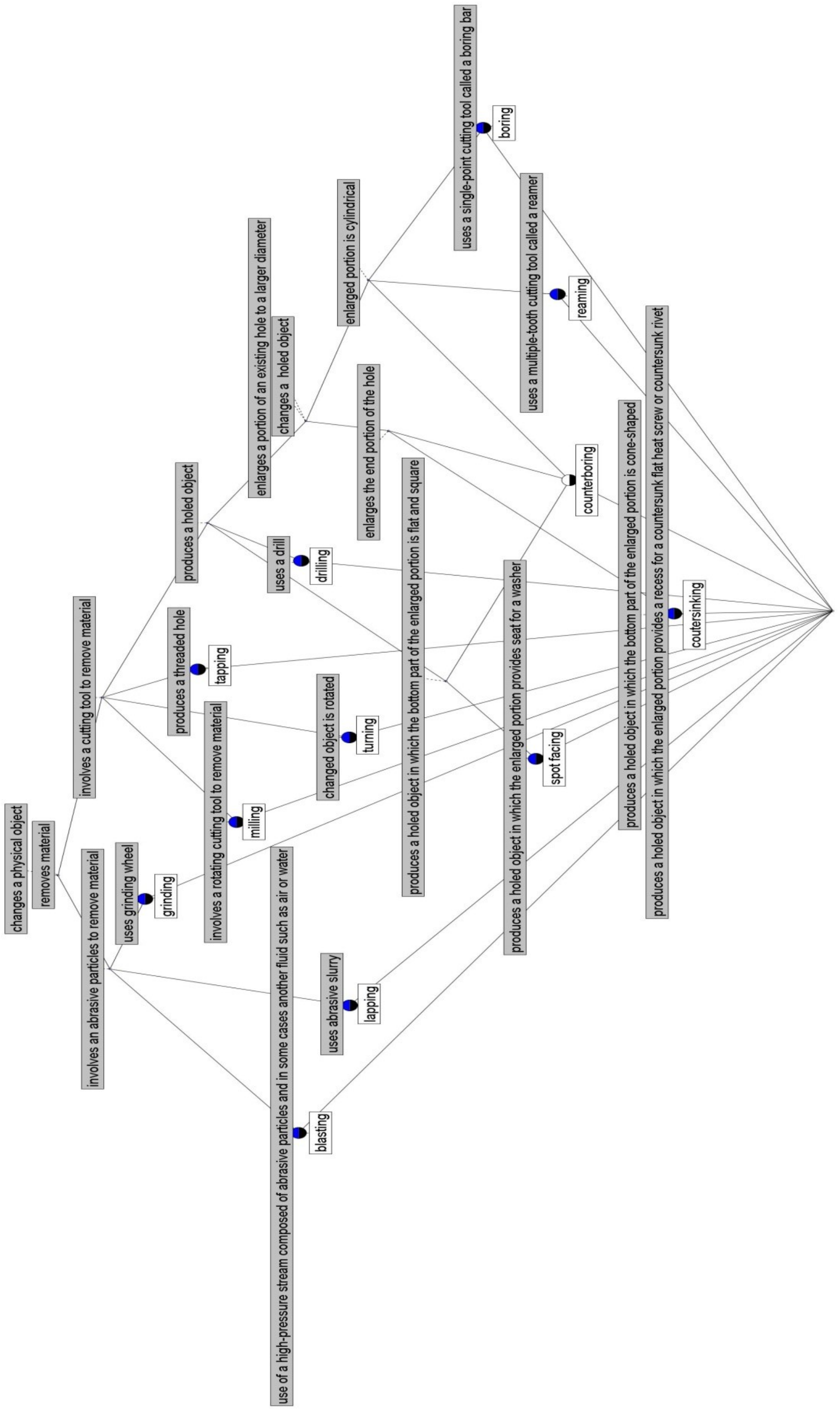


Fig. 5-6 The repositioning of class tapping in machining ontology

5.3. Conclusions

This chapter showed how the proposed method can be used for ontology development.

This chapter illustrated the proposed approach with the development of an ontology for machining processes. The results show the benefits of the proposed methodology both in terms of the correctness of the class hierarchy and the documentation of the design rationale of the ontology.

The pairwise comparison of semantic similarities and the NGD similarities served as a mechanism for two purposes: 1) identifying inconsistent classes in the ontology and 2) providing a global score of the accuracy of the ontology.

After the ontology has been developed, the resulting formal attribute information can also serve to document the design rationale of the ontology. In contrast, existing ontology development methods are based on ad-hoc choices which leave little or no explicit reasons behind the decisions made.

The results of the correlation between the different semantic similarities and Web-based search suggest that multiple similarity measures can be used as a way to validate ontologies. The reason is that the accuracy of the ontology directly influences the correlation values.

Finally, in the experiment of comparison against existing similarity measure, after reposition of class of tapping, the correlation saw an increase of approximately 0.50-5.0% for the Jaccard coefficient against web-based similarity. The results show the ontology was validated, thus proving the adequacy of the proposed methodology. By further modifying the attributes we believe that its performance may increase even more.

Chapter 6

APPLICATION OF THE PROPOSED APPROACH FOR PRODUCT-SERVICE SYSTEMS DEVELOPMENT

6.1. Introduction

Today's industry has developed a number of strategies for addressing sustainability and environment including, minimization of waste and use of environmentally-friendly materials, as well as integration of products and services. Shifting to service-oriented solutions that integrate products with service provision has becoming another alternative for developing new products with less environmental impact. The aim of this business strategy is twofold, firstly it enables enterprises to satisfy the need of the customer through customized solutions and secondly it reduces the environmental impact which can be used to gain competitive advantage [104]. Therefore, many concepts have emerged to realize it, including product service system, dematerialization, functional product, service engineering or servitization [105].

A Product service system (PSS) is a mix of both products and services aimed at better sustainability of both production and consumption. Baines et al. define a product service system as a system of products, services, supporting networks, and infrastructure that is capable to satisfy customer needs with a competitive and environmentally advantage [105].

The design of PSS systems is multi-disciplinary in nature, requiring a wide variety of product life-cycle knowledge [10]. Many tools and methods for PSS design have been proposed in the literature [106], [107], [108]. Existing tools and methods have been developed to address the detailed design and implementation of PSS. None of the existing PSS design approaches provide support the design of PSS systems at the early stage of the design.

Typically, the conceptual design stage starts with a problem definition in which a designer may not have complete knowledge and information related to the problem to be solved. Therefore, at this stage of product development, the designer is not only concerned with finding the solution to the problem to be solved but also in defining the problem itself. Moreover, product design highly depends on the knowledge or experience of skilled designers whose performance can be hampered by inadequate information sharing and exchange [109].

Different approaches have been suggested to support the designer in clarifying the problem definition. As an example, the black box model has been widely used to decompose the problem into sub-problems and then map them to the generic function of the design without thinking of any solutions [110]. Each black box models a real system from scratch and it has material, signal and energy as input and output elements for the system. The result of this technique is a list of required functions or customer requirements. Decomposing the problem is not a simple task as the designer requires in-depth knowledge of the system that is being created. The kinds of input and output elements are always ambiguous and designer faces difficulties in clearly defined them. This may leads to an insufficient problem definition. As a result, the designer needs other tools for clarifying the problem itself and the kind of input required as well the expected output.

Failing in clarifying the problem definition diminishes the effectiveness of the strategy selection of product-services which is an important component in PSS design. The design team should have a clear understanding on the problem to be solved to avoid modifications at later stages of the life-cycle [109]. Therefore, some authors have suggested the use of case-based reasoning to facilitate the service strategy selection in which PSS design problems are solved by using or adapting previously obtained design solutions [109], [111], [9]. CBR enables the designer to avoid repetition of previous mistakes and to achieve best practices in PSS design [112].

Therefore, this chapter provides an application of the proposed semantic similarities for selecting the services strategies for PSS in the conceptual phase of design. A methodology for selecting the service strategies was proposed by incorporating product-class-comparison based on the proposed semantic similarity measure implemented in a CBR system. A product ontology was developed to represent the product for PSS. The effectiveness of the proposed semantic similarity measure in the context of PSS design is discussed in conjunction with a case study.

6.2. Selection of services in product service-systems

One of the main issues in PSS consists on determining the type of service that can be integrated with a given product [15]. Lin et al. [15] propose an approach based on case-based reasoning for selecting service strategies. For the retrieval and comparison of cases, the system compares the attributes of cases that were designed in the past with those of the new problem. Once a case is retrieved, its solution is analyzed to see if it can be used to solve the new problem. A total of 47 cases were extracted from past PSS cases and stored in a case library. A case is described by 12 features that are classified into three categories, namely, user behavior, product and environmental environment. User behavior is specified in terms of place of usage, and frequency of usage. The product is specified in terms of product fashion cycle, volume, weight, useful life, price, and subsequent expenditure. External environment is defined in terms of GDP per capita, population density, area of territory, and temperature range. Each feature is specified with an integer representing a discrete value or ranges of values. The global similarity measure proposed by Kolodner and Simpson [113] is used to calculate the case similarity:

$$S(t,r)=\frac{\sum_{i=1}^{n} w_i \times sim\left(f_i^t, f_i^r\right)}{\sum_{i=1}^{n} w_i} \qquad \text{(Equation 32)}$$

where $S(t,r)$ is the global similarity between the target case t and a source case r, w_i is the weight of feature i, f_i^t is the value of feature i of target case t and f_r^t is the value of feature *i* of an source case *r*.

sim(f_i^t, f_i^r) is calculated according to the following criteria, which is based on the overlap coefficient and a similarity for numerical attributes.

$$sim(f_i^t, f_i^r) = \begin{cases} \frac{|A_i^t \cap A_i^r|}{\min(|A_i^t|, |A_i^r|)} & \text{if } f_i^t, f_i^r \text{ are ontology classes} \\ 1 - \frac{|f_i^t - f_i^r|}{f_i^{max} - f_i^{min}} & \text{if } f_i^t, f_i^r \text{ are numerical features} \end{cases} \qquad \text{(Equation 33)}$$

where A_i^t is the set of formal attributes of the class specified in feature f_i^t; A_i^r is the set of formal attributes of the class specified in feature f_i^r; and f_i^{max}

and f_i^{min} are the maximum and minimum numeric values of feature f_i respectively.

The AHP method was used to determine the weights of all indices. For this purpose, four domain experts were involved in the development of the AHP pair-wise comparisons. Then the consistency ratio was used to evaluate the validity of the pair-wise comparisons assessments. The results of the AHP process is the weights are reported in [15].

6.3. Methodology for product service-systems based on semantic similarity

Here we propose a modification of the approach developed by Lin et al. [15] by incorporating a semantic similarity measure. The objective is to investigate the ability of finding relevant strategies with a minimum of detailed design information which is scarce at the conceptual stage.

Our approach reuses most of the CBR method proposed by Lin et al. [15], but replaces product features with a class of product based on a predefined ontology. For this purpose, a semantic similarity measure is introduced that is based on the comparison of classes in an ontology. The semantic similarity measure quantifies the differences between two classes of product. Ontologies describe a shared understanding about what objects mean in terms of the classes of objects, their taxonomy and also the properties of the objects in each class. Fig. 6-1 illustrates the proposed methodology for selecting service in PSS.

The taxonomy of the ontology is defined in terms of the *is-a* relation. Here, a class c_i is-a subclass of c_j if all members of c_i are also members of c_j. The underlying concept of the semantic similarity measure is that if two classes share exactly the same attributes then we say that these classes are the same. Likewise, the more common attributes that are shared by two classes the more similar they are. Based on a previous work [114], the semantic similarity is calculated using the overlap coefficient (Equation 21).

6.4. Ontology construction

A product ontology was developed based on the procedure described in Chapter 3. The resulting concept lattice is shown in Fig. 6-2. The product ontology extends the upper ontology defined in the ISO 15926 standard [23]. For this purpose, three classes were added as subclasses of *physical object*: substance, mixture, and device. The definitions in the SUMO upper ontology were used to describe these classes [115]. After editing the ontology and save in the OWL language, the resulting classes in the product ontology is shown in Fig. 6-3.

Using FAST, attributes were selected by investigating the process or processes in which the product participates or is involved. Each process was described in terms of the objects that are transformed and the objects that are produced by it. For example, the objects that are transformed during the operation of a copier are the data input by the user, electricity, and paper and the objects that are produced by the same process are the copied printed paper. Thus, the attributes of the copier become: *consumes data*; *consumes electricity*; *consumes paper*; and *produces printed paper*.

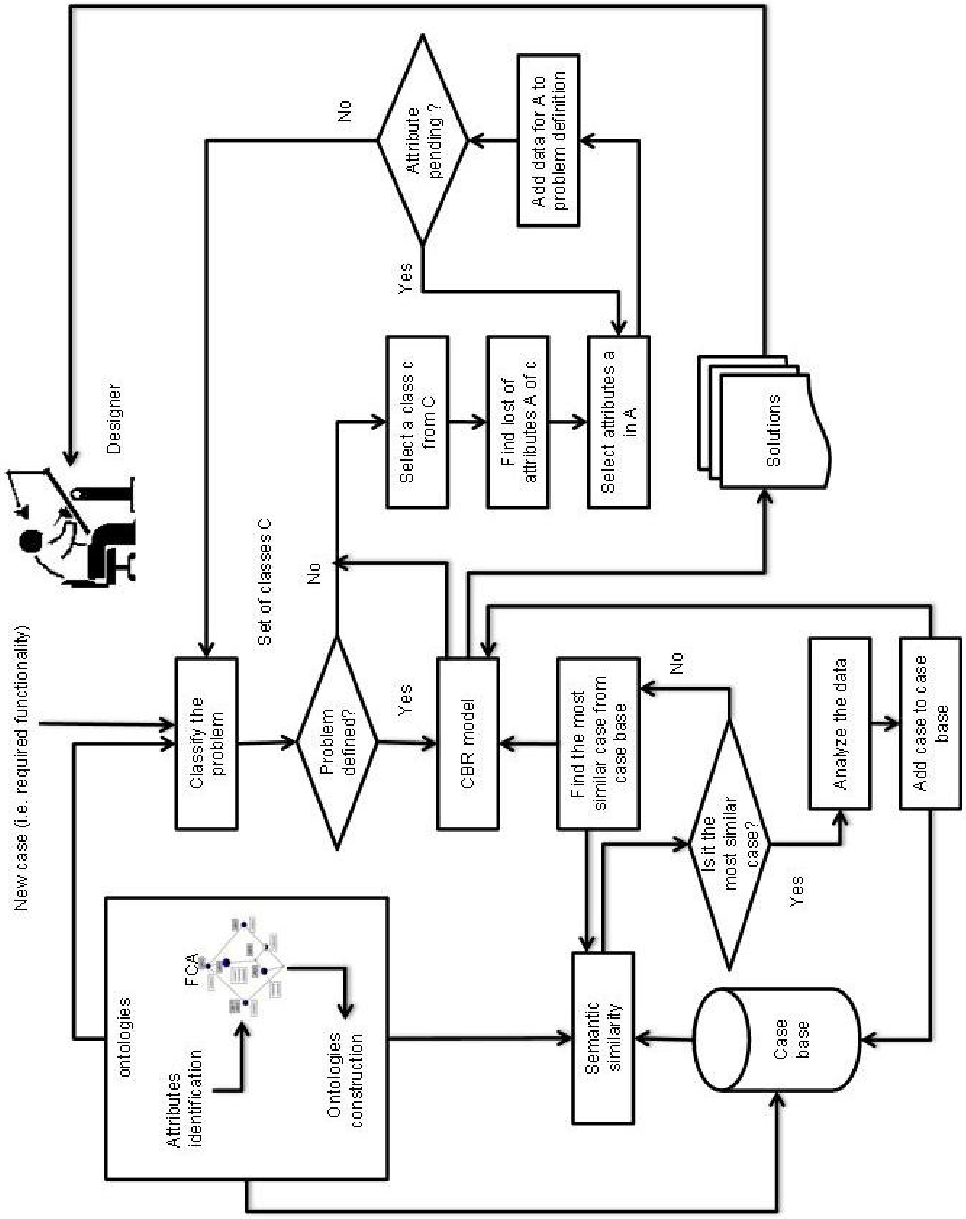


Fig. 6-1 Methodology for PSS

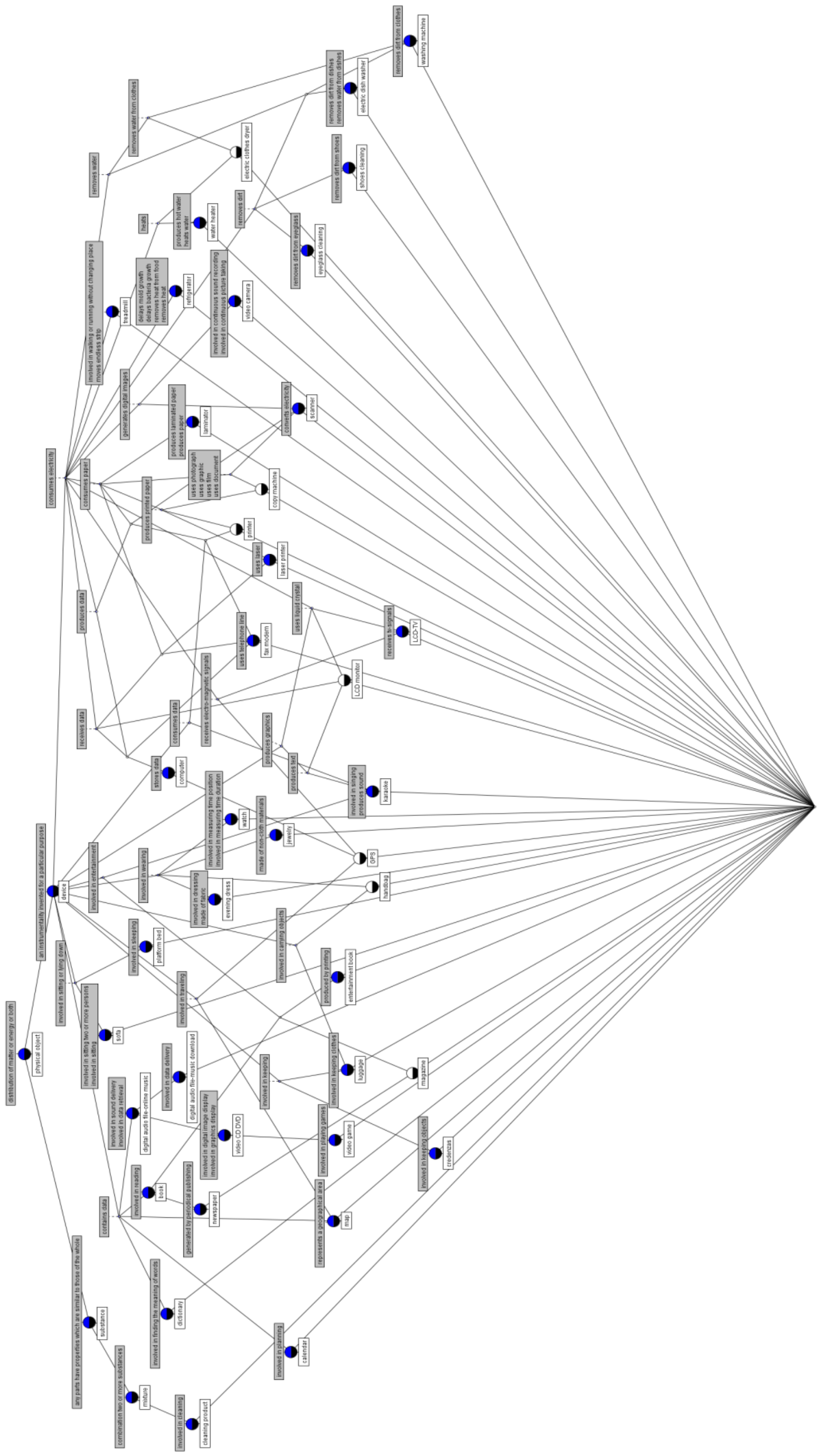


Fig. 6-2 The concept lattice for PSS

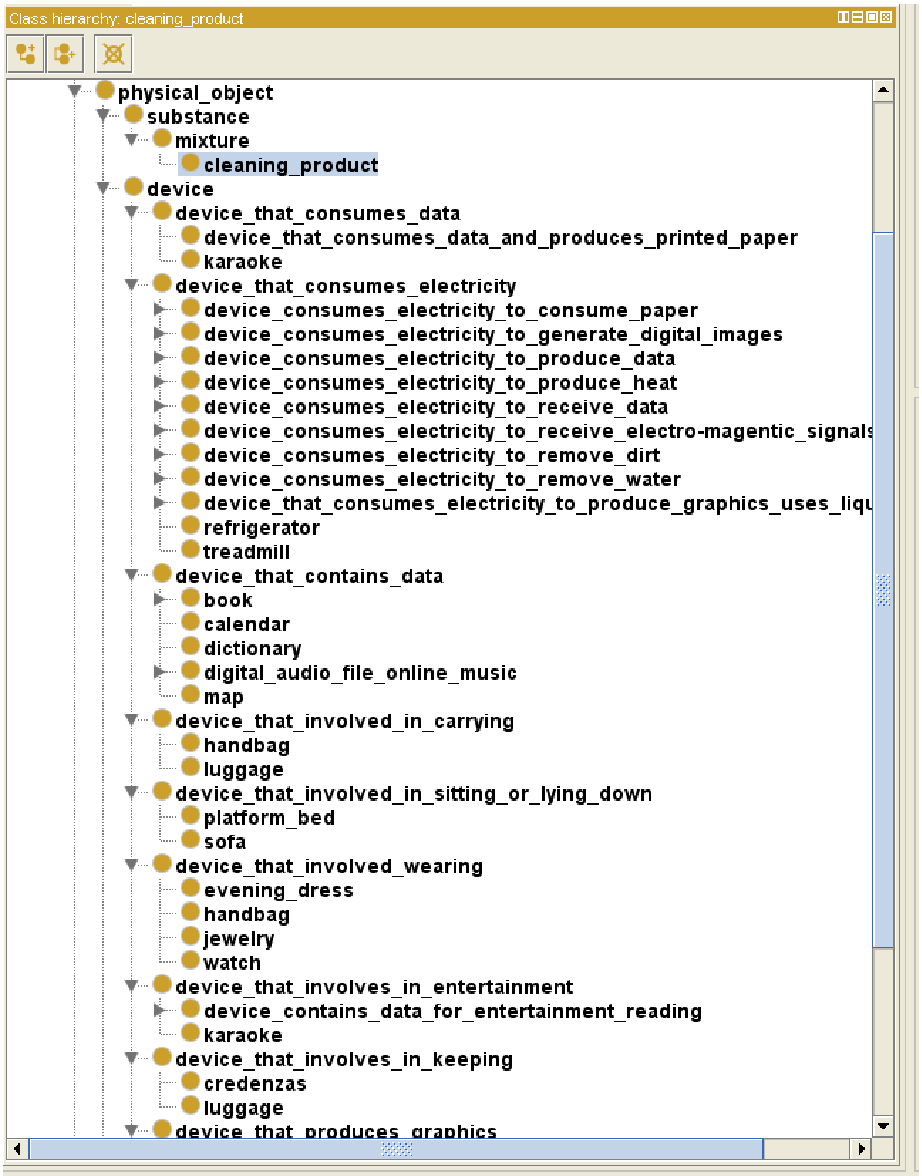


Fig. 6-3 Class hierarchy for product ontology

6.5.Case study

The purpose of this case study is to justify the effectiveness of semantic similarity measure of the proposed approach in CBR system. Two experiments were conducted using publicly available software. The case base was populated with information from 47 successful product services systems.

Each case was described in terms of numerical and/or semantic features. Based on [15], the following numeric features and weights were used: place of usage of the PSS system ($w_1 = 0.116$), frequency of usage of the PSS system ($w_2 = 0.232$), product fashion cycle ($w_3 = 0.042$), product volume ($w_4 = 0.036$), product weight ($w_5 = 0.034$), product useful life ($w_6 = 0.064$), product price ($w_7 = 0.082$), subsequent expenditure ($w_8 = 0.085$), GDP per capita ($w_9 = 0.119$), population density ($w_{10} = 0.079$), area of territory ($w_{11} = 0.052$), and temperature range of the territory ($w_{12} = 0.059$). The allowable values for each numeric feature and their meaning is also explained in [15]. For example, the index used to describe the place of usage of the PSS system is defined for integer values ranging from 1 to 3, where 1 represents indoor, 3 outdoor and 2 both. Among these features, product fashion cycle, volume, weight, useful life, and price are product features. The list of successful PSS cases and case description are found in Appendix H. The objective of this experiment is to evaluate the possibility of using a semantic feature instead of some of the product attributes. The semantic feature consisted of the class of product defined in the product ontology.

As in the original approach, each case is defined in terms of a problem and a solution. The problem part is defined in terms of characteristics of a given product. On the other hand, the solution part of the case provides a suggested service strategy.

A CBR system was developed in Java by extending the open source software FreeCBR. As in any traditional CBR system, each case is defined in terms of a problem and a solution. In this case study, the problem is defined in terms of case features that represent characteristics of a given product. The case features can be numeric or semantic. For numeric features, the index approach proposed by Lin et al. is used [15]. The semantic feature is specified as the class to which the product belongs, which is defined in a product ontology. The similarity for such semantic feature was calculated using equation 21 and the formal attributes of each class. The similarity for this semantic feature was calculated using the Java code described in Section 4.2.1. A screen dump of the CBR system is shown in F. 6-4.

Initially, two experiments were carried out. The objective of experiment 1 was to provide a reference for comparing the proposed approach. For this purpose, all the queries in experiment 1 consisted of values for all the numerical features.

In experiment 2, queries were formulated by replacing two product features (product volume and product weight) by the corresponding class of product from the ontology. The weight for this semantic feature was set to $w_{13} = 0.07$. It is the summation of weight of product volume and product weight which equals to 0.036 and 0.034, respectively.

The case similarity in both experiments were calculated with equations (32) and (33). The queries for both experiments were formulated with the product information from each of the cases stored in the case base. Therefore, 47 problems were defined with the problem data of the 47 cases in the case base, resulting in a total of 94 experiments. The objective was to find the service strategy and then compare the it with the already known service strategy of the corresponding case. For example, problem 1 describes a certain kind of washing machine that was used in PSS that provided a repair service. In this example, it is thus expected that all if not most of the n best matches return repair as the solution.

The execution of each query resulted in a ranked list of matches each of which included product information, the proposed service strategy, and a global similarity value. Then the resulting service strategies were compared against the original service.

Table 6-1 shows the results for both experiments. The best five matches are shown for each problem. From the overall results, it can be observed that there are 9 problems (Nos. 1, 5, 10, 14, 18, 26, 28, 43 and 45) in which the results of experiment 1 are identical with those of experiment 2. For example, the best five service strategies in problem 1 were: refrigerator-repair, computer-repair, water heater-repair, laser printer-repair and LCD monitor-repair, all of which are consistent to the repair service corresponding to the solution of problem 1.

Other problems produced slightly different results. For example, in experiment 2, problems 11, 12, 19, 20, 33, 34, 37, 38, 40, 41 and 42 produced the same five best matches found in the results of experiment 1 but with a different ranking. For example, in problem 11 both experiments resulted in treadmill-lease, dryer-lease, LCD TV-lease, refrigerator-lease and dish washer-lease. However, while treadmill-lease has the highest rank in experiment 1, it appears second in experiment 2.

In addition, there were 27 results (such as problems 2, 3, 4, 6, 7, 8, and 9) that differed in one or two cases. For example, the results for problem 2 include an Internet-based digital calendar which is false positive. On the other hand, some results of experiment 2 were good matches albeit being missing in experiment 1. For example, (sofa-lease and platform bed-lease instead of jewelry-rental and handbag rental) in problem 9 are good matches.

Furthermore, the results of experiment 2 for problems 30 (photocopy-service), 31 (scanning-service) and 32 (laminating-service) are better when compared to the results of experiment 1 in which not only the best 5 matches refer to a service that equals that of the case from which the query was formulated (pay per service unit) but also the product is more compatible with that of the suggested service. For example, experiment 2 for problem 30 resulted in laundry-service, printing-service, eyeglass cleaning-service, scanning-service and fax-service. Among these, printing, scanning and fax can be carried out with a copier machine. These results contrast with those obtained with experiment 1 which included cleaning-service, eyeglass cleaning-service and shoes cleaning service.

Table 6-1 The results for experiments of product-service system.

Problem	Case (product and known solution)		Best five matches			
	Product	Service	Experiment 1 (numeric features only)		Experiment 2 (class feature replaces weight and volume features)	
			Product-service	Similarity	Product-service	Similarity
1	Washing machine	repair	refrigerator repair	94	refrigerator repair	90
			computer repair	91.29	computer repair	89.82
			water heater repair	89.92	water heater repair	88.45
			laser printer repair	89.85	laser printer repair	87.88
			LCD monitor repair	88.11	LCD monitor repair	87.27
2	Refrigerator	repair	water heater repair	95.92	water heater repair	94.45
			washing machine repair	94	computer repair	91.56
			computer repair	93.02	LCD monitor repair	90.43
			LCD monitor repair	91.26	washing machine repair	90
			laser printer repair	83.85	digital calendar	82.12
3	Computer	repair	LCD monitor repair	95.394	LCD monitor repair	94.19
			refrigerator repair	93.02	refrigerator repair	91.56
			water heater repair	93.01	washing machine repair	89.82
			washing machine repair	91.29	water heater repair	89.51
			laser printer repair	85.21	digital calendar	84.97
4	Laser printer	repair	washing machine repair	89.85	washing machine repair	87.88
			printing service	88.4	printing service	87.23
			computer repair	85.21	computer repair	82.87
			refrigerator repair	83.85	refrigerator repair	81.88
			water heater repair	83.83	online karaoke	80.63
5	LCD monitor	repair	computer repair	95.39	computer repair	94.19
			water heater repair	95.34	water heater repair	92.98
			refrigerator repair	91.26	refrigerator repair	90.43
			washing machine repair	88.11	washing machine repair	87.27
			digital calendar	86.16	digital calendar	85.59
6	Water heater	repair	refrigerator repair	95.92	refrigerator repair	94.45
			LCD monitor repair	95.34	LCD monitor repair	92.98
			computer repair	93.01	computer repair	89.51
			washing machine repair	89.92	washing machine repair	88.45
			laser printer repair	83.83	digital calendar	84.17
7	Handbag	repair	jewelry repair	95	jewelry repair	94.15
			watch repair	93.18	watch repair	92.33
			audio book	78.20	treadmill lease	78.39
			LCD TV lease	77.63	LCD TV lease	77.3
			treadmill lease	77.59	handbag rental	76.57
8	Jewelry	repair	handbag repair	95	handbag repair	94.15
			watch repair	94.08	watch repair	92.33
			jewelry rental	80.67	jewelry rental	80.67
			handbag rental	79.77	handbag rental	78.92
			download audio book	75.00	treadmill lease	74.29
9	Watch	repair	jewelry repair	94.08	jewelry repair	92.33
			handbag repair	93.18	handbag repair	92.33
			refrigerator lease	75.09	refrigerator lease	76.99
			jewelry rental	74.75	sofa lease	75.86

			handbag rental	73.85	platform bed lease	74.25
10	Treadmill	Lease	washing machine lease	94.85	washing machine lease	92.95
			LCD TV lease	90.04	LCD TV lease	88.37
			dryer lease	89.68	dryer lease	87.78
			dish washer lease	82.06	dish washer lease	79.26
			refrigerator lease	81.12	refrigerator lease	79.22
11	Washing machine	lease	treadmill lease	94.85	dryer lease	93.67
			dryer lease	94.83	treadmill lease	92.95
			LCD TV lease	93.39	LCD TV lease	91.42
			refrigerator lease	86.27	dish washer lease	84.31
			dish washer lease	85.41	refrigerator lease	82.27
12	LCD TV	lease	washing machine lease	93.39	washing machine lease	91.42
			treadmill lease	90.04	treadmill lease	88.37
			refrigerator lease	88.81	refrigerator lease	86.84
			dish washer lease	88.33	dryer lease	86.76
			dryer lease	88.22	dish washer lease	85.46
13	Sofa	lease	platform bed lease	91.8	platform bed lease	90.05
			credenzas lease	85.29	credenzas lease	82.69
			refrigerator lease	85.20	refrigerator lease	81.89
			treadmill lease	79.075	download audio book	77.43
			dish washer lease	78.02	treadmill lease	76.68
14	Dryer	lease	washing machine lease	94.83	washing machine lease	93.67
			treadmill lease	89.68	treadmill lease	87.78
			LCD TV lease	88.22	LCD TV lease	86.76
			dish washer lease	82.84	dish washer lease	81.41
			refrigerator lease	81.1	refrigerator lease	77.6
15	Platform bed	lease	credenzas lease	93.49	credenzas lease	90.89
			sofa lease	91.8	sofa lease	90.05
			dish washer lease	86.22	dish washer lease	84.52
			refrigerator lease	82.89	music download	84.33
			music download	80.83	download audio book	83.03
16	Refrigerator	lease	LCD TV lease	88.81	LCD TV lease	86.84
			washing machine lease	86.27	washing machine lease	82.27
			sofa lease	85.19	sofa lease	81.89
			platform bed lease	82.89	platform bed lease	80.29
			credenzas lease	82.45	treadmill lease	79.22
17	Credenzas	lease	platform bed lease	93.67	platform bed lease	90.89
			sofa lease	85.42	music download	82.83
			refrigerator lease	82.42	sofa lease	82.69
			dish washer lease	81.5	download audio book	81.53
			music download	80.83	refrigerator lease	78.95
18	Dish washer	lease	LCD TV lease	88.33	LCD TV lease	85.46
			platform bed lease	86.22	platform bed lease	84.52
			washing machine lease	85.41	washing machine lease	84.31
			dryer lease	82.84	dryer lease	81.41
			treadmill lease	82.06	treadmill lease	79.26
19	Luggage box	rental	GPS rental	94.85	GPS rental	91.55
			scanning service	87.5	scanning service	84.2
			cleaning service	84.95	cleaning service	80.83
			video camera rental	83.67	eyeglass cleaning service	79.95
			eyeglass cleaning service	83.25	video camera rental	79.47
20	Video	rental	entertainment book	94.71	multimedia on demand	92.27

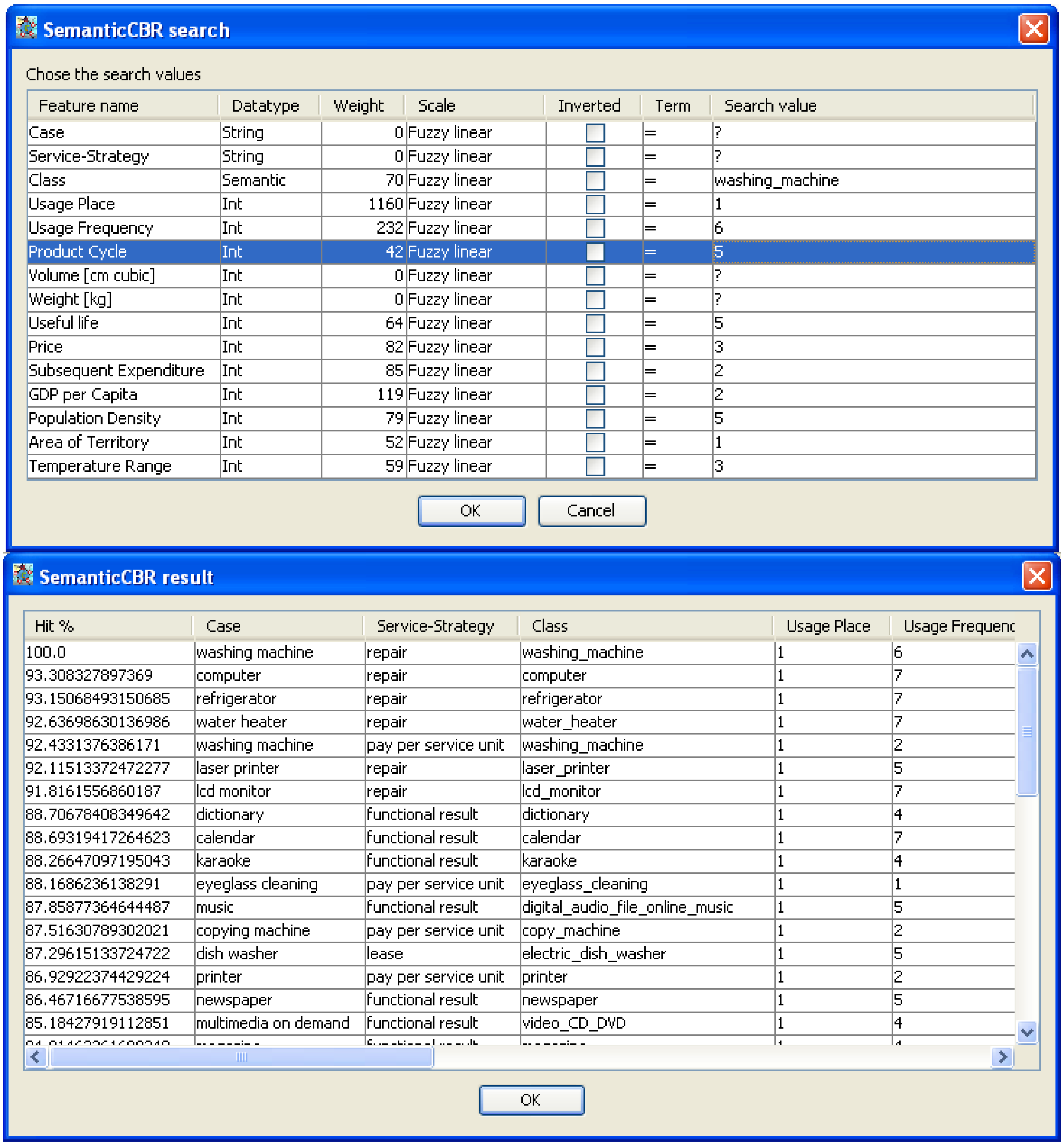
	CD/DVD		rental			
			fax service	92.63	entertainment book rental	91.21
			multimedia on demand	92.27	fax service	90.8
			online magazine	90.84	online music	88.4
			online music	88.4	online magazine	87.34
21	Evening dress	rental	handbag rental	85.38	handbag rental	85.66
			jewelry rental	84.48	jewelry rental	85.66
			video game rental	73.73	video game rental	72.47
			photographer service	70.28	handbag repair	66.33
			video camera rental	70.28	jewelry repair	66.33
22	Entertainme-nt book	rental	video CD/DVD rental	94.71	video CD/DVD rental	91.21
			scanning service	90.29	scanning service	87.66
			online magazine	88.4	online magazine	87.23
			fax service	87.34	fax service	85.84
			multimedia on demand	86.98	eyeglass cleaning service	83.88
23	Video game	rental	entertainment book rental	78.18	video CD/DVD rental	75.73
			jewelry rental	76.00	entertainment book rental	74.68
			video CD/DVD rental	75.73	handbag rental	72.50
			handbag rental	75.10	jewelry rental	72.50
			audio book	73.77	evening dress rental	72.47
24	Jewelry	rental	handbag rental	99.1	handbag rental	98.25
			evening dress rental	84.48	evening dress rental	85.66
			jewelry repair	80.67	jewelry repair	80.67
			video game rental	76.00	handbag repair	74.82
			handbag repair	75.67	watch repair	73
25	Handbag	rental	jewelry rental	99.1	jewelry rental	98.25
			evening dress rental	85.38	evening dress rental	85.66
			jewelry repair	79.77	jewelry repair	78.92
			handbag repair	76.57	handbag repair	76.57
			video game rental	75.10	watch repair	73
26	GPS	rental	luggage rental	94.85	luggage rental	91.55
			eyeglass cleaning service	88.4	eyeglass cleaning service	85.6
			cleaning service	88.3	cleaning service	85.08
			video camera rental	87.02	video camera rental	84.42
			laminating service	84.15	laminating service	80.65
27	DV(video camera)	rental	photographer service	95.9	photographer service	95.9
			cleaning service	88.72	eyeglass cleaning service	85.12
			eyeglass cleaning service	87.02	cleaning service	84.6
			GPS rental	87.02	GPS rental	84.42
			luggage rental	83.67	laundry service	82.6
28	Fax modem	pay per service unit	scanning service	92.95	scanning service	92.08
			video CD/DVD rental	92.63	video CD/DVD rental	90.8
			online dictionary	89.1	online dictionary	88.77
			laundry service	88.72	laundry service	87.65
			eyeglass cleaning service	88.7	eyeglass cleaning service	87.03
29	Printer	pay per service unit	laundry service	93.72	copying service	92.53
			laminating service	92.05	laundry service	92.25
			copying service	91.67	laminating service	91.75
			laser printer repair	88.4	laser printer repair	87.23

			cleaning service	87.9	eyeglass cleaning service	87.03
30	Photostat	pay per service unit	laundry service	97.95	laundry service	93.95
			printing service	91.67	printing service	92.53
			cleaning service	88.07	eyeglass cleaning service	89.23
			eyeglass cleaning service	87.97	scanning service	87.01
			shoes cleaning service	87.26	fax service	86.6
31	Scanning	pay per service unit	eyeglass cleaning service	95.75	eyeglass cleaning service	92.95
			fax service	92.95	fax service	92.08
			laminating service	91.5	laminating service	89.17
			entertainment book rental	90.29	entertainment book rental	87.66
			video CD/DVD rental	87.85	copying service	87.01
32	Laminating	pay per service unit	eyeglass cleaning service	95.75	eyeglass cleaning service	92.95
			printing service	92.05	printing service	91.75
			scanning service	91.5	scanning service	89.17
			laundry service	85.77	laundry service	86.33
			fax service	84.45	copying service	85.45
33	Washing machine	pay per service unit	copying service	97.95	copying service	93.95
			printing service	93.72	eyeglass cleaning service	92.68
			cleaning service	90.12	printing service	92.25
			eyeglass cleaning service	90.02	fax service	87.65
			fax service	88.72	cleaning service	86.9
34	Cleaning product	pay per service unit	laundry service	90.12	laundry service	86.9
			video camera rental	88.72	eyeglass cleaning service	85.08
			eyeglass cleaning service	88.3	GPS rental	85.08
			GPS rental	88.3	copying service	84.85
			copying service	88.07	video camera rental	84.6
35	Shoes cleaning	pay per service unit	copying service	87.26	eyeglass cleaning service	91.53
			laundry service	85.21	copying service	90.36
			eyeglass cleaning service	82.96	laundry service	89.71
			printing service	78.925	laminating service	85.88
			laminating service	78.71	scanning service	85.88
36	Eyeglass cleaning	pay per service unit	laminating service	95.75	laminating service	92.95
			scanning service	95.75	scanning service	92.95
			laundry service	90.02	laundry service	92.68
			fax service	88.7	copying service	89.23
			GPS rental	88.4	printing service	87.03
37	DV(video camera)	pay per service unit	video camera rental	95.9	video camera rental	95.9
			cleaning service	84.62	eyeglass cleaning service	81.02
			eyeglass cleaning service	82.92	copying service	80.55
			GPS rental	82.92	cleaning service	80.5
			copying service	80.88	GPS rental	80.32
38	Music CD (online music)	functional result	online newspaper	97.16	multimedia on demand	96.13
			multimedia on demand	96.13	online newspaper	94.36
			online magazine	94.71	online magazine	91.91
			online dictionary	91.93	online dictionary	90.18

			video CD/DVD rental	88.4	video CD/DVD rental	88.4
39	Magazine	functional result	multimedia on demand	98.58	multimedia on demand	95.08
			online newspaper	94.71	online newspaper	94.71
			online music	94.71	online dictionary	92.63
			online dictionary	94.38	online music	91.91
			video CD/DVD rental	90.84	online karaoke	88.78
40	Karaoke	functional result	multimedia on demand	91.67	online magazine	88.78
			online magazine	90.24	multimedia on demand	88.7
			online music	87.8	online dictionary	86
			online dictionary	87.47	online music	85.63
			online newspaper	84.96	online newspaper	82.79
41	Music CD (music download)	functional result	download audio book	98.7	download audio book	96.95
			platform bed lease	80.83	platform bed lease	84.33
			credenzas lease	80.23	credenzas lease	82.83
			dish washer lease	78.38	dish washer lease	78.92
			online music	78.03	online music	78.03
42	Video CD/DVD(multimedia on demand)	functional result	online magazine	98.58	online music	96.13
			online music	96.13	online magazine	95.08
			online dictionary	95.8	online dictionary	94.05
			online newspaper	93.29	video CD/DVD rental	92.27
			video CD/DVD rental	92.27	online newspaper	90.49
43	MAP	functional result	online magazine	85.4	online magazine	83.5
			online dictionary	83.98	online dictionary	83.13
			multimedia on demand	83.98	multimedia on demand	82.08
			luggage rental	80.79	luggage rental	78.63
			online newspaper	80.11	online newspaper	78.21
44	Newspaper	functional result	online music	97.16	online magazine	83.5
			online magazine	94.71	online dictionary	83.13
			multimedia on demand	93.29	multimedia on demand	82.08
			digital calendar	89.12	luggage rental	78.63
			online dictionary	89.09	online newspaper	78.21
45	Dictionary	functional result	multimedia on demand	95.8	multimedia on demand	94.05
			online magazine	94.38	online magazine	92.63
			online music	91.93	online music	90.18
			fax service	89.1	fax service	88.77
			online newspaper	89.09	online newspaper	87.3
46	Calendar	functional result	online newspaper	89.12	online newspaper	87.37
			online music	86.27	LCD monitor repair	85.59
			LCD monitor repair	86.16	computer repair	84.97
			online dictionary	84.51	online music	84.52
			computer repair	84.40	water heater repair	84.17
47	Book	functional result	music download	98.7	music download	96.95
			platform bed lease	79.53	platform bed lease	83.03
			credenzas lease	78.93	credenzas lease	81.53
			handbag repair	78.20	dish washer lease	78.78
			dish washer lease	77.08	sofa lease	77.43

In order to evaluate the ranking performance of the proposed semantic similarity in capturing the semantic similarity between services, an additional experiment was conducted (experiment 3). Experiment 3 excluded the numeric product features of volume and weight as well as the semantic feature.

In order to corroborate the influence of the semantic similarity, another evaluation was conducted in which we counted the cases in the best five


SemanticCBR search

Chose the search values

Feature name	Datatype	Weight	Scale	Inverted	Term	Search value
Case	String	0	Fuzzy linear		=	?
Service-Strategy	String	0	Fuzzy linear		=	?
Class	Semantic	70	Fuzzy linear		=	washing_machine
Usage Place	Int	1160	Fuzzy linear		=	1
Usage Frequency	Int	232	Fuzzy linear		=	6
Product Cycle	Int	42	Fuzzy linear		=	5
Volume [cm cubic]	Int	0	Fuzzy linear		=	?
Weight [kg]	Int	0	Fuzzy linear		=	?
Useful life	Int	64	Fuzzy linear		=	5
Price	Int	82	Fuzzy linear		=	3
Subsequent Expenditure	Int	85	Fuzzy linear		=	2
GDP per Capita	Int	119	Fuzzy linear		=	2
Population Density	Int	79	Fuzzy linear		=	5
Area of Territory	Int	52	Fuzzy linear		=	1
Temperature Range	Int	59	Fuzzy linear		=	3

OK Cancel

SemanticCBR result

Hit %	Case	Service-Strategy	Class	Usage Place	Usage Frequenc
100.0	washing machine	repair	washing_machine	1	6
93.308327897369	computer	repair	computer	1	7
93.15068493150685	refrigerator	repair	refrigerator	1	7
92.63698630136986	water heater	repair	water_heater	1	7
92.4331376386171	washing machine	pay per service unit	washing_machine	1	2
92.11513372472277	laser printer	repair	laser_printer	1	5
91.8161556860187	lcd monitor	repair	lcd_monitor	1	7
88.70678408349642	dictionary	functional result	dictionary	1	4
88.69319417264623	calendar	functional result	calendar	1	7
88.26647097195043	karaoke	functional result	karaoke	1	4
88.1686236138291	eyeglass cleaning	pay per service unit	eyeglass_cleaning	1	1
87.85877364644487	music	functional result	digital_audio_file_online_music	1	5
87.51630789302021	copying machine	pay per service unit	copy_machine	1	2
87.29615133724722	dish washer	lease	electric_dish_washer	1	5
86.92922374429224	printer	pay per service unit	printer	1	2
86.46716677538595	newspaper	functional result	newspaper	1	5
85.18427919112851	multimedia on demand	functional result	video_CD_DVD	1	4

OK

Fig. 6-4 Screen dump of the user interface of the semantic CBR system.

results that were common to those in experiment 1. The results are summarized in Table 6-2. In other words, the ideal number of common cases is 5. The presence of the semantic similarity measure in experiment 2 resulted in an average of 4.32 common cases, while its absence in experiment 3 resulted in an average of 3.77. This means that in the absence of data for product volume and weight, the use of the semantic feature shows an improvement of almost 14% compared to not using it. From this, it can be concluded that ontology-based semantic similarities have the ability to emulate (at least to some extent) the numeric product features.

Table 6-2 Comparison of identical cases

Problem	Number of best cases that are identical with Experiment 1	
	Experiment 2 (using a class feature instead of volume and weight)	Experiment 3 (only numeric features but volume and weight are excluded)
1	5	5
2	4	5
3	4	4
4	4	4
5	5	5
6	4	4
7	4	2
8	4	4
9	3	3
10	5	5
11	5	5
12	5	5
13	4	4
14	5	5
15	4	4
16	4	5
17	4	4
18	5	4
19	5	3
20	5	4
21	3	4
22	4	3
23	4	3
24	4	3

25	4	3
26	5	3
27	4	2
28	5	4
29	4	4
30	3	3
31	4	5
32	4	4
33	5	4
34	5	1
35	4	4
36	3	3
37	5	2
38	5	4
39	4	4
40	5	5
41	5	4
42	5	4
43	5	0
44	3	5
45	5	4
46	4	4
47	4	4
Average	4.32	3.77

Precision and recall graphs were calculated for the three experiments for all 47 queries. For this purpose, the evaluation focuses on the capability of the proposed method to retrieve services that match to the already known solution to problem query.

Precision is the ratio of the number of relevant services retrieved to the total number of irrelevant and relevant services retrieved (Equation 34), while recall corresponds to the ratio of the number of relevant services retrieved to the total number of relevant services in the case base (Equation 35).

$$precision_{n-best}(x) = \frac{A}{A+C} \qquad \text{(Equation 34)}$$

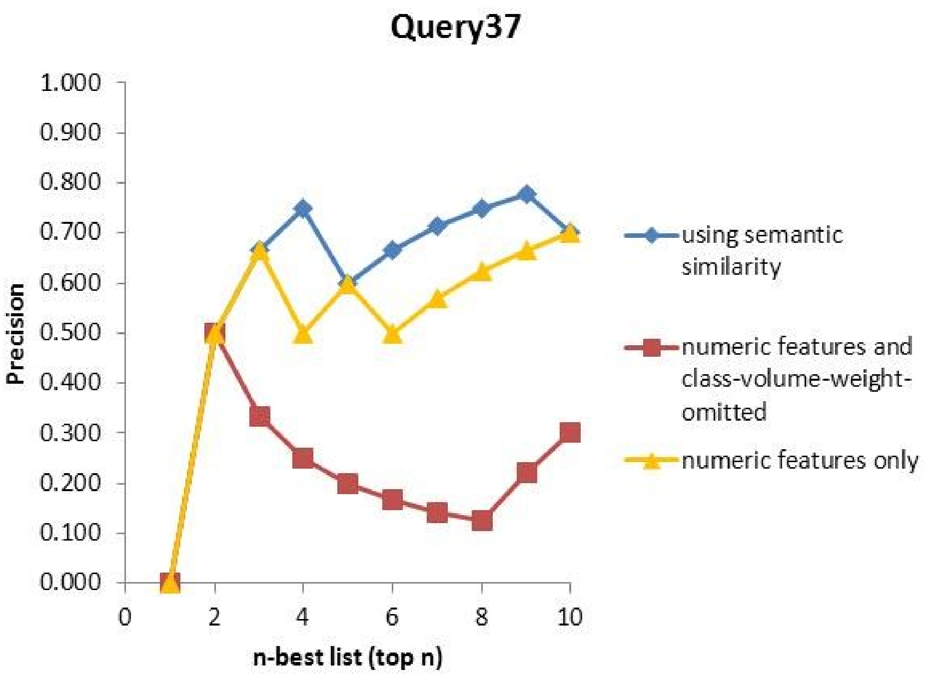
$$recall_{n-best}(x) = \frac{A}{A+B} \qquad \text{(Equation 35)}$$

where A is the number of relevant services retrieved, B is the number of relevant services not retrieved and C is the number of irrelevant services retrieved.

The precision-recall graphs for all 47 problems can be found in Appendix I. The x-axis represents the proportion of the ranking list, while the y-axis depicts the corresponding precision or recall value. No significant improvement can be observed in the results of experiment 2 (using semantic features) against the other two experiments. However, some of the results are favorable to the proposed approach such as problems No. 37 and 43.

Fig. 6-5 and Fig. 6-6 visualize the precision-recall graphs for problem No. 37 and 43, respectively. The n-best evaluations show that, the experiment of using semantic features performs significantly better than the other methods. Problem No. 37 is the query information about a video camera and the most successful type of service that couples with the video camera is pay per service unit. Another example is the results of problem No. 43 in which the product is map and the matches solution to the problem is functional service. In the absence of the semantic features, no relevant results could be obtained.

Table 6-3 summarizes the average precision for both experiment 2 and 3. Average precision and recall is calculated by averaging the precision and recall value with the total number of problem which is 47. As we can see from Table 6-2, for example at the best 5 solutions, the value for precision and recall of the proposed approach presented very small improvement of 5.5% - 5.80% and with respect to that of experiment 3 (numeric features but volume and weight are excluded).



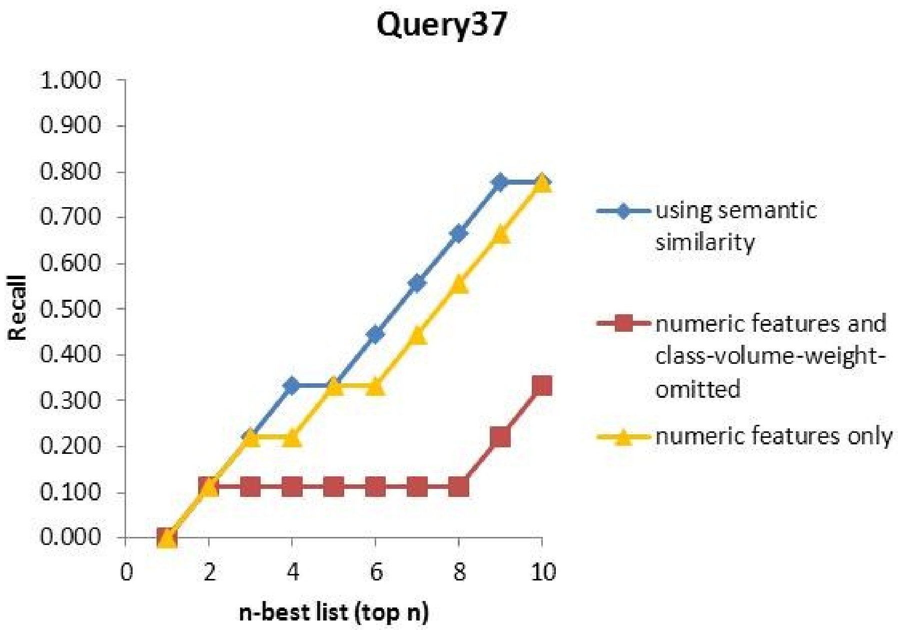
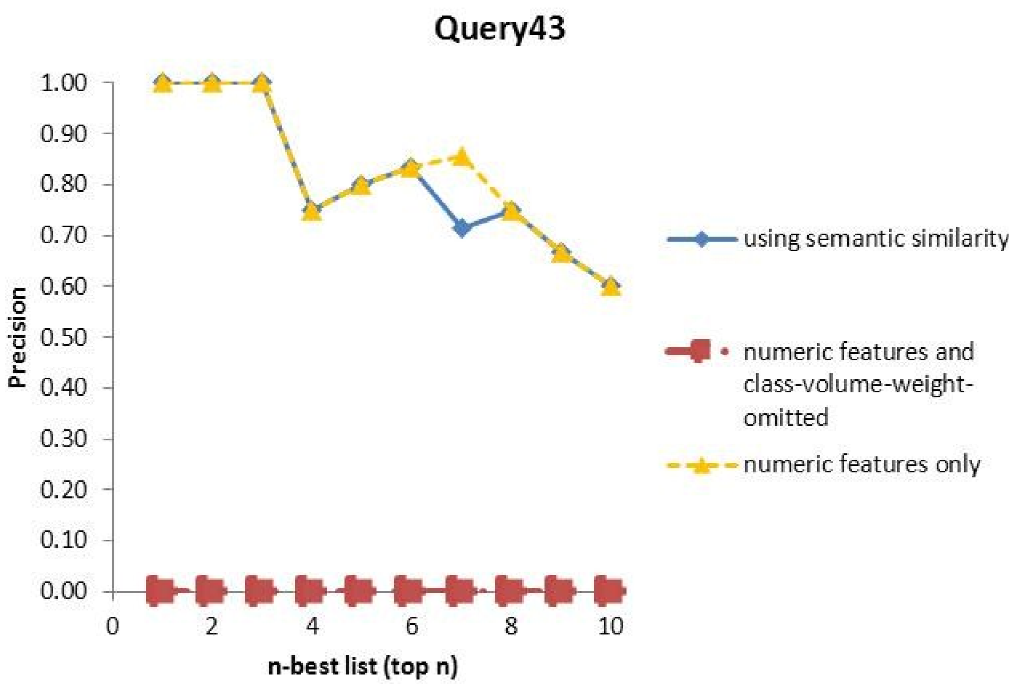


Fig. 6-5 Precision (upper part) – recall (lower part) graphs for problem No. 37



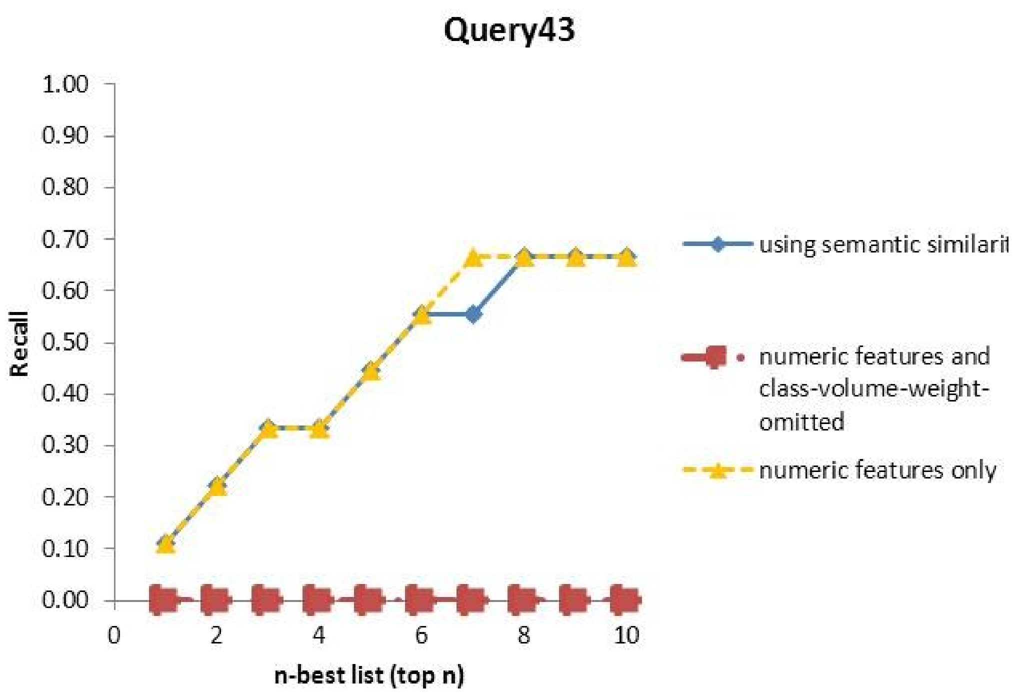


Fig. 6-6 Precision (upper part) – recall (lower part) graphs for problem No. 43

Table 6-3 Comparison of average precision and recall

	Average precision		Average recall	
n-best	With class of product ontology	Without class of product ontology	With class of product ontology	Without class of product ontology
1	0.936	0.872	0.112	0.104
2	0.862	0.830	0.205	0.198
3	0.801	0.752	0.286	0.269
4	0.750	0.713	0.357	0.340
5	0.711	0.672	0.422	0.400
6	0.681	0.649	0.484	0.462
7	0.642	0.611	0.540	0.508
8	0.595	0.566	0.577	0.539
9	0.558	0.539	0.619	0.576
10	0.499	0.509	0.626	0.604

6.6. Conclusions

This chapter presented semantic similarity measure of the proposed approach for determining the service selection during the design of product service systems. The foundation of the proposed approach is a product ontology that is developed in a systematic way using formal concept analysis and a selection of attributes based on the description of the objects transformed and produced by the processes associated to a given product. Another aspect of the proposed approach is the use of a CBR system that uses a combination of numeric and semantic similarity measures to determine the service strategy for a PSS.

A CBR system for Product Service Systems demonstrated the effectiveness of the proposed similarity measures. The semantic similarity measure is based on the attribute information determined during the formal concept analysis stage of the ontology construction. In the CBR case study, the combination of the ontology and the semantic similarity proved useful when some details such as weight and volume are not available. Therefore, the designer can be relieved by needing less data to define a given design problem, which is particularly important during the conceptual stage of the design.

Nevertheless, in a few instances the proposed approach resulted in mismatches. This could be due to the lack of attributes in the FCA context table. For example, the addition of attributes that emphasize the difference between software and hardware products could reduce the number of false positives for problem 2.

The results of the experiments demonstrated that the semantic similarity measure of the proposed approach can replace some of the quantitative information of the product that may not be available at the conceptual stage of the design.

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1. Major contributions

This thesis comes in a time when research on semantic similarity is growing at high pace (Fig. 7-1). In this thesis, the semantic similarity method for comparing classes of products and processes in a unified framework has been introduced and explained. Also, this research supports the reuse, share and exchange of the heterogeneous information and knowledge by providing a meaningful representation of products and processes. The foundations of the proposed approach are: (1) a semantic similarity measure based on Formal Concept Analysis (FCA) and, (2) Formal Attribute Specification Template (FAST).

The main distinguishing of the proposed approach is the formal attributes information. Most of the semantic similarity measures are limited to shortest path length. However, this thesis presents how to use the attribute information effectively in measuring semantic similarity of classes of products and processes. Furthermore, experiments in Chapter 4 also demonstrate the ability of the semantic similarity measures to compute the taxonomy that contains multiple-inheritance classes. In general, the results of evaluation against human judgment and NGD similarity, the semantic similarity measures of the proposed approach performed better than existing similarity measures. The correlation coefficient of proposed approach presented an improvement of 1.7 – 8.9% with respect to that of human judgment, while an improvement of 9.28-13.50% with respect to NGD similarity.

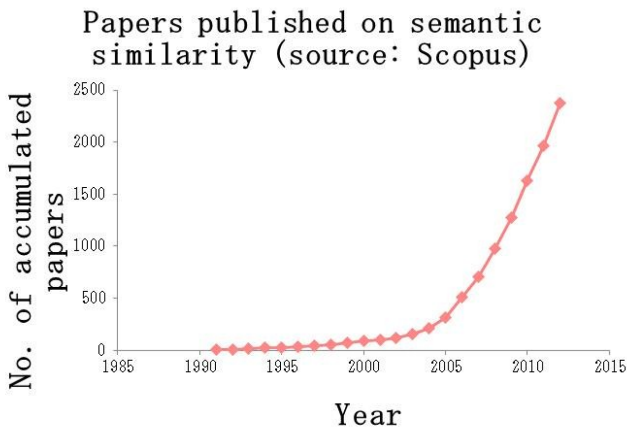


Fig. 7-1 The trend of papers published on semantic similarity

Another important element in this research is the use of FAST that allows for systematic identification of formal attributes common to all members of classes of products and processes that distinguish them from members of another class. In Chapter 5, FAST was used for the systematic construction of machining processes ontology. The results of evaluation against NGD similarity depict that the formal attribute information serves the design rationale and justification of the ontology. Also, the proposed approach can be used to develop the ontology and it helps to evaluate and improved the ontology. Therefore, the methodology has the following unique characteristics:

- FCA provides a degree of flexibility in developing ontologies. The proposed approach can be used to update the ontology as it evolves.

- FAST described in Chapter 3 provides a mechanism to ensure the accuracy and consistency of knowledge acquisition and conceptualization.

One of important application of the proposed approach is determining the service selection during the early stage of design of product service systems as presented in Chapter 6. In this chapter, we present the developed product ontology using FAST. The developed CBR-system is used determine the service strategy for PSS by incorporating product-class-comparison based on the proposed semantic similarities. Through the numerical experiment, we have shown that the proposed approach is promising for discovering knowledge associated to product and process when some details of information are not available.

7.2. Future work

The work presented in this thesis demonstrates the high potential of semantic similarity method in products and processes. This research clearly contributes to strategies that exploit the capability of information retrieval to achieve competitive advantage. However, the proposed approach needs more refininig. There are several areas in which this research can be expanded.

Mechanisms for the automatic identification of potential classes and their characteristics can extend this research. An interesting work in that direction is the approach by Poshyvanyk and Marcus [116] in which automatic formal context generation is part of a scheme to locate information in source code.

Due to the rapid growth of information sources on the web, there is a need for developing semantic similarity methods which would compute among concepts belonging to different class hierarchies [117]. Therefore, there is a potential to make use of the semantic similarity measure of the proposed approach towards ontology alignment with a reference ontology to improve the mapping process between two classes of different ontologies to different classes in mediating ontology. An interesting work is proposed by [48] using semantic similarity measure with reference ontology.

References

[1] R. L. Goldstone, "Similarity, interactive activation and mapping," *Journal of Experimental Pschology: Learning, Memory and cognition,* vol. 20, no. 1, pp. 3-28, 1994.

[2] W. V. Quine, "Ontological Relativity and Other Essays. in Holt, A., Understanding environmental and geographical compleities through similarity matching," Complexity Inetrnational, 2000.

[3] A. Holt, "Understanding environmental and geogrpahical complexities through similarity matching," Complexity International, 2000.

[4] M. Quillian, "Semantic Memory in Saruladha, K., Semantic similarity measures for information retrieval systems using ontology," MIT Press, 1968.

[5] A. Tversky and I. Gati, "Studies of similarities," in *Cognition and Categorization*, Hillsdale, NJ, Erlbaum, 1978, pp. 79-98.

[6] S. Russel and P. Norvig, "Artificial intelligence: A modern approach, in Saruladha, K.," 2003.

[7] M. M. Richter, "Similarity," in *Case-Based Reasoning for Signals and Imaging*, Springer Verlag, 2007, pp. 25-90.

[8] R. M. Felder and R. W. Rousseau, Elementary Principles of Chemical Processes, New York, U.S.A.: John Wiley & Sons, Inc., 2000.

[9] B. Bayer, "Conceptual information modeling for computer aided support of chemical process design," VDI Verlag GmbH, Duesseldorf, 2003.

[10] T. Tomiyama and B. R. Meijer, "Direction of next generation product development," in *Advances in design*, Springer, 2006, p. 31.

[11] M. L. Maher, "Process models for design synthesis," AI Magazine Volume 11(4), 1990.

[12] S. Szykman, J. Racz, C. Bochenek and R. D. Sriram, "Web-based system for design artifact modeling," *Design Studies,* vol. 21, no. 2, pp. 145-165, 2000.

[13] S. K. Ong, A. Y. Nee and Q. L. Xu, "Design reuse for embodiment and detail design," in *Design Reuse in Product Development Modeling*, World Scientific Pub. Co Inc., 2009, pp. 237-247.

[14] J. J. Shah, D. K. Jeon, S. D. Urban, P. Bliznakov and M. Rogers, "Database Infrstructure for Supporting Engineering Design Histories," *Computer-Aided Design,* vol. 28, no. 5, pp. 347-360, 1996.

[15] K. Lin, L. Shih, S. Lu and Y. Lin, "Strategy selection for product service systems using case-based reasoning," *African Journal of Business Management, Vol. 4(6),* pp. 987-994, 2010.

[16] P. Cunningham, "A taxonomy similarity mechanisms for case-based reasoning," UCD-CSI, 2008.

[17] A. Maedche and S. Staab, "Comparing Ontologies— Similarity Measures and a Comparison Study," Institute AIFB, University of Karlsruhe, Karlsruhe, Germany, 2001.

[18] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of 15th International Conference on Machine Learning*, Madison, Wisconsin, 1998.

[19] V. Cross and X. Hu, "Using semantic similarity in ontology alignment," in *Sixth International Workshop on Ontology Matching*, 2011.

[20] V. Olehschuk and A. Pedersen, "Ontology based semantic similarity for documents," in *14th International Workshop on Database and Expert System Application*, Prague, 2003.

[21] A. Bramantoro, S. Krishnaswamy and M. Indrawan, "A semantic distance measure for matching web services," in *LCNS 3807, WISE 2005 Workshops*, New York, 2005.

[22] J. S. Madin, "Advancing ecological research with ontologies," *Trends in Ecology &*

Evolution,Vol 23(3), pp. 159-168, 2008.

[23] R. Batres, M. West, D. Leal, D. Price, K. Masaki, Y. Shimada, T. Fuchino and Y. Naka, "An Upper Ontology based on ISO 15926," *Computers And Chemical Engineering,* vol. 31, p. 519–534, 2007.

[24] A. Budanitsky and G. Hirst, "Evaluating WordNet-based Measures of Lexical Semantic Relatedness," *Computational Linguistics,* vol. 32, p. 13–47, 2005.

[25] H. Al-Mubaid and H. A. Nguyen, "A cluster-based approach for semantic similarity in the biomedical domain," in *Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '06)*, New York, 2006.

[26] J. J. Jiang and D. W. Conrath, "(1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy.," in *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, Taiwan, 1997.

[27] U. Priss, "Formal Concept Analysis in Information Science," *Annual Review of Information Science and Technology, Vol. 40,* pp. pp. 521-543, 2006.

[28] G. Fu and A. G. Cohn, "Utility Ontology Development with Formal Concept Analysis," 2008.

[29] G. Stumme, "Using ontologies and formal concept analysis for organizing business knowledge," 2002.

[30] P. Resnik, "Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research II,* vol. 11, no. 1, pp. 95-130, 1999.

[31] A. Budanitsky and G. Hirst, "Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In Turney," in *Pro- ceedings of theWorkshop onWordNet and Other Lexical Resources, Second Meet- ing of the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*, Pittsburgh, PA, 2001.

[32] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of 15th International Conference on Machine Learning*, Madison, Wisconsin, 1998.

[33] R. Rada, H. Mili, E. Bicknell and M. Blettner, "Development and Application of a Metric on Semantic Nets in Cross and Hu," *IEEE Transaction on Systems, Man, and Cybernetics,* vol. 19, pp. 17-30, 1989.

[34] C. Leacock and M. Chodorow, "Combining local context and WordNet Similarity for word sense identification in Cross and Hu," MIT Press, 1998.

[35] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *32nd Annual Meeting of the Associations for Computational Lingustics* , New Mexico, 1994.

[36] M. Batet, D. Sanchez and A. Valls, "An ontology-based measure to compute semantic similarity in biomedicine," *Journal of Biomedical Informatics,* vol. 44, pp. 118-125, 2011.

[37] D. Sánchez, M. Batet, D. Isern and A. Valls, "Ontology-based semantic similarity: A new feature-based approach.," *Expert Systems with Applications,* vol. 39, no. 9, p. 7718–7728, 2012.

[38] P. Resnik, "Using information content to evaluate semantic similarity in a taxanomy," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal, Quebec, Canada, 1995.

[39] Princeton University, "WordNet 3.1: A lexical database dor the english language," 2010. [Online]. Available: http://wordnet.princeton.edu/wordnet/.

[40] D. Sánchez, M. Batet, D. Isern and A. Valls, "Ontology-based semantic similarity: A new feature-based approach.," *Expert Systems with Applications,* vol. 39, no. 9, p. 7718–7728, 2012.

[41] N. Seco, T. Veale and J. Hayes, "in Seddiqui and Aono, Metric of Intrinsic Information Content for Measuring Semantic Similrity in an Ontology," in *the 16th European Conference on Artificial Intelligence* , 2004.

[42] M. Batet, D. Sánchez and A. Valls, "An ontology-based measure to compute semantic similarity in biomedicine," *Journal of Biomedical Informatics,* vol. 44, no. 1, p. 118–25, 2011.

[43] A. Tversky, "Features of similarity," *Psychological Review,* vol. 84, no. 4, pp. 327-352, 1977.

[44] E. G. M. Petrakis, G. Varelas, A. Hliaoutakis and P. Raftopoulou, "Design and Evaluation of Semantic Similarity Measures for Concepts Stemming from the Same or Different Ontologies object instrumentality.," *Journal of Digital Information Management,* vol. 4, pp. 233-237, 2006.

[45] M. A. Rodríguez and M. Egenhofer, "Determining semantic similarity among entity classes from different ontologies," *IEEE Transactions on Knowledge and Data Engineering,* vol. 15, pp. 442-456, 2003.

[46] M. H. Seddiqui and M. Aono, "An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size," *Web Semantics: Science, Services and Agetns on the World Wide Web,* vol. 7, pp. 344-356, 2009.

[47] H. Al-Mubaid and H. A. Nguyen, "A cluster-based approach for semantic similarity in the biomedical domain.," in *In Proceedings of the 28th Annual International Conference of the 28th IEEE Engineering in Medicine and Biology Society (EMBS '06)* , New York, USA, 2006.

[48] V. Cross, P. Silwal and D. Morell, "Using a reference ontology with semantic similarity for ontology alignment," in *7th International Conference on Formal Ontology in Information Systems*, Graz, Austria, 2012.

[49] N. Jyotirmaya, S. Timothy, S. Kumara and S. Shooter, "A Methodology for Product Family Ontology Development Using Formal Concept Analysis and Web Ontology Language," *Journal of computing and information science in engineering, vol. 6(2),* vol. 6, no. 2, pp. 103-113, 2006.

[50] B. Chandrasekaran, "The Functional Representation Language: A Framework for Reasoning About Functions and Casual Processes of Devices," 1993.

[51] G. J. Klir, Architecture of Systems Problem Solving, New York: Plenum Press, 1985.

[52] W. Marquadt, "An Object-Oriented Representation of Structured Process Models," Vols. 16, pp. S329-S336, 1992.

[53] R. Bogusch and W. Marquardt, "A formal representation of process model equations," vol. 21, no. 10,pp. 1105-1115, 1997.

[54] R. Knappe, T. Andreasen and H. Bulskov, "Domain-specific similarity and retrieval," in *11th International Fuzzy Systems Association World Congress, IFSA 2005*, Beijing, China, 2005.

[55] M. A. Rodríguez and M. Egenhofer, "Determining semantic similarity among entity classes from different ontologies," *IEEE Transactions on Knowledge and Data Engineering,* vol. 15, pp. 442-456, 2003.

[56] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology ,* vol. 26, no. 3, pp. 297-302, 1945.

[57] J. Hian , M. Kamber and J. Pei, "Getting to know your data," in *Data mining: concepts and techniques*, United States of America, Morgan Kaufmann, 2013, p. 71.

[58] J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2nd ed. Morgan Kaufmann Pub., an imprint of Elsevier, 2006.

[59] D. van der Weken, M. Nachtegael and E. E. Kerre, "Using similarity measures and homegeneity for the comparison of images," *Image and Vision Computing,* pp. 695-702, 2004.

[60] ISO-TC-184, "ISO-TC-184/SC4, SO 10303, "Industrial Automation Systems and Integration - Product Data Representation and Exchange- Part 11: Description Methods:

The EXPRESS Language Reference Manual," ISO TC 184/SC4, 1994.

[61] D. Leal, "ISO 15926 "Life Cycle Data for Process Plant": an Overview," *Oil & Gas Science and Technology, 60,* vol. 4, pp. 629-637, 2005.

[62] J. F. Sowa, Knowledge Representation: Logical,Philosophical and Computational Foundations, Brooks/Cole, 2000.

[63] A. Pease, I. Niles and J. Li , "The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications.," in *Proceedings of the AAAI Workshop on Ontologies and the SW*, 2002.

[64] NIST, "NIST IDEF0 Function Modeling Method, IDEF integrated DEFinition method," 1993. [Online]. Available: http://www.idef.com/IDEF0.htm.. [Accessed 1 December 2010].

[65] R. Batres, M. West, D. Leal, D. Price, K. Masaki, Y. Shimada, T. Fuchino and Y. Naka, "An upper ontology based on ISO 15926," *Computers and Chemical Engineering 31,* vol. 31, no. 519-534, pp. 519-534, 2007.

[66] R. Hai, M. Theissen and W. Marquadt, "An ontology based approach for operational process modeling," *Advanced Engineering Informatics,* pp. 748-759, 2011.

[67] M. Theissen, R. Hai and W. Marquadt, "A framework for work process modeling in the chemical industries," *Computers and chemical Engineering,* pp. 679-691, 2011.

[68] R. Hai, M. Theissen and W. Marquadt, "An Extensible Modeling Language for the Representation of Work Processes in the Chemical and Process Industries," in *Graph Transformations and Model-Driven Engineering*, Berlin-Heidelberg, Springer-Verlag, 2010, pp. 655-682.

[69] J. S. Gero and U. Kannengiesser, "A function–behavior–structure ontology of processes," vol. 21, no. 4, 2007.

[70] R. Stevens, C. A. Coble and S. Bechhofer, "Ontology-based knowledge representation for bioinformatics," *Briefings in bioinformatics,* vol. 1, no. 4, pp. 398-414, 2000.

[71] S. Yevtushenko, "Concept Explorer, Open source java software.," 2009. [Online]. Available: http://sourceforge.net/projects/conexp, Release 1.3.

[72] G. Stumme, "Exploration Tools in Formal Concept Analysis," 1995.

[73] B. Ganter, "Formal Concept Analysis: algorithmic aspects," TU Dresden, Dresden, 2002.

[74] Stanford Center for Biomedical Informatics Research., [Online]. Available: <http://protege.stanford.edu>.

[75] Web-Ontology working group, "OWL web ontology language (2nd Ed.)," <http://www.w3.org>, 2004.

[76] Jena, "HP Labs Semantic Web Toolkit.," [Online]. Available: <http://jena.sourceforge.net/>..

[77] National Academy of Sciences, "Unit manufacturing processes: Issues and opportunities in research," 1995. [Online]. Available: http://www.nap.edu/catalog.php?record_id=4827. [Accessed 18 December 2011].

[78] D. Baxter, R. Roy, A. Doultsinou, J. Gao and M. Kalta, "A knowledge management framework to support product-service systems," *International Journal of Computer Integrated Manufacturing, Volume 22, Issue 12,* pp. 1073-1088, 2009.

[79] d. S. Lazaro and D. T. Engquist, "An Intelligent Design for Manufacturability System for Sheet-Metal Parts," vol. 1, no. 117-123, 1993.

[80] A. Soman, S. Padhye and I. Matthew, "Toward an automated approach to the design of sheet metal components," vol. 17, no. 3, 187-204, 2003.

[81] H. Yang, W. F. Lu and A. C. Lin, "PROCASE: a case-based process planning system for machining of rotational parts," *Journal of Intelligent Manufacturing,* vol. 5, no. 6, 411-430, pp. 411-430, 1994.

[82] N. F. Noy and D. McGuinness, "Ontology Development 101: A Guide to Creating your

First Ontology," Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, Standford USA, 2001.

[83] D. Djurić, D. Gaševic and V. Devedžić, "Ontology Modeling and MDA," vol. Vol. 4, no. No. 1, pp. 109-128., 2005.

[84] P. Cimiano, A. Hotho, G. Stumme and J. Tane, "Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies," Sydney, Australia, 2004.

[85] M. Vegetti, G. Henning and H. Leone, "Product Ontotlogy. Definition of an Ontology for The Complex Product Modelling Domain," in *Proceedings of the 2nd Mercosur Congress on Chemical Engineering*, Rio de Jeneiro, Brazil, 2005.

[86] S. Lemaignan, A. Siadat, J.-Y. Dantan and A. Semenenko, "MASON: A Prposal For an Ontology of Manufacturing Domain," in *Proceedings of te IEEE Workshop on DIstributed Intelligent SYstems: Collective Intelligence and Its Applications*, Prague, 2006.

[87] S. Borgo and P. Leitao, "The role of Foundational Ontologies in Manufacturing Domain Applications," in *In R. Meersman, Z. Tari et al. (eds.). OTM Confederated International COnferences, ODBASE 2004*, Cyprus, 2004.

[88] M. Grueninger and D. Arnaud, "A First-order Cutting Process Ontology for Sheet Metal Parts," in *Proceedings of the 2009 conference on Formal Ontologies Meet Industry*, Vicenza, Italy, 2009.

[89] M. Uschold and M. King, "Towards a methodology for building ontologies," in *Workshop on basic ontological issues in knowledge sharing*, 1995.

[90] M. Grueninger and M. S. Fox, "Methodology for the Design and Evaluation of Ontologies," in *workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, 1995.

[91] M. Fernandez-Lopez, A. Gomez-Perez and N. Juristo, "Methontology: From ontological art towards ontological engineering," AAAI Technical Report SS-97-06, 1997.

[92] A. Gomez-Perez, "Evaluation of Taxanomic Knowledge in Ontologies and Knowledge Bases," in *Twelfth Workshop on Knowledge Acquisition, Modelling, and Management (KAW '99)*, Alberta, Canada, 1999.

[93] R. E. Schafrik, "Unit Manufacturing and Assembly Process, in: J. Lee," in *Modern Manufacturing, Mechanical Engineering Handbook*, CRC Press LLC, 1999, pp. 13-8 - 13-34.

[94] S. Kalpakjian and S. R. Schmid, Manufacturing Engineering and Technology, Sixth Edition, Prentice-Hall, 2010.

[95] B. S. Nagendra Parashar and R. K. Mittal, Elements of manufacturing processes, Prentice-Hall of India Private Limited, 2007.

[96] E. P. Degarmo, J. T. Black and R. A. Kohser, Materials and Processes in Manufacturing (10th ed.), Wiley, 2010.

[97] S. Yevtushenko, "Concept Explorer, Open source java software.," 2009. [Online]. Available: http://sourceforge.net/projects/conexp, Release 1.3.

[98] B. Handley, D. M. Marshall and C. Coon, Principles of Engineering, Delmar, 2012.

[99] Anoynamous, "Blasting Technical Information," [Online]. Available: http://mcfinishing.com/resources/blastingtech.pdf.

[100] Stanford Center for Biomedical Informatics Research., [Online]. Available: <http://protege.stanford.edu>.

[101] J. P. David, C. Maranhao, P. Faria, A. Abrao, J. C. Rubio and L. R. Silvia, "Precision radial turning of AISI D2 steel," *Int. J. Adv. Manuf. Technol., 42,* pp. 842-849, 2009.

[102] R. L. Cilibrasi and P. M. B. Vitanyi, " The Google Similarity Distance," *IEEE Transactions On Knowledge and Data Engineering,* vol. 19, pp. 370-383, 2007.

[103] D. Giustini, "How big is Google Scholar? 500 million Documents?," [Online]. Available: http://blogs.ubc.ca/googlescholar/2006/04/how-big-is-google-scholar-500-million-documents/.

[104] N. Maussang, T. Sakao, P. Zwolinski and D. Brissaud, "A model for designing product-service systems using functional analysis and agent based model," in *International conference onengineering design, ICED '07*, Paris, France, 2007.

[105] T. S. Baines, H. W. Lightfoot, S. Evans, A. Neely, R. Greenough, J. Peppard, R. Roy, E. Shehab, A. Braganza, A. Tiwari, J. R. Alcock, J. P. Angus, M. Bastl, A. Cousens, P. Irving, M. Johnson, J. Kingston, H. Lockett, V. Martinez, P. Michele, D. Tranfield, I. M. Walton and H. Wilson, "State-of-the-art in product-service systems," in *Proceedings of the Institution of Mechanical Engineers -- Part B -- Engineering Manufacture, Vol. 221, No. 10.*, 2007.

[106] N. Morelli, "Developig new product service system (PSS): methodologies and operational tools," Vols. 14, 1495-1501, 2006.

[107] N. Boughnim and B. Yannou, "Using blueprinting mehod for developing prodcut-service systems," in *International Conference on Engineering design*, Melbourne, 2005.

[108] T. Sakao, Y. Shimomura, M. Comstock and E. Sundin, "Service engineering for value customization," in *3rd International Conference on Engineering Desig*, Melbourne, 2005.

[109] H.-E. Tseng, C.-C. Chang and S.-H. Chang, "Applying Case-based Reasoning for Product Configuration in Mass Customization Environments," *Expert Systems with Applications,* pp. 913-925, 2005.

[110] M. Ogot and G. Okudan-Kremer, "Problem definition and determination of need," in *Engineering Design: A Practical Guide*, Togo Press, LLC, 2004, pp. 145-161.

[111] M. C. Wu, Y. F. Lo and S. H. Hsu, "A Case-Based Reasoning Apporach to Generating New Product Ideas," *Int. J. Adv. Manuf Technology,* vol. 30, pp. 166-173, 2006.

[112] S. Perera and I. Watson, "Nirmani: An integrated Case-Based system for strategic design and estimating," in *Progress in case-base reasoning Lecture notes in Computer Science*, 1995, pp. 185-200.

[113] J. Kolodner and R. L. Simpson, "Case-based reasoning in Lin, K. H., Shih, L. H., Lu, S. S., and Lin, Y. T., 2010, Strategy Selection for Product Service Systems Using Case-based Reasoning," *African Journal of Business Management Vol. 4(6),* 1993.

[114] S. Akmal and R. Batres, "Ontology-based Semantic Similarity Measures for Product-Service System Design," in *International Symposium on System Integration*, Kyoto, Japan, 2011.

[115] A. Pease, "Suggested Upper Merged Ontology (SUMO)," [Online]. Available: <http://www.ontologyportal.org/>. [Accessed 2010 December 2010].

[116] D. Poshyvanyk and A. Marcus, "Combining Formal Concept Analysis with Information Retrieval for Concept Location in Source Code.," in *15th IEEE International Conference on Program Comprehension (ICPC '07) (pp. 37–48).* , 2007.

[117] Y. Zhao and W. Halang, "Rough concept lattice based ontology similarity measure.," in *Proceedings of the 1st international conference on Scalable information systems - InfoScale '06*, 2006.

[118] W. R., "Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts," in *Ordered Sets*, Dordrecht-Boston, U.S.A., Springer Netherlands, 1982, p. 445–470.

[119] B. A. Davey and H. A. Priestly, Introduction to Lattices and Orders, Cambridge University Press, 1990.

[120] M. Fernando-Lopez, "Overview of Methodologies for Building Ontologies," in *IJCAI-99 workshop ontologies and problem-solving methods (KRR5)*, Stockholm, Sweden, 1999.

[121] D. L. Marca and C. L. McGowan, "IDEF0 and SADT: A modeler's guide. In Batres et. al, An Upper Ontology based on ISO 15926," 2005.

[122] M. Missikoff and R. Navigli, "Applying the unified process to large-scale ontology building. In Batres et. al, An Upper Ontology based on ISO 15926," in *In Proceedings of 16th IFAC world congress (IFAC).*, 2005.

[123] M. Uschold and M. Gruninger, "Ontologies: Principles. Methods and Applications Engineering Review," Vols. 11(2), pp. 93–113, 1996.

[124] A. Gomez-Perez, M. Fernandez-Lopez and A. J. de Vicente, "Towards a Method to Conceptualize Domain Ontologies," Working notes of the workshop Ontological Engineering, ECAI'96, 1996.

Appendix A: Formal Concept Analysis

Formal Concept Analysis (FCA) can be used to design taxonomy from a list of potential classes and their respective attributes. FCA is an analysis technique for information processing based on applied lattice and order theory that can be used to generate taxonomies [118], is selected as the lattice generator.

A lattice is a partially ordered set with a least upper bound (also known as supremum) and a greatest lower bound (also known as infimum) [119]. In this research, the nodes in the lattice represent classes and the edges represent subclass relations.

In order to construct the lattice, FCA requires information to be organized in a so-called *formal context*. A formal context is defined as a set $K := \langle O, A, Y \rangle$, where O is a set whose elements are called formal objects, A is a set whose elements are called formal attributes, and Y is an incidence relation. The relation Y is defined for all pairs $\langle o, a \rangle \in Y$ such that formal object o has formal attribute a as in (*bicycle*, *has wheels*).

Table A-1 A context table

	att_1	att_2	att_3	att_4	att_5	att_6
ob_1	×					
ob_2		×				
ob_3	×		×	×		
ob_4			×			×
ob_5			×	×	×	

Formal contexts can be represented by a cross table, such as the one shown in Table A-1 or as an incidence matrix. In either case, the formal objects are listed in the rows and the formal attributes in the columns of the table. If a formal object has an attribute, which means that there is a binary relation between them, a checkmark is inserted in that cell. Alternatively, a formal context can be represented by an incidence matrix, by replacing the checkmarks with 1s and empty cells with 0s. In the proposed methodology, the potential classes are considered as formal objects, and the information obtained in the characterization step is considered as attributes.

A formal concept is defined as the pair $\langle O_i, A_i \rangle$ such that:

1. $O_i \subseteq O$, $A_i \subseteq A$,
2. Every object in O_i has every attribute in A_i. Conversely, A_i is the set of attributes shared by all the objects in O_i,
3. For every object in O that is not in O_i, there is an attribute in A_i that the object does not have,
4. For every attribute in A that is not in A_i, there is an object in O_i that does not have that attribute.

In other words, a formal concept $\langle O_i, A_i \rangle$ is obtained when

- $A' := \{a \in A \mid \langle o,a \rangle \in Y \, \forall o \in O_i\}$
- $O' := \{o \in O \mid \langle o,a \rangle \in Y \, \forall a \in A_i\}$
- $O_i \subseteq O$, $A_i \subseteq A$, $O' = A_i$, $A' = O_i$

where A' is the set of formal attributes common to all formal objects in O_i, and O' represents the set of formal objects that has all the attributes in A_i. O_i and A_i are respectively the extent and the intent of the formal concept. The formal concepts obtained from the context table, Table A-1, are shown in Table A-2.

Table A-2 Formal concepts form the context table, Table 2-1

ID	Formal Concept
c_1	$(\{ob_1, ob_2, ob_3, ob_4, ob_5\}, \{\emptyset\})$
c_2	$(\{ob_2\}, \{att_2\})$
c_3	$(\{ob_1, ob_3\}, \{att_1\})$
c_4	$(\{ob_3, ob_4, ob_5\}, \{att_3\})$
c_5	$(\{ob_3, ob_5\}, \{att_3, att_4\})$
c_6	$(\{ob_4\}, \{att_3, att_6\})$
c_7	$(\{ob_3\}, \{att_1, att_3, att_4\})$
c_8	$(\{ob_5\}, \{att_3, att_4, att_5\})$
c_9	$(\{\emptyset\}, \{att_1, att_2, att_3, att_4, att_5, att_6\})$

Formal concepts can be partially ordered into a lattice, such that a concept is a subconcept of another concept: $\langle O_i, A_i \rangle \leq \langle O_j, A_j \rangle$ iff $A_i \subseteq A_j$.

Several lattice-construction algorithms have been proposed. Lattices in this research were generated by means of the Grail algorithm [71], which is implemented in the software Concept Explorer. When the lattice has been obtained, it can be visualized and analyzed. In these algorithms, operations are applied to identify the concepts that can be obtained by the intersection of others.

In FCA, a lattice also serves as a visual aid that helps to explain the relations between the formal concepts. The lattice provides the transparency of the different meanings of concept lattice. For example, the lattice can be viewed as a hierarchical classification of the objects and representation of all attributes implications.

The lattice corresponding to the concepts of Table A-1 is shown in Fig. A-1. A circle labeled by an object (a filled circle in Fig. A-1) represents the concept with the smallest extent containing that object. Conversely, a circle labeled by an attribute (a small circle in Fig. A-1) represents the concept with the smallest intent containing that attribute.

From a concept lattice, the set of formal objects of a concept can be obtained by following all the paths that lead down from that concept. For example, the objects of c_3 in Fig. A-1 are {ob_1, ob_3}. Conversely, to obtain the set of formal attributes of a concept, we trace all the paths that lead up from that concept. For example, the formal attributes of c_7 are {att_1, att_3, att_4}.

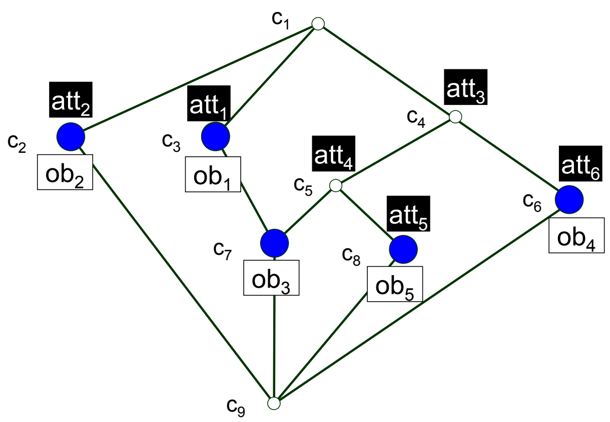


Fig. A-1 A concept lattice

As a result, an edge in the lattice means that a concept is a subconcept of another concept (superconcept—subconcept relation). The superconcept-subconcept relation is transitive. Consequently, if a node A is a subconcept of B, and B is also a subconcept of C, A is a subconcept of C. This means that a subconcept inherits all the attributes from all its superconcepts.

The top (supremum) and bottom (infimum) concepts have a particular meaning. The top concept includes all the formal objects of the nodes below. The bottom concept has all the formal attributes of the nodes above.

The original formal context is not guaranteed to be complete. Therefore, approaches are needed for improving the lattice. One of such approaches is the so-called object exploration. According to Stumme [72] object exploration is a “structured brainstorming” that consists of suggesting implications to the lattice-designer and then evaluating the validity of each implication. If a given implication is found to be incorrect, the lattice-designer determines the attributes that are needed in order to distinguish the conflicting objects. This approach assumes that all objects of the context are given, but the set of attributes is incomplete.

Appendix B: Draw a concept lattice by hand

The primitive method to draw the concept lattice diagram is by using pencil and paper. The following is the method on how to draw the concept lattice diagram. In this example, the concept lattice of Fig- A-1 and the corresponding context table of Table A-1 (as shown below) were used to illustrate the method.

	att_1	att_2	att_3	att_4	att_5	att_6
ob_1	×					
ob_2		×				
ob_3	×		×	×		
ob_4			×			×
ob_5			×	×	×	

Step 1 Determine all formal concepts of a small formal context; $K := \langle O, A, Y \rangle$

a. Write for each attributes m ∈ M the attribute extent {m}′ to a list of attribute extents

No.		extent	Found as
e_1	:=	$\{ ob_1, ob_3 \}$	$\{ att_1 \}'$
e_2	:=	$\{ ob_2 \}$	$\{ att_2 \}'$
e_3	:=	$\{ ob_3, ob_4, ob_5 \}$	$\{ att_3 \}'$
e_4	:=	$\{ ob_3, ob_5 \}$	$\{ att_4 \}'$
e_5	:=	$\{ ob_5 \}$	$\{ att_5 \}'$
e_6	:=	$\{ ob_4 \}$	$\{ att_6 \}'$

b. Identify all pairwise intersections for any two sets and listed them in the list of attribute extents. If necessary, extend the table to include a set that is not yet defined.

No.		extent	Found as
e_1	:=	$\{ ob_1, ob_3 \}$	$\{ att_1 \}'$
e_2	:=	$\{ ob_2 \}$	$\{ att_2 \}'$
e_3	:=	$\{ ob_3, ob_4, ob_5 \}$	$\{ att_3 \}'$
e_4	:=	$\{ ob_3, ob_5 \}$	$\{ att_4 \}'$
e_5	:=	$\{ ob_5 \}$	$\{ att_5 \}'$
e_6	:=	$\{ ob_4 \}$	$\{ att_6 \}'$
e_7	:=	$\{ ob_3 \}$	$e_1 \cap e_3 \cap e_4$
e_8	:=	$\emptyset$	$\{ att_1, att_2, att_3, att_4, att_5, att_6 \}'$

c. Extend the list by the set O

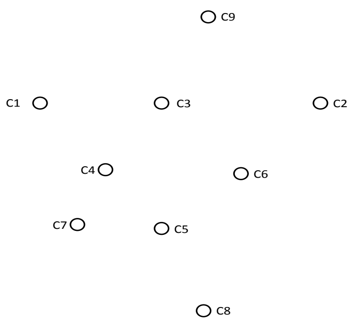
No.		extent	Found as
e_1	:=	{ ob_1, ob_3 }	{ att_1 }'
e_2	:=	{ ob_2 }	{ att_2 }'
e_3	:=	{ ob_3, ob_4, ob_5 }	{ att_3 }'
e_4	:=	{ ob_3, ob_5 }	{ att_4 }'
e_5	:=	{ ob_5 }	{ att_5 }'
e_6	:=	{ ob_4 }	{ att_6 }'
e_7	:=	{ ob_3 }	$e_1 \cap e_3 \cap e_4$
e_8	:=	$\emptyset$	{ att_1 , att_2 , att_3 , att_4 , att_5 , att_6 }'
e_9	:=	{ ob_1, ob_2, ob_3, ob_4, ob_5 }	$\emptyset$

d. Determine the intent A' for every concept A in the list to obtain s list of all formal concepts (A, A') of $\langle O, A, Y \rangle$.

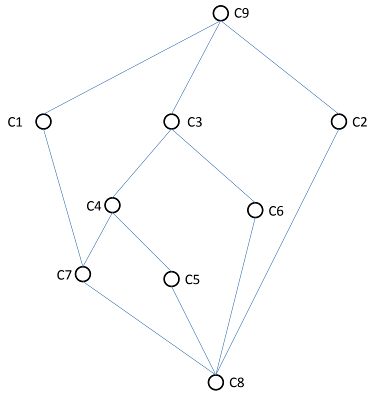
Concept No.	extent	,	intent
1	({ ob_1, ob_3 }	,	{ att_1 })
2	({ ob_2 }	,	{ att_2 })
3	({ ob_3, ob_4, ob_5 }	,	{ att_3 })
4	({ ob_3, ob_5 }	,	{ att_3 , att_4 })
5	({ ob_5 }	,	{ att_3 , att_4 , att_5 })
6	({ ob_4 }	,	{ att_3 , att_6 })
7	({ ob_3 }	,	{ att_1 , att_3 , att_4 })
8	($\emptyset$	,	{ att_1 , att_2 , att_3 , att_4 , att_5 , att_6 })
9	({ ob_1, ob_2, ob_3, ob_4, ob_5 }	,	$\emptyset$)

Step 1 Draw a line diagram of a small formal context; $K := \langle O, A, Y \rangle$

e. Draw a circle for each of the formal concepts. If $(A_1, B_1) \leq (A_2, B_2)$ and $(A_1, B_1) \neq (A_2, B_2)$ holds, then we say that (A_1, B_1) is a subconcept of (A_2, B_2). A circle for a concept is always positioned higher than the all circle for its proper subconcepts.



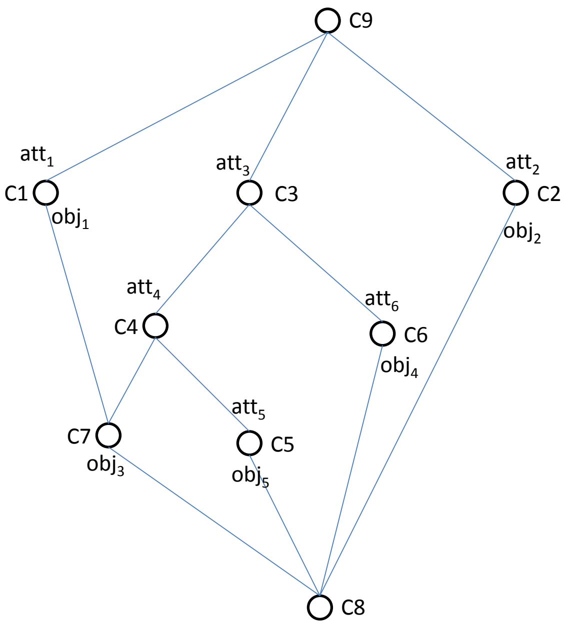
f. Write the attribute names



g. Determine object concepts

Object, o	object intent { o }'	No. of concept
ob_1	{ att_1 }	1
ob_2	{ att_2 }	2
ob_3	{ att_1 , att_3 , att_4 })	7
ob_4	{ att_3 , att_6 })	6
ob_5	{ att_3 , att_4 , att_5 })	5

h. Write object names to diagram



Appendix C: Questionnaire

WHICH HOME APPLIANCE IS SIMILAR TO ELECTRIC KETTLE?

This survey is conducted to identify which home appliance as listed in Table 1 has "*similarity of function*" with **ELECTRIC KETTLE.** An electric kettle consumes electricity to generate heats. Subsequently, it **heats** water and produces hot water.

Please rank the following object pairing according to their "similarity of function". Please give a rank 1 – 17, where 1 for most similar and 17 for not match.

表１の項目で、似た機能を持つものから順に１-１７の順位付けをしてください。

Table C-1

Object pair	Your Rank
Electric kettle - room electric heater ポットー暖房	
Electric kettle - water heater ポットー給湯器	
Electric kettle - hair dryer ポットードライヤー	
Electric kettle - electric blanket ポットー電気毛布	
Electric kettle - washing machine ポットー洗濯機	
Electric kettle - electric clothes dryer ポットー乾燥機	
Electric kettle – refrigerator ポットー冷蔵庫	
Electric kettle - room air-conditioner ポットーエアコン	
Electric kettle - electric dish washer ポットー洗浄器	
Electric kettle - microwave ポットー電子レンジ	
Electric kettle - electric oven	

ポットーオーブン
Electric kettle – toaster ポットートースター
Electric kettle - television set ポットーテレビ
Electric kettle - conventional electric fan ポットー扇風機
Electric kettle – blender ポットーミキサー
Electric kettle - bread machine ポットーホームベーカリ
Electric kettle - vacuum cleaner ポットー掃除機

Table C-2 The respondent answer (part 1 of 3)

	Pair	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
Electric kettle	room_electric_heater	2	7	2	4	3	5	2	5	2	8
	water_heater	1	1	1	1	1	1	1	1	1	1
	hair_dryer	8	11	6	6	4	7	7	6	6	5
	electric_blanket	7	10	7	3	10	8	9	7	7	6
	washing_machine	16	16	10	13	15	13	12	11	16	15
	electric_clothes_dryer	6	9	8	9	16	9	5	8	3	2
	refrigerator	11	12	13	14	9	12	10	17	11	11
	room_air-conditioner	17	8	14	8	8	6	6	16	9	10
	electric_dish_washer	13	2	11	15	17	14	11	15	12	16
	microwave_oven	3	3	9	2	2	2	3	3	5	9
	electric_oven	4	6	4	5	5	3	4	2	4	4
	toaster	5	4	3	10	11	4	8	4	8	3
	television_set	14	17	17	17	13	17	14	13	17	17
	conventional_electric_fan	15	13	15	11	12	16	15	14	14	13
	blender	10	14	12	7	7	10	17	12	15	12
	bread_machine	12	5	5	12	6	11	13	9	10	7
	vacuum_cleaner	9	15	16	16	14	15	16	10	13	14
	2sigma					62.81					
	d_i	43	49	29	43	62	30	32	40	29	28

Table C-2 The respondent answer (part 2 of 3)

	Pair	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20
Electric kettle	room_electric_heater	2	3	4	2	5	2	7	2	1	8
	water_heater	1	1	1	1	1	1	1	1	5	1
	hair_dryer	3	2	5	10	6	6	8	6	2	7
	electric_blanket	5	4	7	7	7	5	7	5	4	12
	washing_machine	11	13	14	8	8	7	17	7	8	10
	electric_clothes_dryer	4	5	8	11	10	3	6	3	6	9
	refrigerator	15	14	10	12	11	17	5	15	9	11
	room_air-conditioner	10	9	15	13	12	12	7	4	12	13
	electric_dish_washer	14	12	16	9	14	10	12	16	10	14
	microwave_oven	9	6	3	6	2	5	4	11	16	2
	electric_oven	8	7	2	4	3	4	4	10	7	3
	toaster	7	5	6	3	4	2	5	8	3	4
	television_set	17	17	17	14	17	13	10	12	13	17
	conventional_electric_fan	12	10	12	15	13	12	8	17	17	15
	blender	16	15	11	16	16	9	6	13	14	6
	bread_machine	6	8	9	5	9	6	6	9	15	5
	vacuum_cleaner	13	16	13	17	15	8	9	14	11	16
	2sigma					62.81					
	d_i	31	22	26	36	25	134	134	46	134	49

Table C-2 The respondent answer (part 3 of 3)

	Pair	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30
Electric kettle	room_electric_heater	5	2	5	2	5	7	6	4	3	6
	water_heater	1	1	2	1	1	1	1	1	1	1
	hair_dryer	4	5	9	3	3	5	4	3	2	5
	electric_blanket	6	7	7	4	2	8	12	13	4	2
	washing_machine	15	12	11	12	14	13	11	12	11	13
	electric_clothes_dryer	8	6	6	11	9	10	7	5	8	4
	refrigerator	10	17	8	17	12	15	10	11	9	7
	room_air-conditioner	9	16	12	10	4	6	8	2	17	8
	electric_dish_washer	12	11	14	16	10	12	14	10	10	9
	microwave_oven	11	3	3	9	11	2	9	6	5	11
	electric_oven	2	4	1	5	7	3	3	7	7	10
	toaster	3	8	4	8	8	4	2	8	6	3
	television_set	16	15	16	15	17	17	13	17	14	17
	conventional_electric_fan	13	10	10	6	12	14	16	15	12	12
	blender	14	14	13	14	13	11	15	16	13	15
	bread_machine	7	9	17	7	6	9	17	9	16	14
	vacuum_cleaner	17	13	15	13	16	16	5	14	15	16
	2sigma					62.81					
	d_i	29	30	38	39	41	31	50	37	37	47

Appendix D: Existing methodologies for constructing ontologies

There are several works on the methodology for constructing ontology that are proposed for engineering domain. Uschold and King [89] propose a methodology that consists of four steps: (1) identify the scope and purpose of ontology; (2) construct the ontology by capturing knowledge through identification of the key concepts, coding knowledge for representing the concepts and relationships obtained in previous step in a formal language, and integrating knowledge with the existing ontologies; (3) evaluate the ontology; and (4) document the ontology. In order to identify concept in the ontology, they adopt a middle-out approach in which the most important concept are identified first and then, the concept is identified in abstract and finally specialized into other concepts.

Grüninger and Fox propose the use of competency questions to define requirements as an initial step in the ontology design process [90]. They define competency questions as questions that a knowledge-based system should be able to answer. Specifically, the competency questions help to identify the main classes and their attributes, relations and axioms on the ontology. Then, the specifications are formalized using first-order logic and defined with respect to the axioms in the ontology. By adding axioms to the ontology, it becomes possible to define the classes and relations in the ontology by constraining on their interpretation [90]. The final step is to add completeness theorems that define the conditions under which the solutions to the questions are complete. This methodology is developed in the domain of enterprise modeling in the framework of the TOVE project.

Noy and McGuinness' method [82] consists of seven steps for constructing the ontology. They start by defining the scope, domain and the user of the ontology. They suggest the use of competency questions for defining the scope of ontology. This methodology considers reusing the existing ontologies available on the web and then refining or extending the source. A list of relevant terminology is built to avoid concept and property redundancies. Then, a class hierarchy is developed. Finally they fill-in details of each class and relationships.

Fernández-López et al. [91] propose a framework called Methontology for developing the ontologies. The methodology is illustrated in the domain of chemical substances [92]. It starts by specifying why the ontologies are built, who the user is, and where the ontology will be used. Next, the knowledge of the specified domain is collected. Subsequently, in the phase of conceptualization, a glossary of possible terms in the given domain is built. Then, the conceptual knowledge is transformed into a formal-language representation. The result of this stage is the ontology codified in a formal language that can be verified and validated. In this methodology, a maintenance step is suggested to control and to rectify the changes in ontologies [120]. As life cycle based on evolving prototypes, it allows the ontology user to modify, add, and remove definitions in the ontology at any time. The final part consists of documentation.

Methodologies that have been described in the literature propose common stages such as specification, conceptualization, formalization, evaluation, and documentation. Requirements specification can be carried out by identifying key concepts by means of activity modeling [121], use cases [122] and competency questions [123]. This concepts are then defined based on the more general concepts provided by the upper ontology.

Conceptualization is the task of defining classes. Ontology evaluation is important in order to ensure that the built ontology meets the application requirement. Gómez-Pérez [124] introduces five criteria in evaluating an ontology: consistency, completeness, conciseness, expandability and sensitiveness. Grüninger and Fox [90] proposed the use of competency questions for ontology evaluation. Noy & McGuinness [82] evaluate the ontology by finding possible errors during the specification classes and class hierarchy.

Appendix E: Ontologies applications in products and processes

Several efforts are reported on the use of hierarchical structure in products and processes domain. In the area of product customization, Tseng et al. [50] present a CBR system to support conceptual product design. In their work, a numeric similarity measure is combined with part-whole information that has a tree representation. Another similar work is that of Cobb and Agogino [51] who developed a CBR system for designing Micro-Electro-Mechanical Systems (MEMS). They discuss the results of a case-retrieval experiment in which MEMS are described in terms of functional and structural features. These features are numeric, which suggest that case retrieval is carried out by means of a numeric similarity.

In an attempt to generate new product ideas, Wu et al. [111] propose a CBR system in which a product is represented as a numeric vector consisting of 87 elements. Each element represents a product attribute. The product attributes are organized into five dimensions: interface modality, task, physical feature, environment, and users. Some of the attributes in the interface modality resemble the use of the participation relation defined in ISO 15926 such as specifying the parts of the body involved in [the use of] a given product. The task dimension represents the tasks to be performed by the user through the use of the product. Attributes in this dimension are equivalent to specific processes associated to a product. The physical dimension is for attributes such as product sizes. Environment includes attributes such as indoor or outdoor places. Finally, attributes in the user dimension characterize the user in terms of gender, age, etc. Every attribute in the product vector requires a value that represents the relevancy to that attribute.

Lin et al. [15] propose the use of CBR to support the design of product service systems (PSS). Specifically, their CBR selects service strategies for a given product. A case is described in terms of 12 features which are grouped into three categories, namely, user behavior, product, and environmental environment. User behavior is specified in terms of place of usage, and frequency of usage. The product is specified in terms of features describing its fashion cycle, volume, weight, useful life, price, and subsequent expenditure. External environment is defined in terms of GDP per capita, population density, area of territory, and temperature range. Each feature is quantified using integer values. The case similarity is obtained by using a weighted summation of all the feature similarities. The weights are determined by means of the Analytic Hierarchy Process (AHP).

Appendix F: RMSE, MAPE, R for machining process ontologies

Table F-1 Evaluation of C_1=milling using the class hierarchy of the machining ontology

C_2	sim_{NGD}	sim_{WP}
counterboring	0.55	0.50
milling	1.00	1.00
countersinking	0.57	0.44
drilling	0.80	0.57
spot facing	0.53	0.50
boring	0.73	0.44
reaming	0.66	0.44
turning	0.81	0.67
tapping	0.67	0.44
grinding	0.80	0.33
blasting	0.64	0.33
lapping	0.65	0.33
RMSE		0.67
MAPE		0.24
R		0.20

Table F-2 Evaluation of C_1=countersinking using the class hierarchy of the machining ontology

C_2	sim_{NGD}	sim_{WP}
counterboring	0.86	0.83
milling	0.57	0.44
countersinking	1.00	1.00
drilling	0.63	0.60
spot facing	0.80	0.55
boring	0.67	0.67
reaming	0.77	0.67
turning	0.54	0.44
tapping	0.71	0.67
grinding	0.54	0.22
blasting	0.60	0.22
lapping	0.60	0.22
RMSE		0.86
MAPE		0.20
R		0.15

Table F-3 Evaluation of C_1=drilling using the class hierarchy of the machining ontology

C_2	sim_{NGD}	sim_{WP}
counterboring	0.61	0.67
milling	0.80	0.57
countersinking	0.63	0.60
drilling	0.99	1.00
spot facing	0.57	0.67
boring	0.77	0.60
reaming	0.72	0.60
turning	0.77	0.57
tapping	0.74	0.50
grinding	0.76	0.29
blasting	0.66	0.29
lapping	0.65	0.29
RMSE		0.45
MAPE		0.24
R		0.20

Table F-4 Evaluation of C_1=spotfacing using the class hierarchy of the machining ontology

C_2	sim_{NGD}	sim_{WP}
counterboring	0.80	0.80
milling	0.53	0.50
countersinking	0.80	0.55
drilling	0.57	0.40
spot facing	1.00	1.00
boring	0.65	0.55
reaming	0.71	0.55
turning	0.51	0.50
tapping	0.67	0.55
grinding	0.49	0.25
blasting	0.51	0.25
lapping	0.57	0.25
RMSE		0.87
MAPE		0.18
R		0.14

Table F-5 Evaluation of C_1=boring using the class hierarchy of the machining ontology

C_2	sim_{NGD}	sim_{WP}
counterboring	0.69	0.83
milling	0.73	0.44
countersinking	0.67	0.67
drilling	0.77	0.50
spot facing	0.65	0.55
boring	1.00	1.00
reaming	0.79	0.83
turning	0.72	0.44
tapping	0.78	0.83
grinding	0.70	0.22
blasting	0.67	0.22
lapping	0.71	0.22
RMSE		0.61
MAPE		0.28
R		0.22

Table F-6 Evaluation of C_1=reaming using the class hierarchy of the machining ontology

C_2	sim_{NGD}	sim_{WP}
counterboring	0.77	0.83
milling	0.66	0.44
countersinking	0.77	0.67
drilling	0.72	0.50
spot facing	0.71	0.55
boring	0.79	0.83
reaming	1.00	1.00
turning	0.64	0.44
tapping	0.80	0.83
grinding	0.64	0.22
blasting	0.66	0.22
lapping	0.69	0.22
RMSE		0.86
MAPE		0.25
R		0.20

Table F-7 Evaluation of C_1=turning using the class hierarchy of the machining ontology

C_2	sim_{NGD}	sim_{WP}
counterboring	0.55	0.50
milling	0.87	0.67
countersinking	0.58	0.44
drilling	0.80	0.57
spot facing	0.51	0.50
boring	0.77	0.44
reaming	0.66	0.44
turning	1.00	1.00
tapping	0.71	0.44
grinding	0.86	0.33
blasting	0.65	0.33
lapping	0.66	0.33
RMSE		0.60
MAPE		0.26
R		0.22

Table F-8 Evaluation of C_1=tapping using the class hierarchy of the machining ontology

C_2	sim_{NGD}	sim_{WP}
counterboring	0.72	0.83
milling	0.86	0.44
countersinking	0.73	0.67
drilling	0.86	0.50
spot facing	0.68	0.55
boring	0.92	0.83
reaming	0.86	0.83
turning	1.00	0.44
tapping	1.00	1.00
grinding	0.84	0.22
blasting	0.85	0.22
lapping	0.88	0.22
RMSE		0.06
MAPE		0.39
R		0.30

Table F-9 Evaluation of C_1=grinding using the class hierarchy of the machining ontology

C_2	sim_{NGD}	sim_{WP}
counterboring	0.53	0.25
milling	0.80	0.33
countersinking	0.54	0.22
drilling	0.76	0.29
spot facing	0.49	0.25
boring	0.70	0.22
reaming	0.64	0.22
turning	0.78	0.33
tapping	0.65	0.22
grinding	1.00	1.00
blasting	0.68	0.67
lapping	0.73	0.67
RMSE		0.70
MAPE		0.35
R		0.30

Table F-10 Evaluation of C_1=blasting using the class hierarchy of the machining ontology

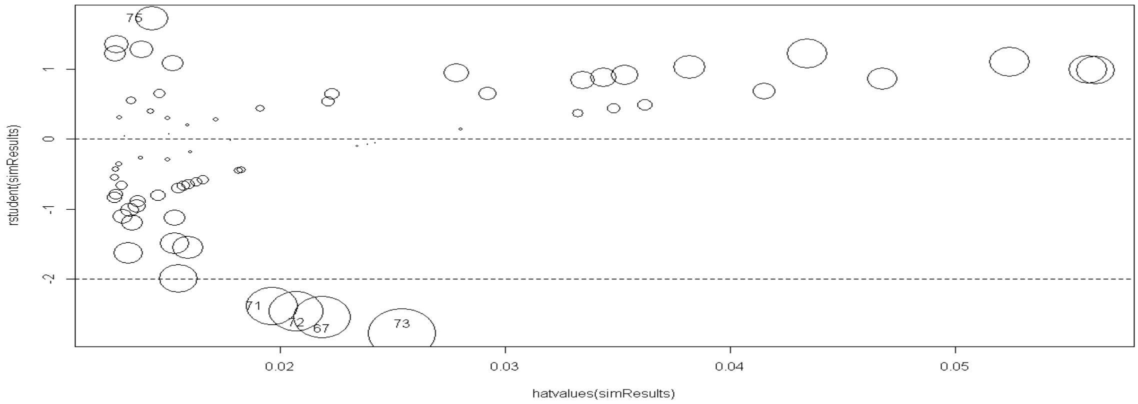
C_2	sim_{NGD}	sim_{WP}
counterboring	0.55	0.25
milling	0.64	0.33
countersinking	0.60	0.22
drilling	0.66	0.29
spot facing	0.51	0.25
boring	0.67	0.22
reaming	0.66	0.22
turning	0.61	0.33
tapping	0.70	0.22
grinding	0.68	0.67
blasting	1.00	1.00
lapping	0.72	0.67
RMSE		0.81
MAPE		0.32
R		0.28

Table F-11 Evaluation of C_1=lapping using the class hierarchy of the machining ontology

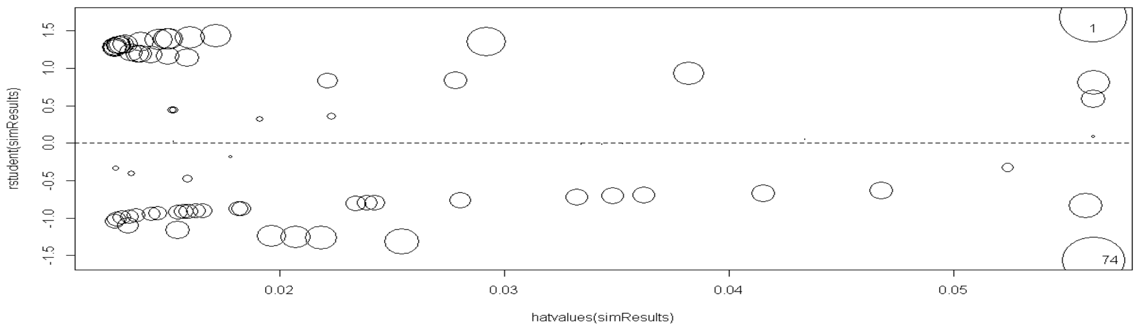
C_2	sim_{NGD}	sim_{WP}
counterboring	0.59	0.25
milling	0.65	0.33
countersinking	0.60	0.22
drilling	0.65	0.29
spot facing	0.57	0.25
boring	0.71	0.22
reaming	0.69	0.22
turning	0.65	0.33
tapping	0.71	0.22
grinding	0.73	0.67
blasting	0.72	0.67
lapping	1.00	1.00
RMSE		0.84
MAPE		0.34
R		0.30

Appendix G: Regression analysis

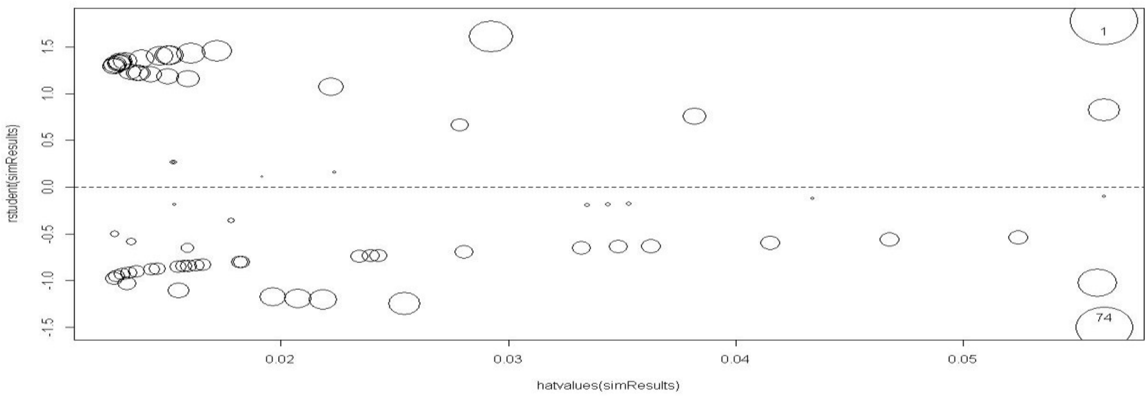
Following graphs are the results of studentized residual against hat value for each similarity measure against Google Scholar



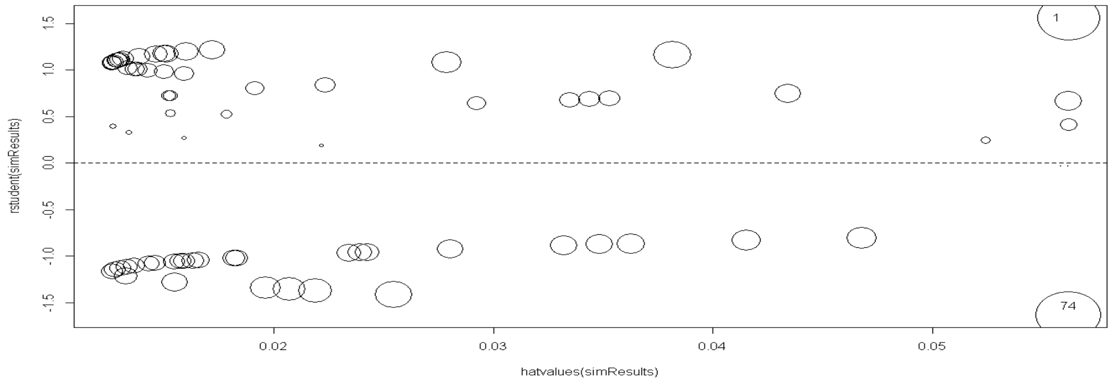
sim_{Lin}



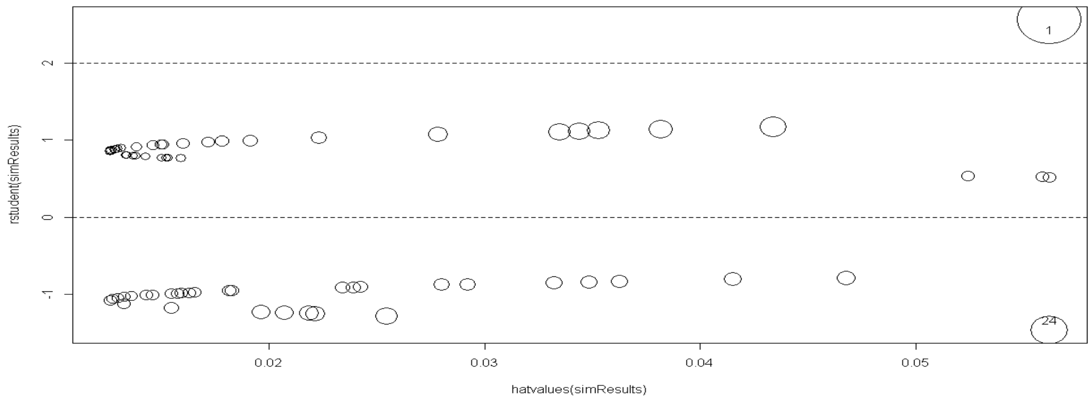
sim_{Dice}



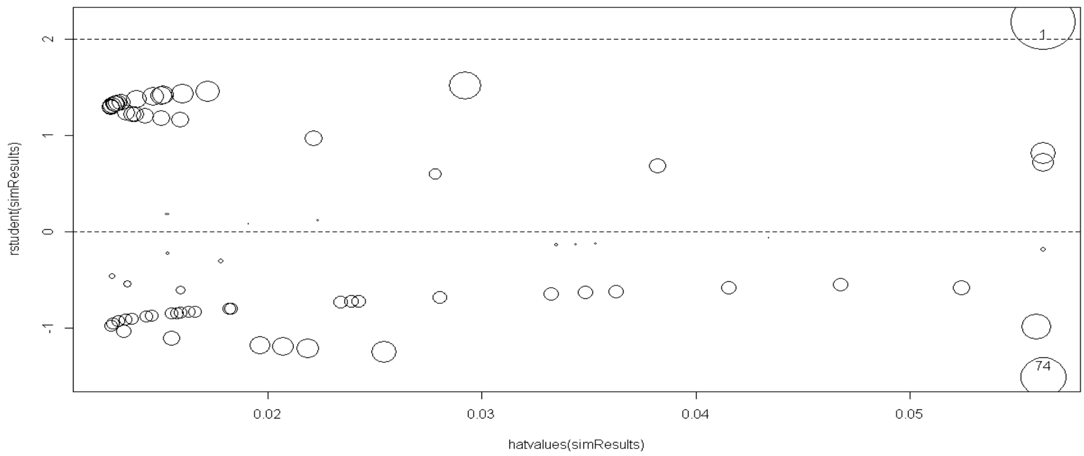
$sim_{All\ confidence}$



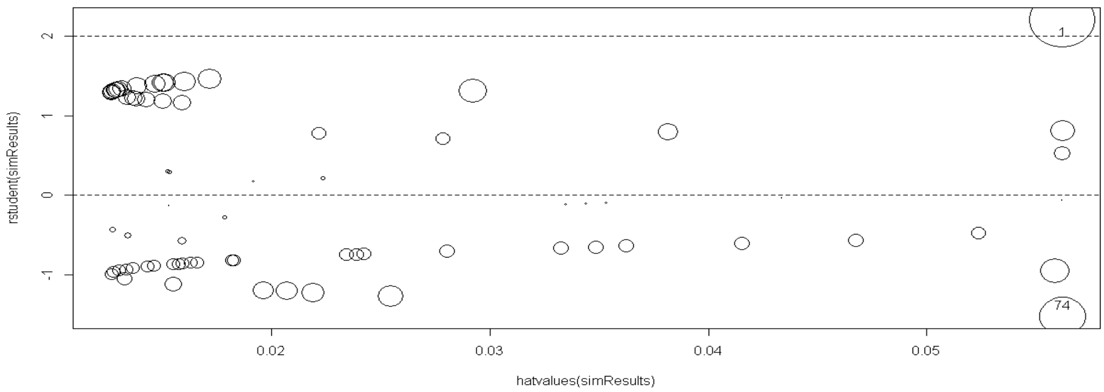
$\text{sim}_{\text{overlap}}$



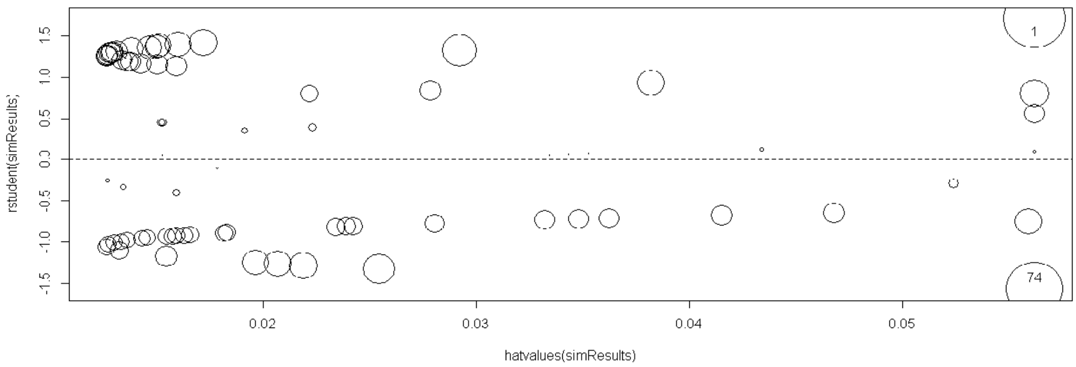
$\text{sim}_{\text{vanDerWeken1}}$



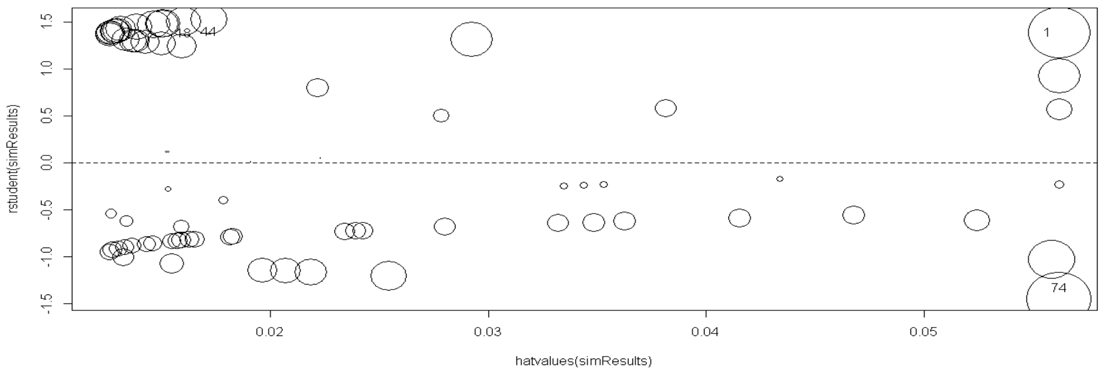
$\text{sim}_{\text{vanDerWeken2}}$



$sim_{Jaccard}$



sim_{cosine}



$sim_{Tversky}$

Appendix H: Successful PSS cases and case description

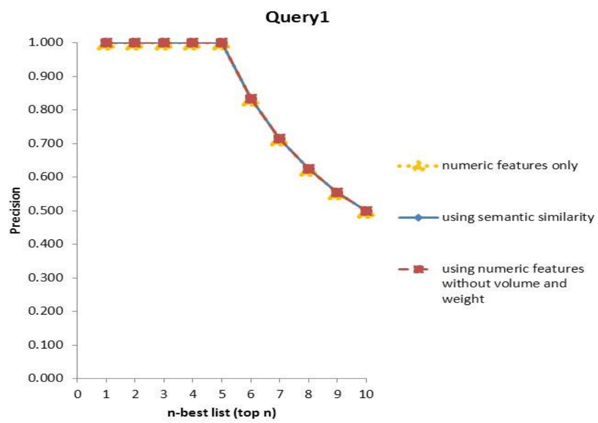
Table H-1 Successful PSS cases [15]

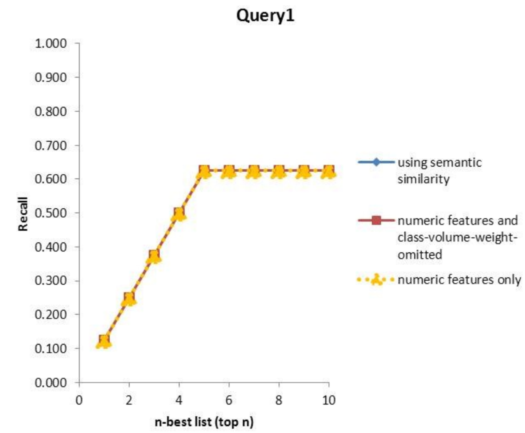
PSS strategies	Cases
Maintenance and repair	(1) washing machine repair, (2) refrigerator repair, (3) computer repair, (4) laser printer repair, (5) LCD monitor repair, (6) water heater repair, (7) handbag maintenance and repair, (8) jewelry maintenance and repair, (9) watch maintenance and repair
Leasing	(10) treadmill rental, (11) washing machine rental, (12) LCD TV rental, (13) sofa rental, (14) dryer rental, (15) platform bed rental, (16) refrigerator rental, (17) credenzas rental, (18) dish washer rental
Rental	(19) luggage box rental, (20) video CD/DVD rental, (21) evening dress rental, (22) entertainment book rental, (23) video game rental, (24) jewelry rental, (25) handbag rental, (26) GPS rental, (27) DV rental
Pay per service unit	(28) fax service, (29) printing service, (30) copying service, (31) scanning service, (32) laminating service, (33) clothes washing service, (34) cleaning service, (35) shoes cleaning service, (36) eyeglass cleaning service, (37) photographic service
Functional result	(38) online music, (39) online magazine, (40) online KTV, (41) music download, (42) multimedia on demand, (43) digital map, (44) online newspaper, (45) online dictionary, (46) digital calendar, (47) book (audio book) download

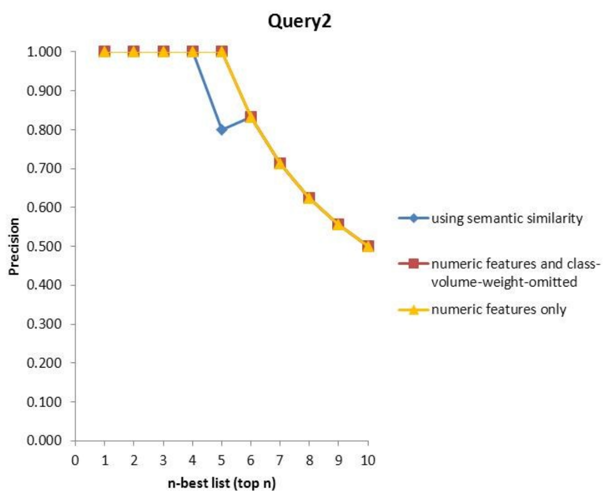
Table H-2 Case description			
Categories	Index	Definition	Index value
User behavior	Place of usage	The place where the product is often used	1 = indoor, 2 = both, 3 = outdoor
	Frequency of usage	The frequency that the product is often used	1 = 1 time / month, 2 = 2 times /month, 3 = 4 times / month, 4 = 8 times / month, 5 = 1 time / day, 6 = more than 1 time / day, 7 = anytime
Product	Product fashion cycle	The period of the product existing in the market (The four stages are product introduction, growth, maturity and decline)	1 = less than 3 months, 2 = 3~6 months, 3 = 6~9 months, 4 = 9~12 months, 5 = more than 12 months
	Volume	Volume of the product	1 = about 1,000 cm^3, 2 = about 5,000 cm^3, 3 = about 50,000 cm^3, 4 = about 100,000 cm^3, 5 = more than 200,000 cm^3
	Weight	Weight of the product	1 = about 4 kg, 2 = about 8 kg, 3 = about 12 kg, 4 = about 20 kg, 5 = more than 30 kg
	Useful life	It is the length of time that any manufactured item can be expected to be 'serviceable' or supported by its originating manufacturer.	1~100 years
	Price	Expenditure of the product purchasing	1 = less than 30 US dollars, 2 = 31 ~ 300 US dollars, 3 = 301 ~ 900 US dollars, 4 = 901 ~ 1500 US dollars, 5 = more than 1500 US dollars
	Subsequent expenditure	The total expenditure of the product after you bought it.	1 = less than 30%, 2 = 30~60%, 3 = more than 60%

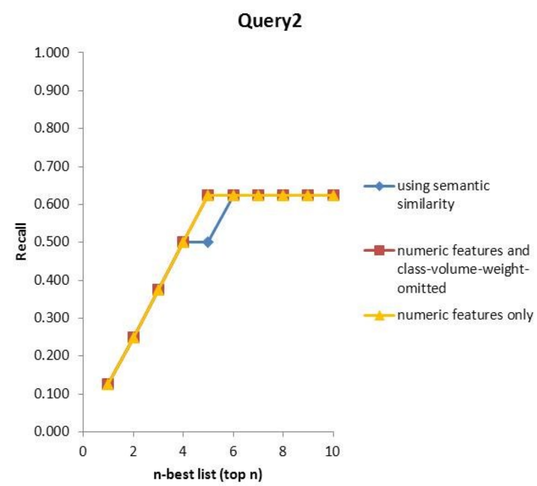
Categories	Index	Definition	Index value
External environment	GDP per capita	Gross Domestic Product per capita of target market	1 = less than 10,000 US dollars, 2 = 10,000 ~ 20,000 US dollars, 3 = 20,001 ~ 30,000 US dollars, 4 = 30,001 ~ 40,000 US dollars, 5 = more than 40,000 US dollars
	Population density	Number of people per km2 of the target market	1 = less than 50 persons /km^2, 2 = 51 ~ 100 persons /km^2, 3 = 101 ~ 150 persons / km^2, 4 = 151 ~ 200 persons / km^2, 5 = more than , 200 persons / km^2
	Area of territory	Total area of target market	1 = less than 5 km^2, 2 = 5 ~ 37km^2, 3 = 38 ~ 69 km^2, 4 = 70~100 km^2, 5 = more than , 100 km^2
	Temperature range	The difference between minimum and maximum monthly mean temperature in target market	1 = less than 5 ^{0}C, 2 = 5 ~ 10^{0}C, 3 = 10.1 ~ 15 ^{0}C, 4 = 15.1~20 ^{0}C, 5 = more than 20 ^{0}C

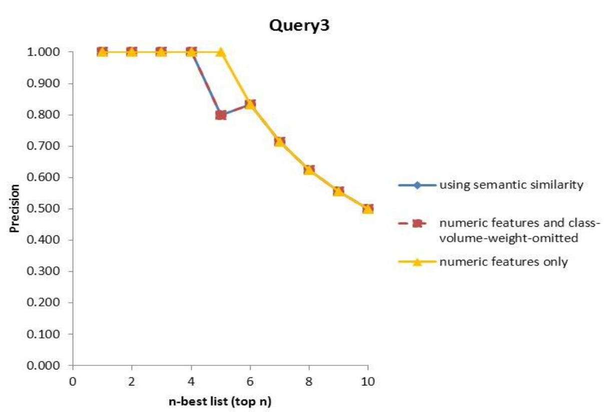
Appendix I: Precision-recall graphs

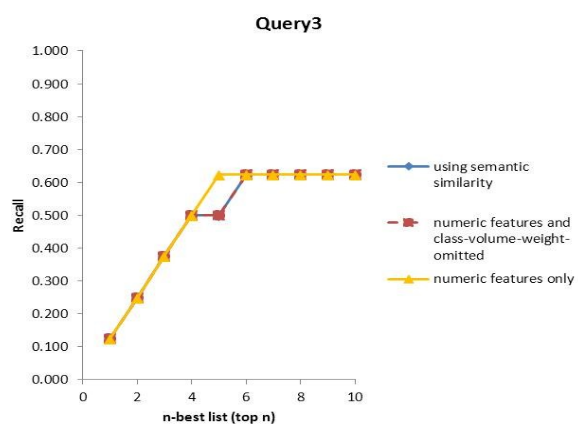


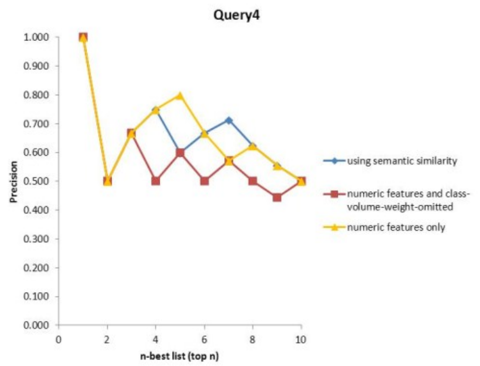


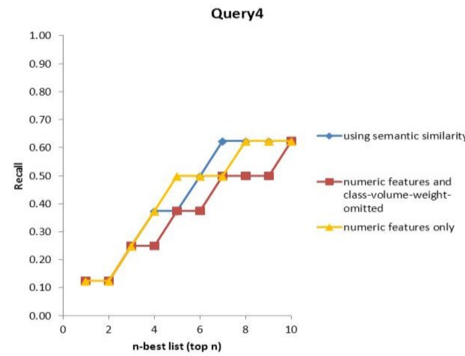


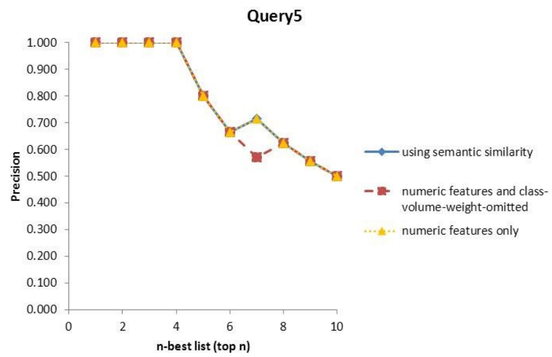


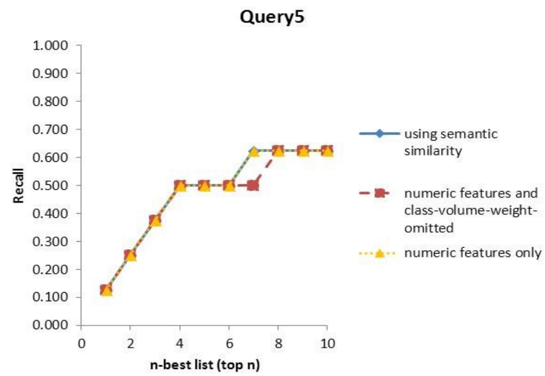


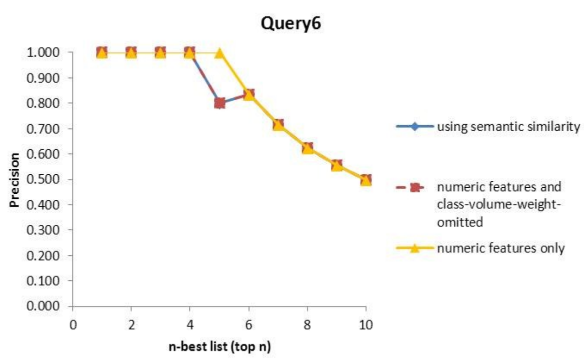


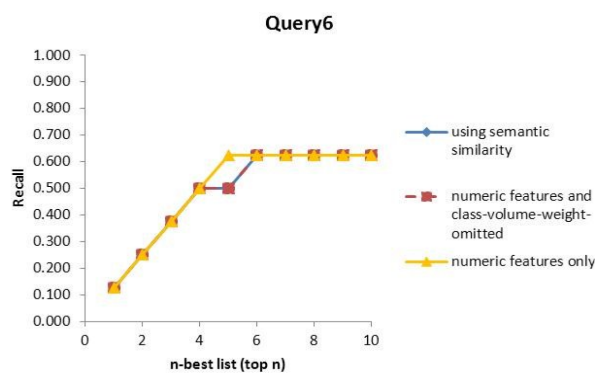

Query4
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

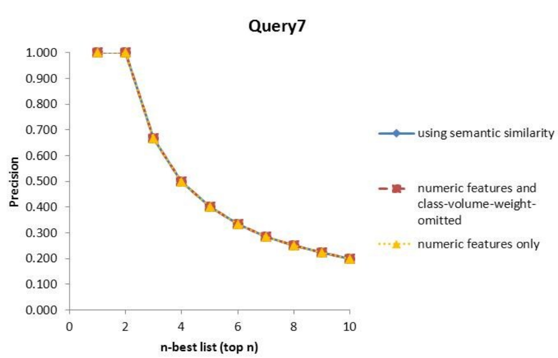

Query4
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

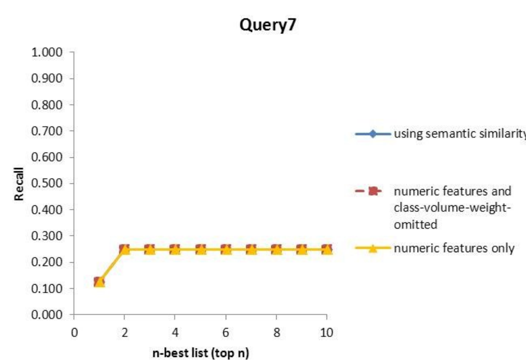

Query5
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

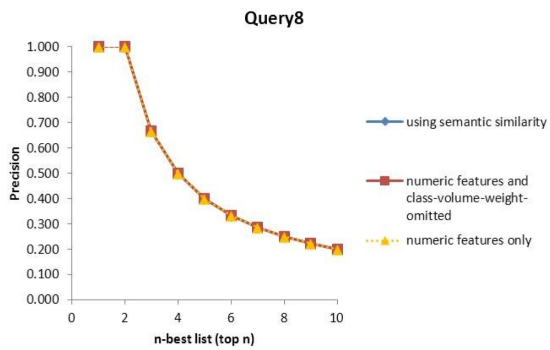

Query5
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

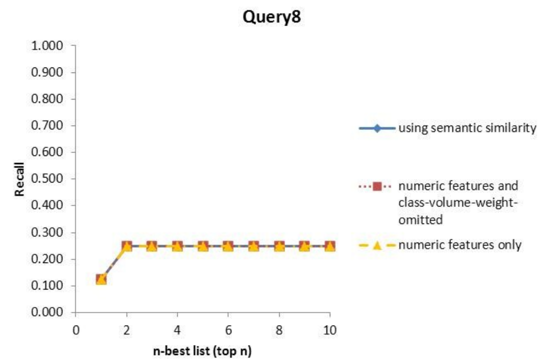

Query6
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

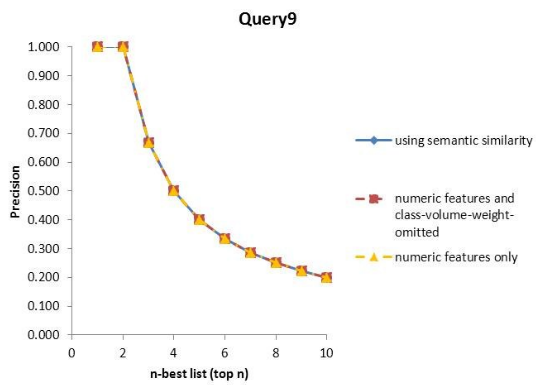

Query6
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

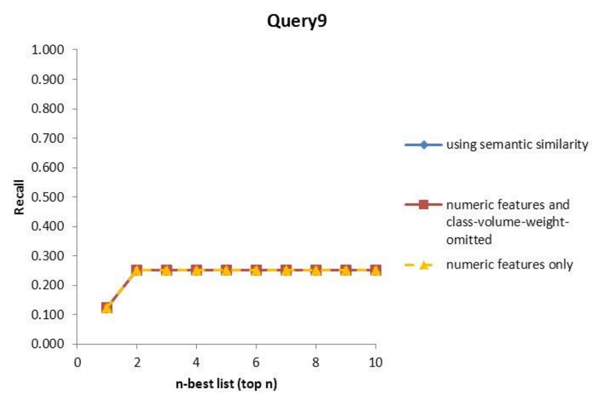

Query7
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

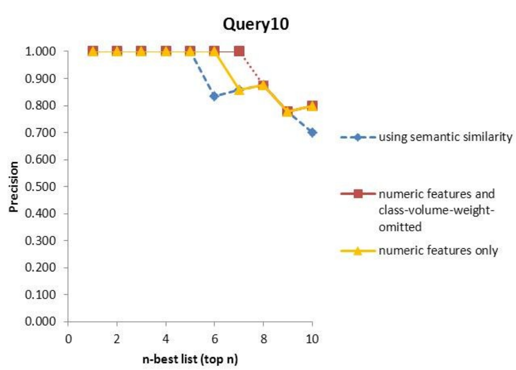

Query7
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

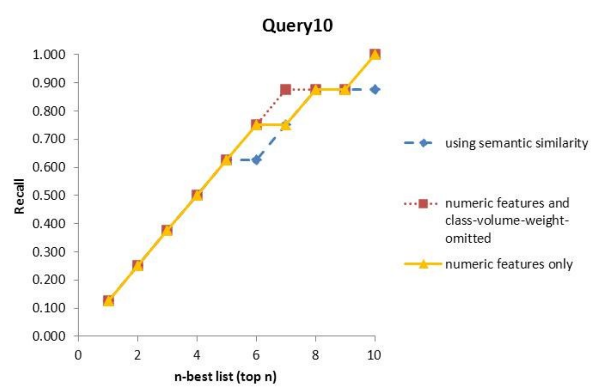

Query8
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

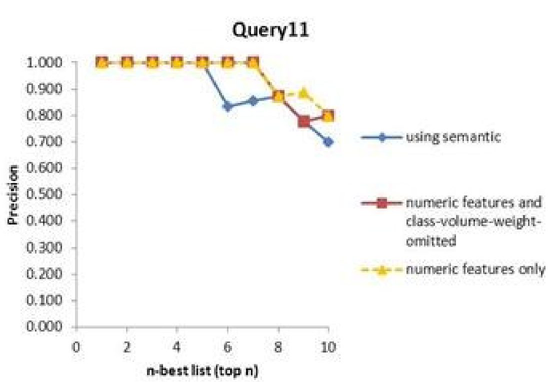

Query8
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

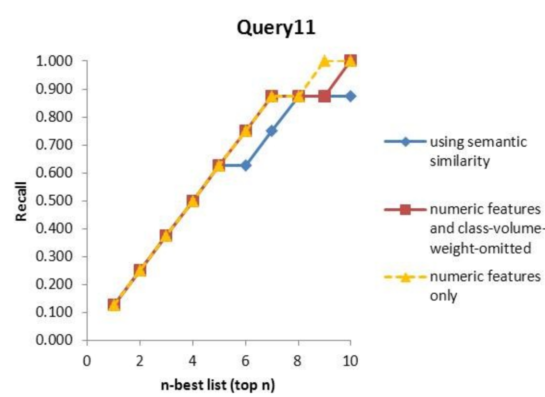

Query9
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

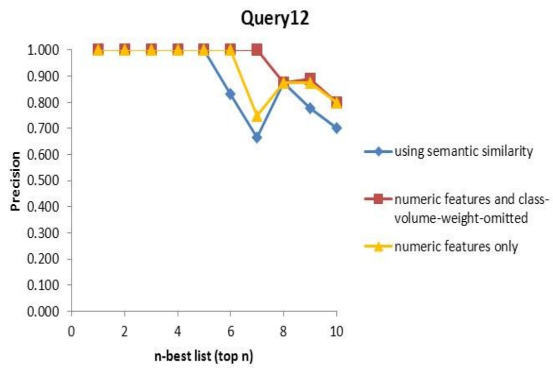

Query9
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

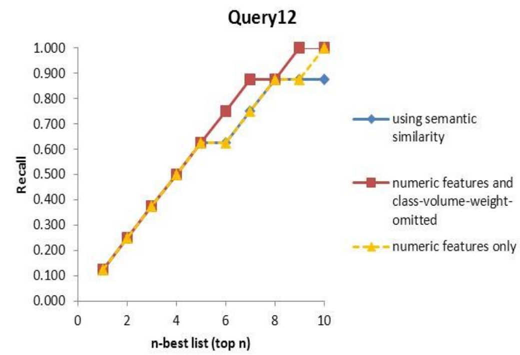

Query10
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

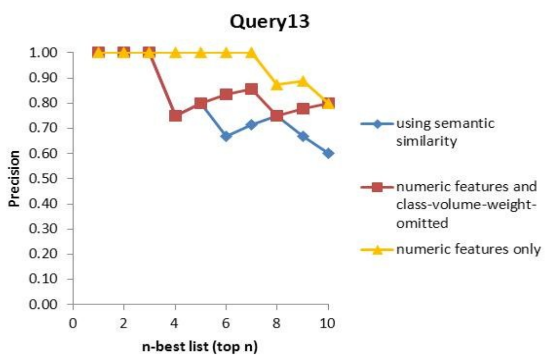

Query10
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

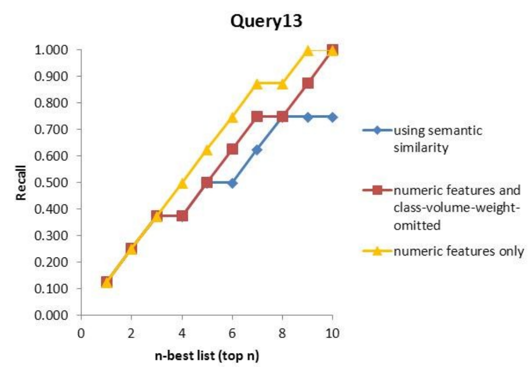

Query11
Precision
n-best list (top n)
using semantic
numeric features and class-volume-weight-omitted
numeric features only

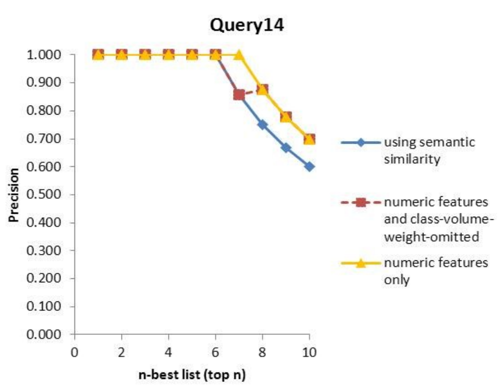

Query11
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

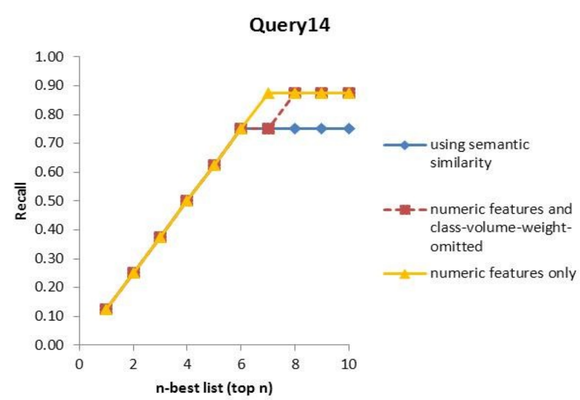

Query12
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

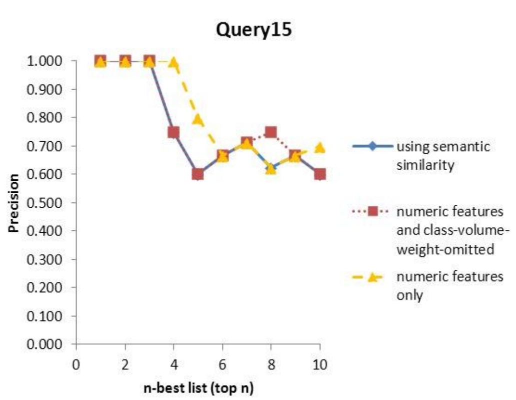

Query12
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

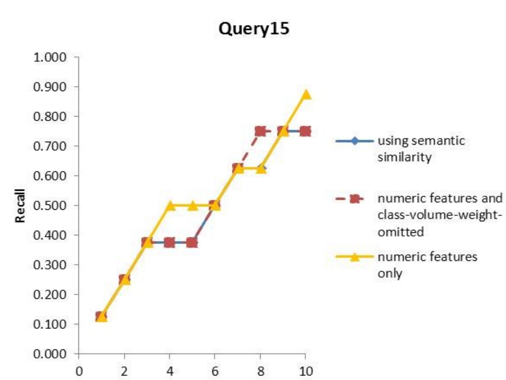

Query13
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only


Query13
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

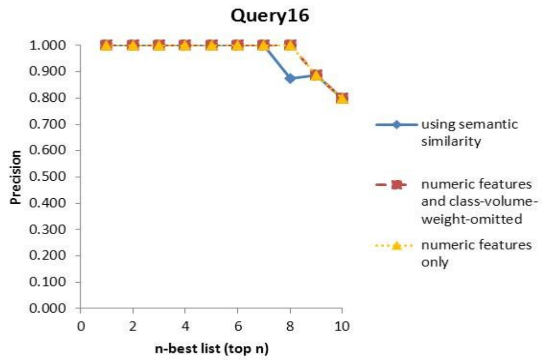

Query14
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

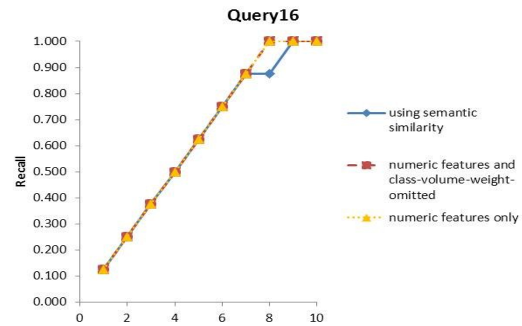

Query14
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

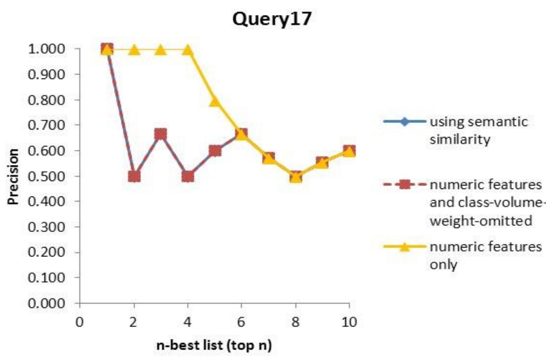

Query15
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

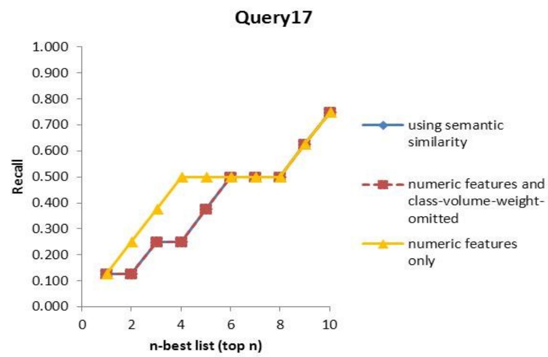

Query15
Recall
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

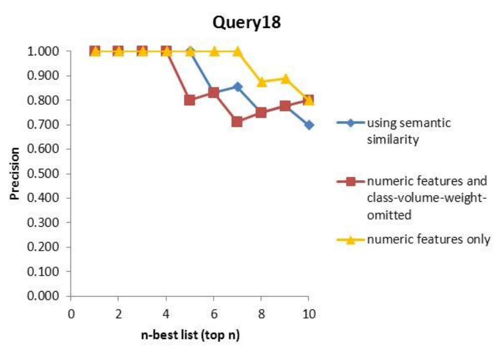
Appendix I

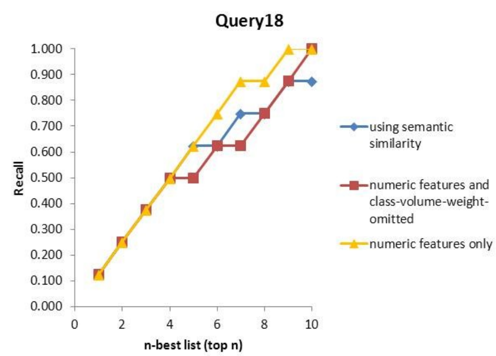

Query16
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

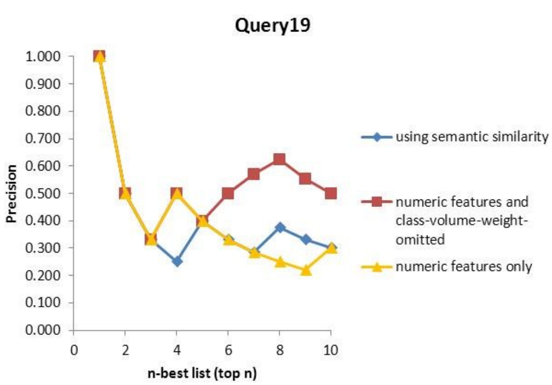

Query16
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

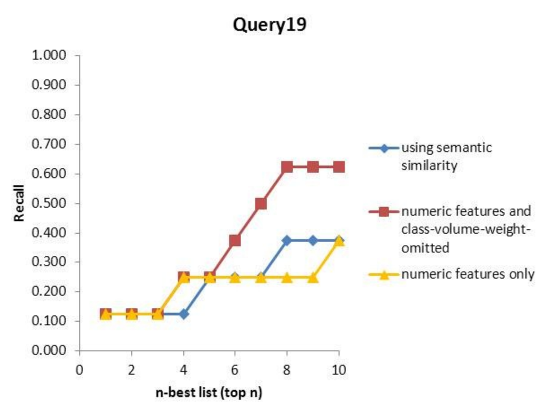

Query17
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

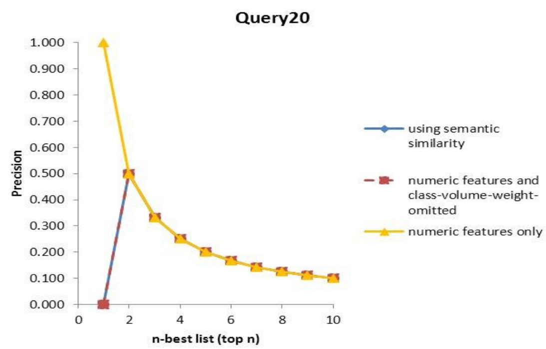

Query17
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

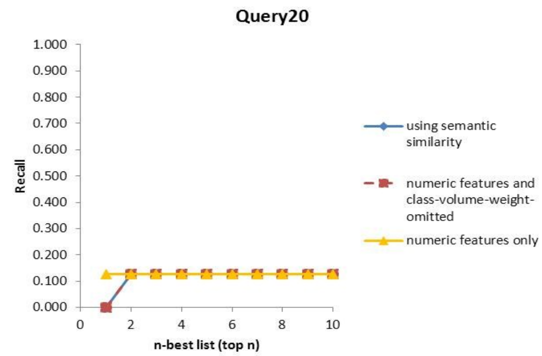

Query18
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

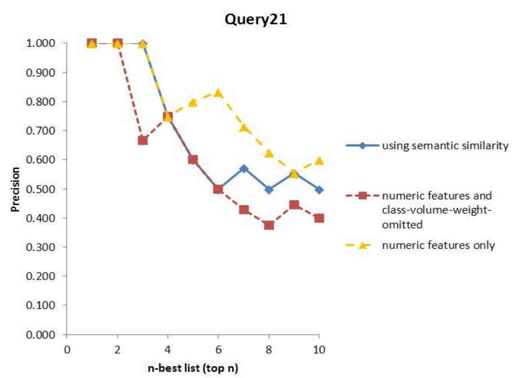

Query18
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

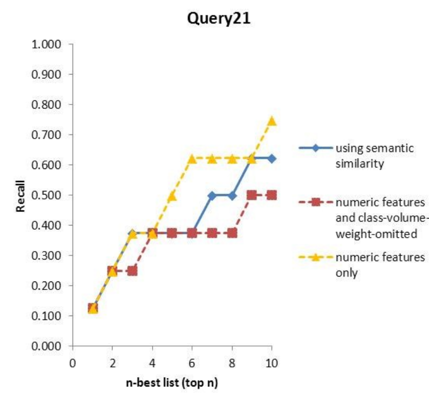

Query19
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

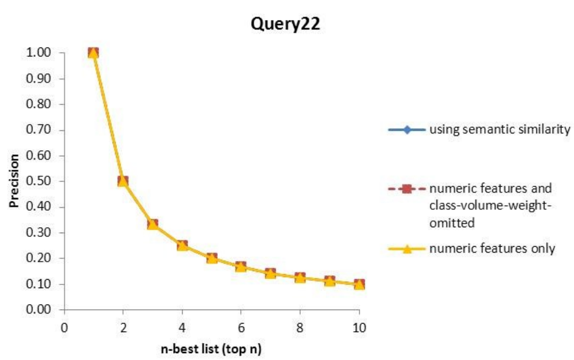

Query19
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

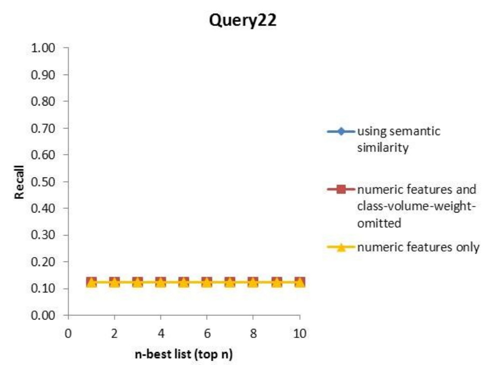

Query20
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

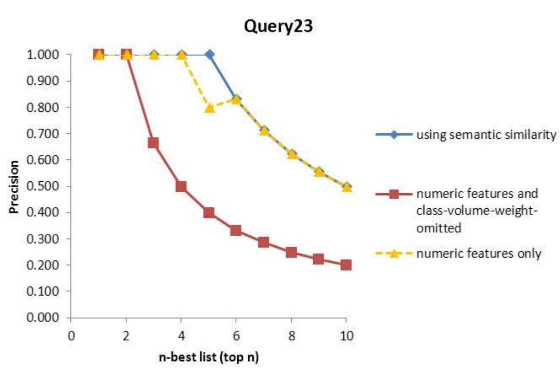

Query20
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

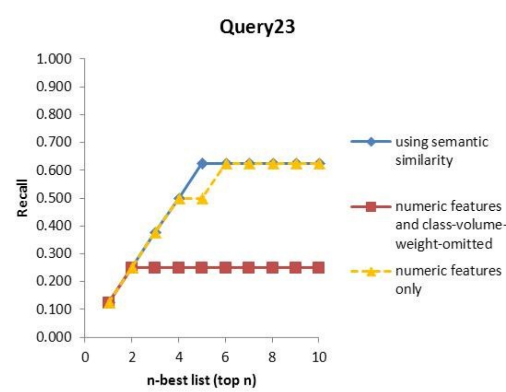

Query21
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only


Query21
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

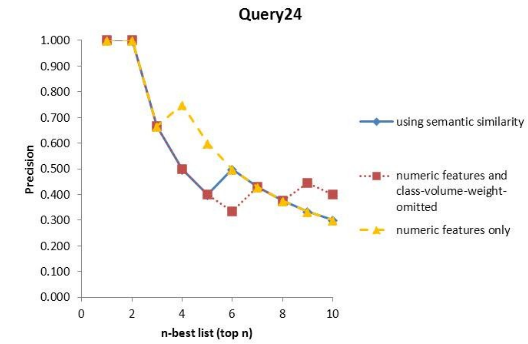

Query22
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

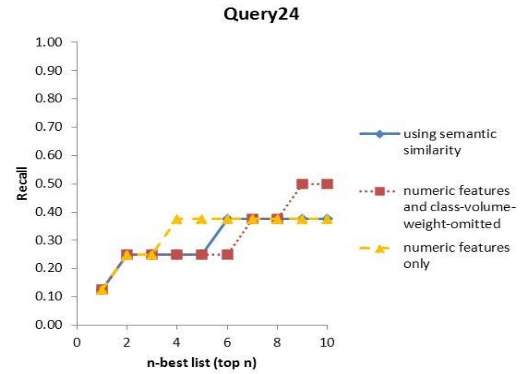

Query22
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

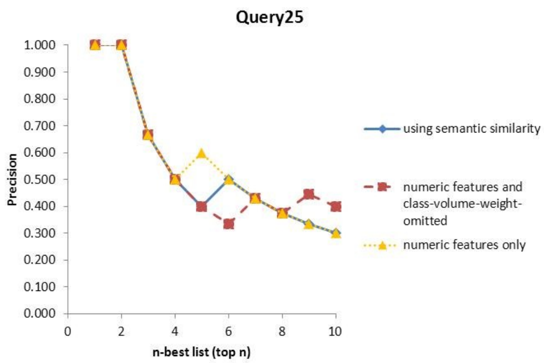

Query23
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

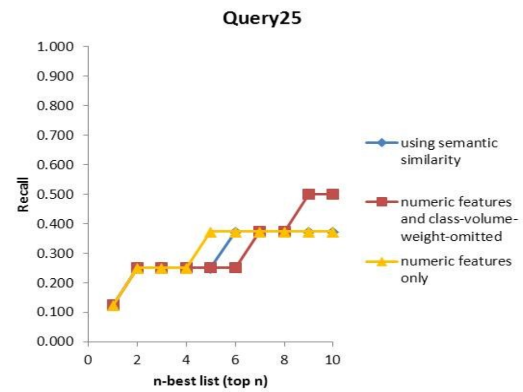

Query23
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

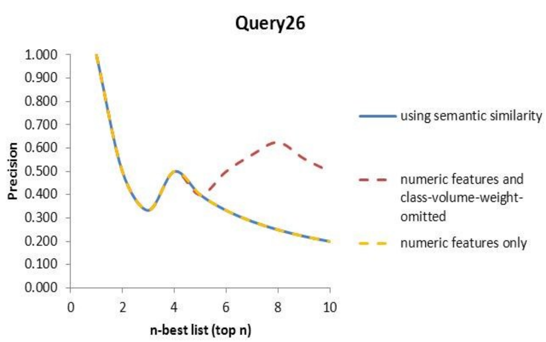
Appendix I

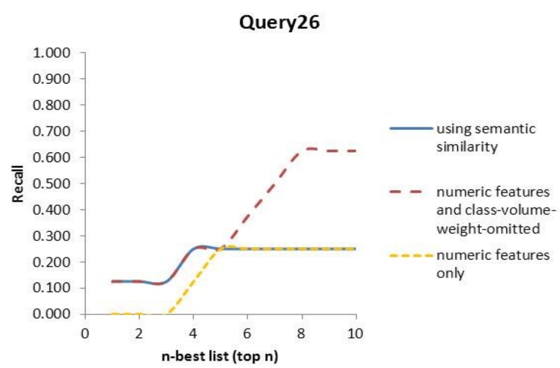

Query24
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

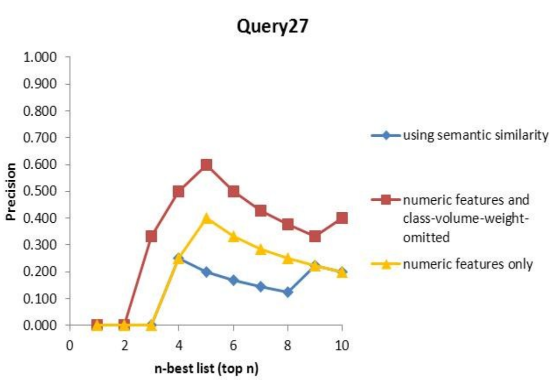

Query24
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

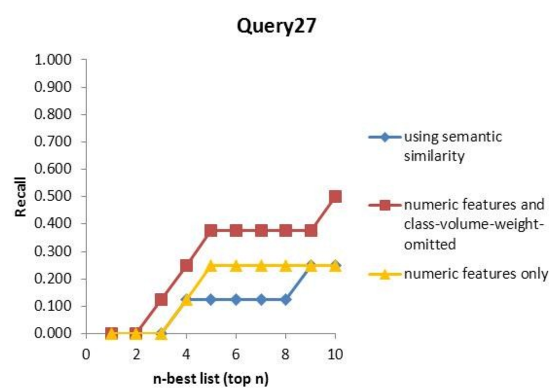

Query25
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

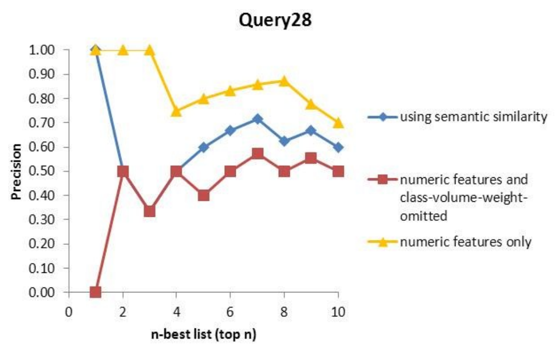

Query25
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

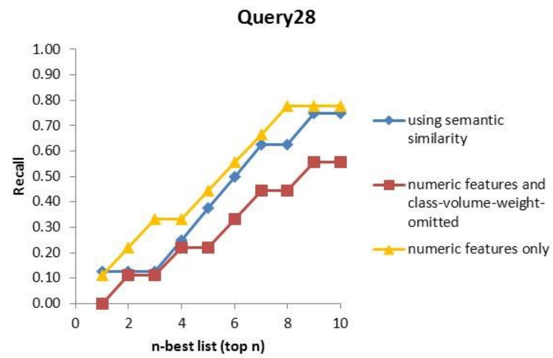

Query26
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

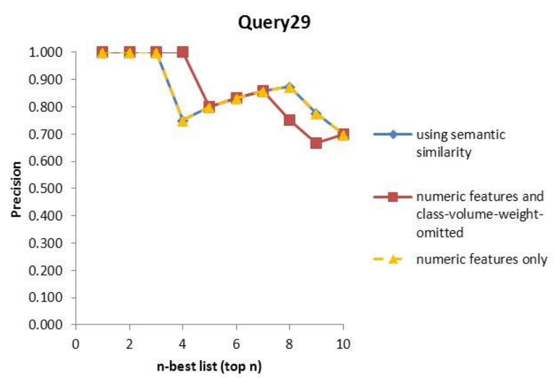

Query26
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

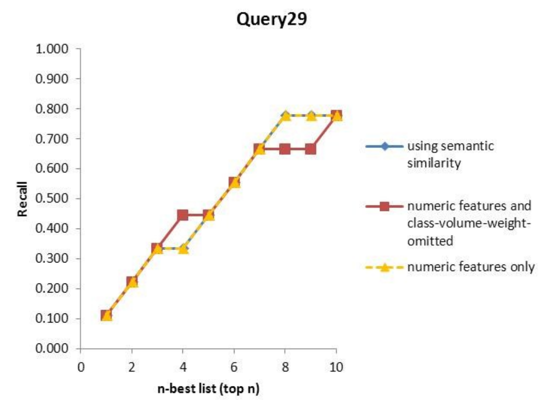

Query27
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

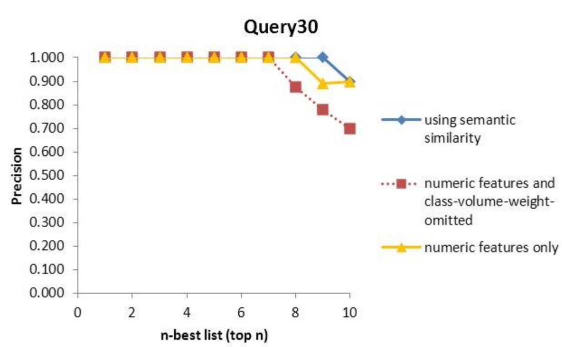

Query27
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

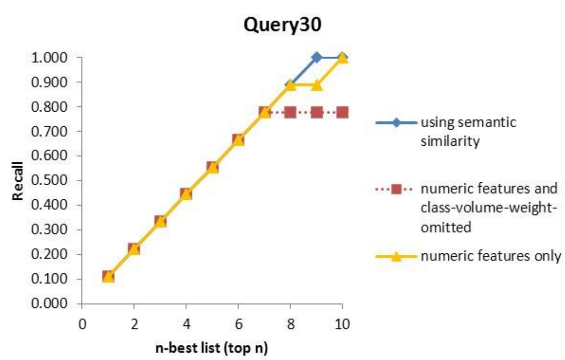

Query28
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

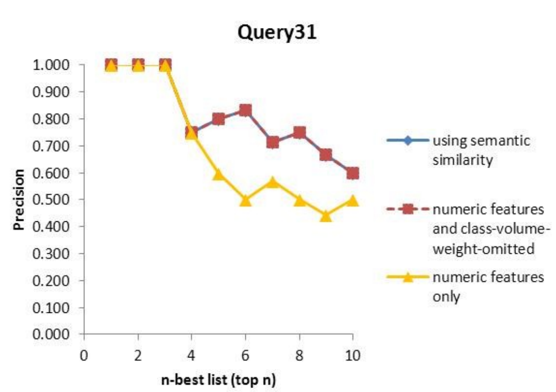

Query28
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

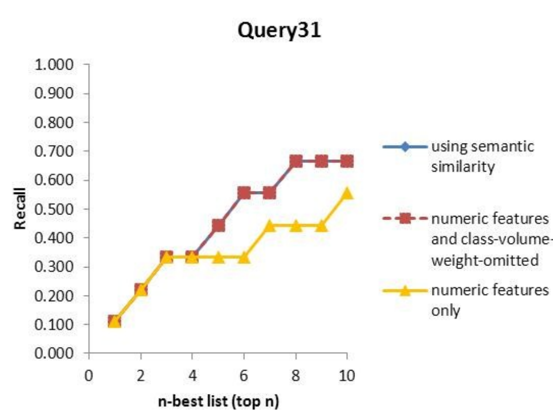

Query29
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

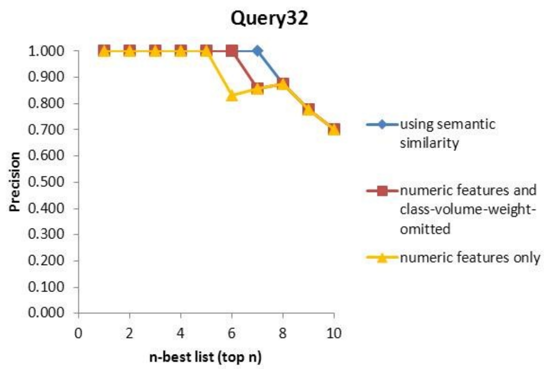

Query29
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

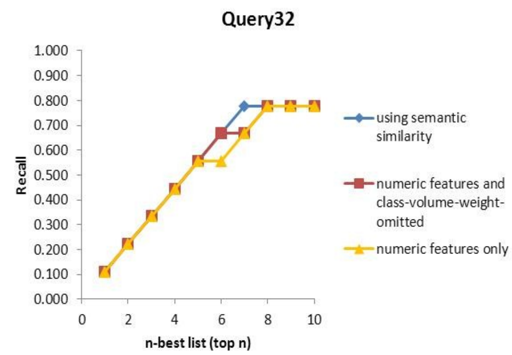

Query30
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

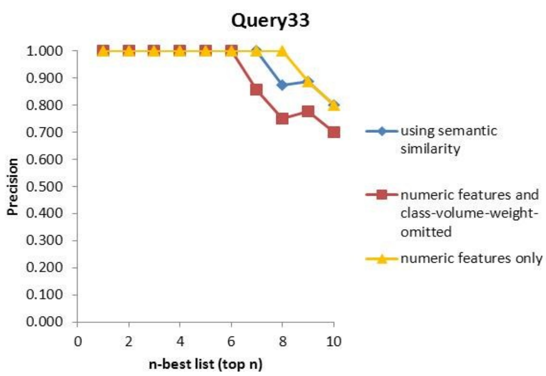

Query30
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

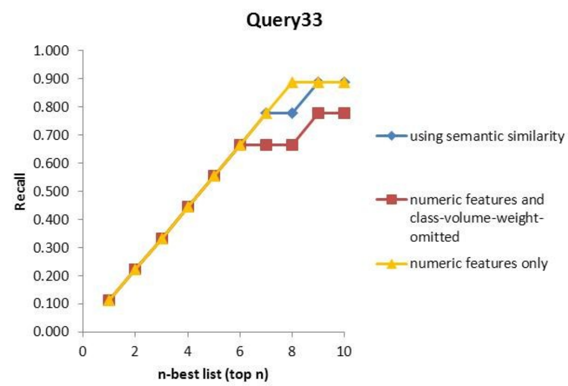

Query31
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

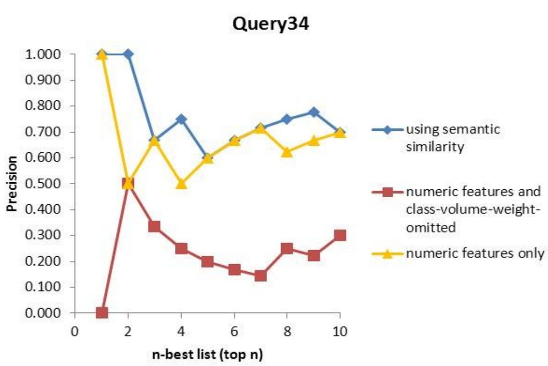

Query31
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

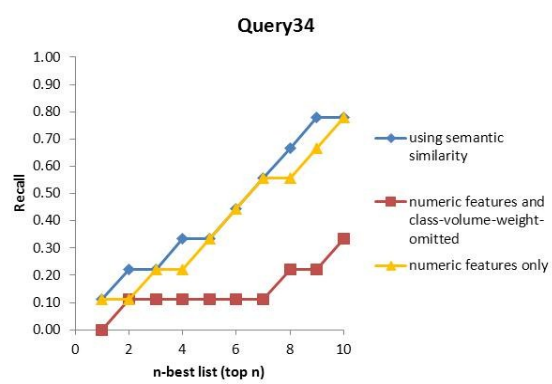

Query32
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

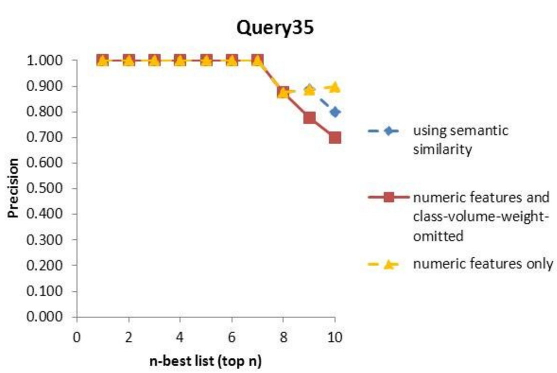

Query32
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

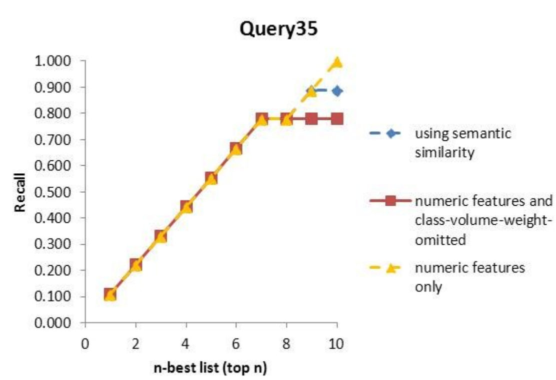

Query33
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

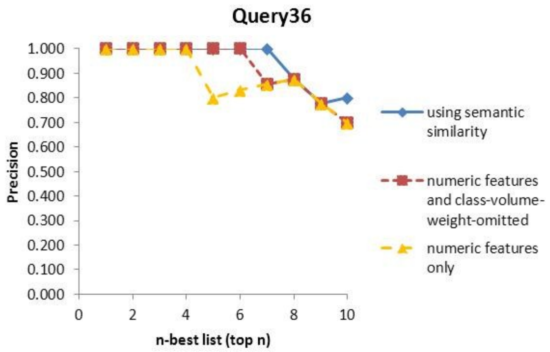

Query33
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

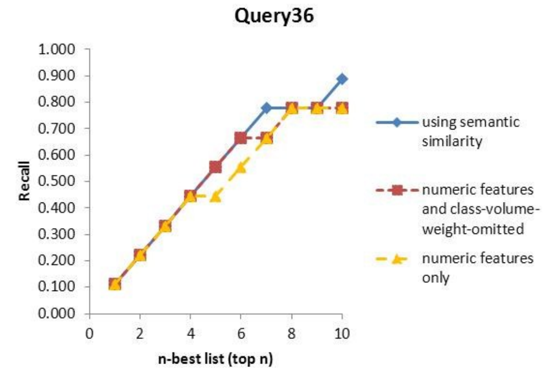

Query34
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

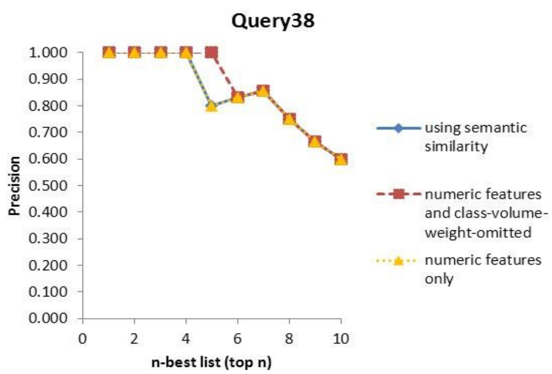

Query34
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

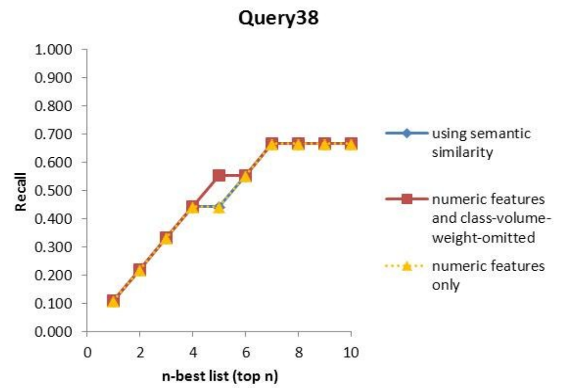

Query35
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

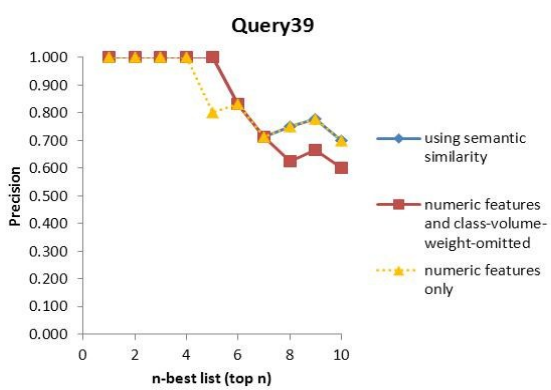

Query35
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

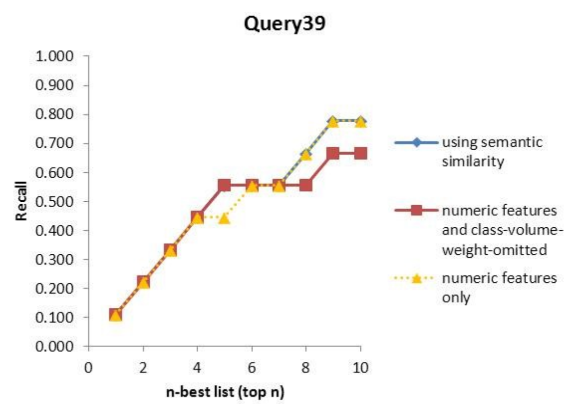

Query36
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

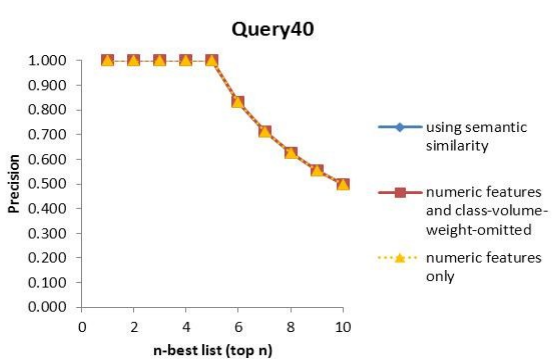

Query36
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

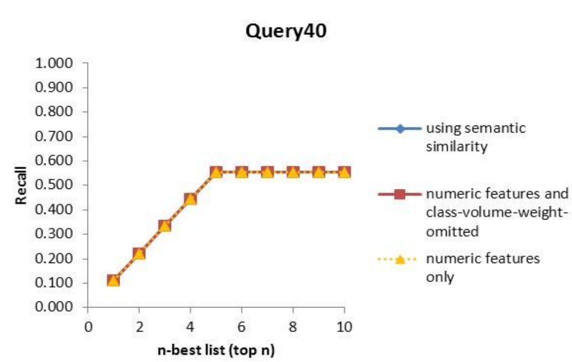

Query38
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

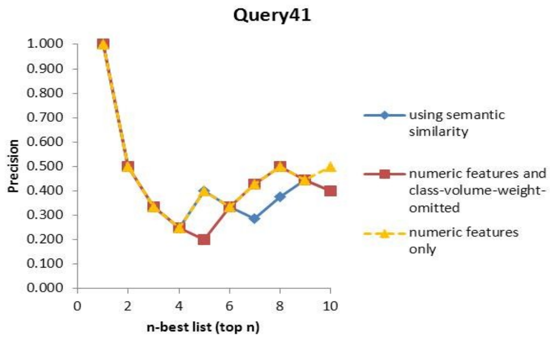

Query38
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

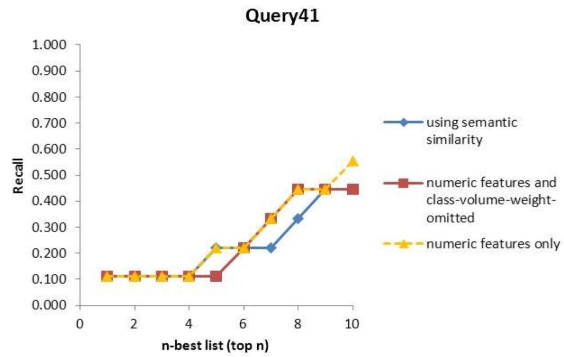

Query39
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

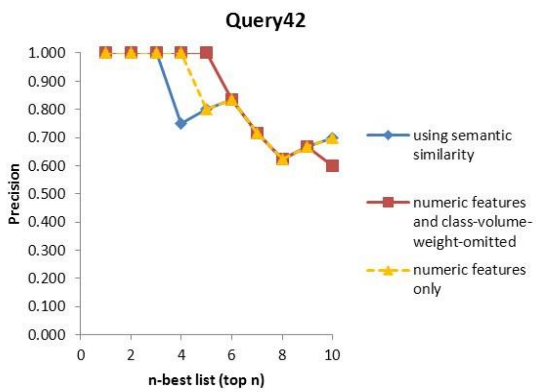

Query39
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

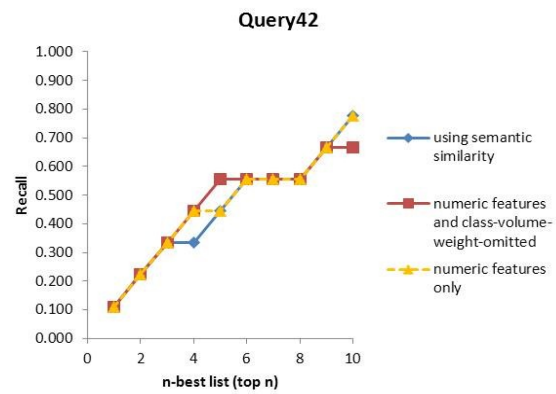

Query40
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

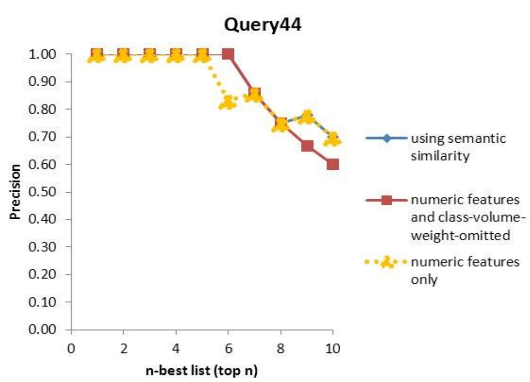

Query40
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

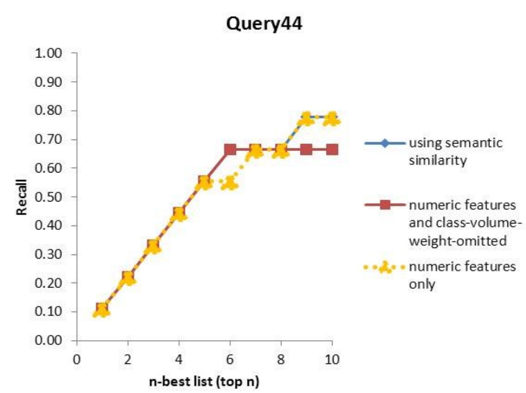

Query41
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

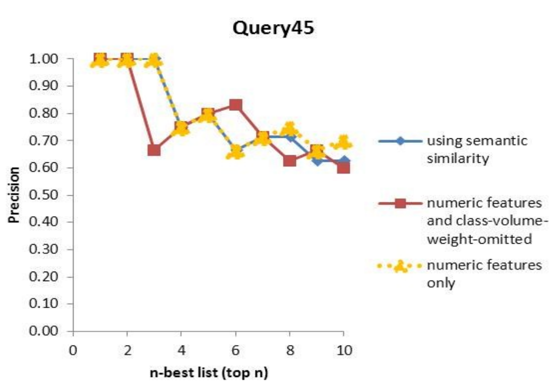

Query41
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

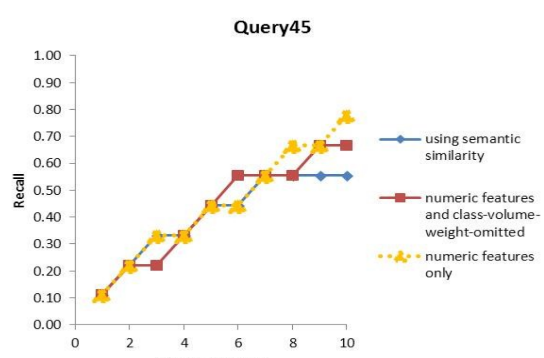

Query42
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

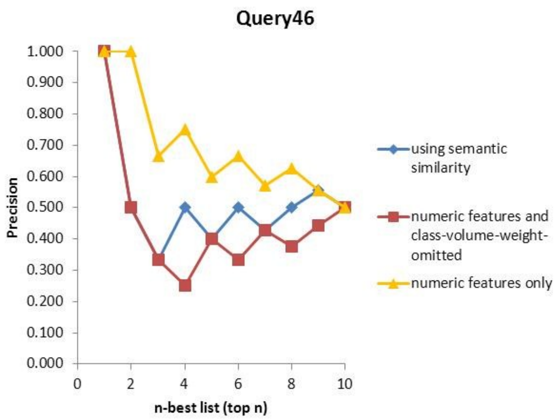

Query42
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

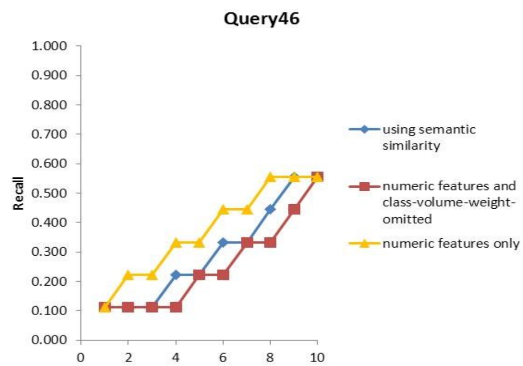

Query44
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

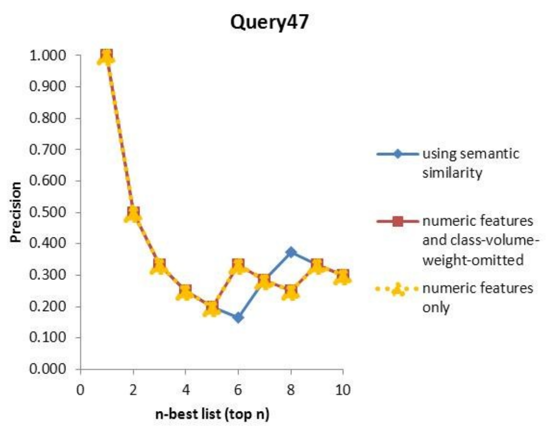

Query44
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only

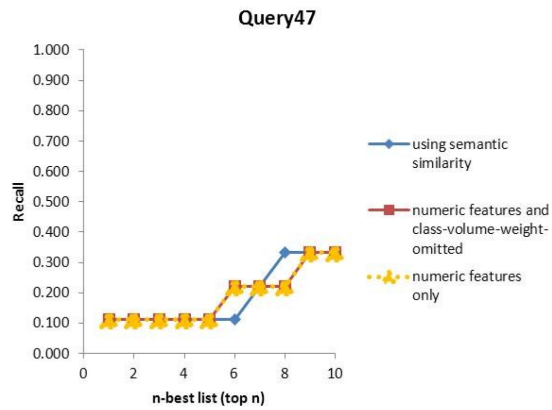

Query45
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only


Query45
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only


Query46
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only


Query46
Recall
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only


Query47
Precision
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only


Query47
Recall
n-best list (top n)
using semantic similarity
numeric features and class-volume-weight-omitted
numeric features only