Ecole Nationale Supérieure des Mines de Paris

THESE

pour obtenir le grade de

DOCTEUR

Discipline : Automatique, Informatique et Robotique

Ecole Doctorale :

présentée et soutenue publiquement

par

Reena T. N. Shetty

Le 12 Novembre 2008

Enrichissement de réseaux sémantiques par la proximité de concepts

Jury

Rapporteurs : Bruno Bachimont, Enseignant-chercheur à l'UTC et Directeur Scientifique à l'INA
Jean Charlet, Chercheur, Inserm U729/STIM, Ecole Centrale de Paris

Examinateurs : Yves Rouchaleau, Professeur, Ecole des mines de Paris
Marie-Thérèse Ménager, Direction programme ToxNuc-E, CEA

Directeur : Joël Quinqueton, Directeur de thèse - Professeur, Université de Montpellier
Encadrant : Pierre-Michel Riccio, Ingénieur de recherche, Laboratoire LGI2P / Ecole des Mines d'Alès

Remerciements

Je remercie Yannick Vimont, directeur du LGI2P, ainsi que la direction de l'Ecole des Mines d'Alès pour m'avoir permis de préparer cette thèse dans leur établissement.

Je désire remercier sincèrement mon directeur de thèse, Joël Quinqueton, professeur au LIRMM de l'Université Montpellier II, dont les qualités sont nombreuses tant sur le plan scientifique que humain. Son don précieux de toujours ressortir les points positifs d'une situation complexe est l'une de ses grandes qualités.

Je remercie mon encadrant de thèse, Pierre-Michel Riccio, Ingénieur de recherche, Laboratoire LGI2P / Ecole des Mines d'Alès. Je tiens aussi à remercier mes collègues Imane Anoir et Jean Villerd pour l'aide apportée dans le recueil initial des données et dans la validation de mon modèle.

Je suis très sensible à la présence dans ce jury de Bruno Bachimont, Bruno Bachimont, Enseignant-chercheur à l'UTC et Directeur Scientifique à l'INA, Jean Charlet, Chercheur, Inserm U729/STIM, Ecole Centrale de Paris, Yves Rouchaleau, Professeur, Ecole des mines de Paris et Marie-Thérèse Ménager, Direction programme ToxNuc-E, CEA.

Je remercie également les enseignants chercheurs, les secrétaires et l'ensemble du personnel du LGI2P pour l'ambiance inoubliable et l'esprit d'amitié.

Merci aussi à mes collègues et amis Nicolas, et Fabien ainsi qu'à tous les autres qui se reconnaîtront ici.

Merci enfin à mes parents, à mon mari, ma sœur, mon frère et à ma famille ainsi qu'à mes amis qui m'ont permis de me ressourcer à chaque retour dans Inde.

Résumé étendu

Les dernières années ont vu le déferlement d'une vague d'information sous forme électronique, due à l'usage croissant du World Wide Web (WWW). Pour beaucoup, le World Wide Web est devenu un moyen essentiel pour fournir et rechercher de l'information, conduisant à une forte accumulation de données. La recherche sur Internet dans sa forme présente devient vite exaspérant car les données disponibles peuvent être superficielles et de formes très diverses. Les utilisateurs du Web en ont assez d'obtenir des ensembles gigantesques de réponses à leurs requêtes simples, ce qui les oblige à investir de plus en plus de temps à analyser les résultats à cause de leur grand nombre. Et alors de nombreux résultats s'avèrent non pertinents et les liens les plus intéressants restent en dehors de l'ensemble des résultats.

Le Chapitre1 introduit la motivation de notre travail de recherche. L'une des principales explications concernant la difficulté à effectuer une recherche d'information efficace est que les ressources existantes sur le web sont exprimées sous une forme destinée à la compréhension humaine. En d'autres termes, ces données deviennent vite inutilisables et inexploitables par la machine et l'intervention humaine s'avère être nécessaire pour obtenir de bon résultats. Ainsi, l'un des principaux challenges envisagé par les utilisateurs du web, tel que les fournisseurs et les utilisateurs de données, est d'imaginer des outils intelligents ainsi que des théories autour de la représentation et le traitement des connaissances dans le but de créer des données exploitables par la machine.

Le Chapitre 2 évalue et étudie les méthodes existantes et leurs limitations. Beaucoup de chercheurs ont déjà travaillé dans cette voie. La création du Web sémantique, basé sur le concept d'ontologie permet de rendre les données compréhensible par la machine. Le Web sémantique constitue l'une des solutions les plus intéressantes proposée par la communauté des chercheurs. L'objectif est de proposer une représentation intelligente des données qui soit exploitable par la machine. En d'autres termes, cette représentation doit lui permettre d'avoir une meilleure « compréhension » des documents et d'améliorer ainsi la qualité de la recherche parmi l'information existante. L'accent est mis sur la réflexion nécessaire à la construction de la signification du concept relié aux réseaux pour la représentation des connaissances. L'idée est de tendre vers la production semi-automatique voire complètement automatique de résultats de grande qualité. Autrement dit, l'objectif est de minimiser l'intervention humaine est de maximiser la qualité des résultats obtenus. Récemment, le développement d'ontologies a gagné rapidement l'attention d'un grand nombre de chercheurs à travers le monde. Aussi, il n'existe pas de réel consensus sur la définition du concept d'ontologie.

Le chapitre 3 présente la plate-forme ToxNuc-E et le positionnement de notre recherche autour de cette plate-forme. Etant donné l'importance pratique et théorique du développement d'ontologies, il n'est pas surprenant de retrouver un grand nombre de chercheurs, fervents et engagés dans ce domaine de recherche. Dans le cadre de notre travail de recherche nous proposons l'approche, dite ESN (« *Extended Semantic Network* ») qui représente une approche innovante dans le domaine de la représentation des connaissances et des ontologies. Contrairement aux approches classiques, basées sur les mots clés, l'approche ESN consiste à construire des réseaux en recherchant des ensembles d'associations entre les nœuds sémantiques et les relations de proximité sur TocNuc-E.

Le Chapitre 4 précise le concept de Réseau de modélisation proximale, généré par des modèles mathématiques. L'idée de base de ESN est de trouver une représentation des connaissances et une méthode de construction d'ontologies qui soit efficace afin de surmonter les contraintes existantes inhérentes à la recherche d'information et aux problèmes de classification. Notre approche se décompose suivant deux phases. La première phase consiste à traiter une grande quantité d'information textuelle en utilisant des modèles mathématiques pour automatiser la construction d'ontologies évolutives. Cette phase de notre proposition donne comme résultat un réseau de mots. Celui-ci est calculé en utilisant des outils mathématiques venant de l'analyse de données et la classification automatique. Ainsi, la création d’un réseau de proximité repose alors sur la proximité des mots dans un document.

Le chapitre 5 étudie la modélisation des réseaux sémantiques et introduit un modèle de conception proposé par nous pour permettre efficace coût efficacité de la conception. Le Réseau sémantique est essentiellement un graphe orienté étiqueté permettant l'utilisation de règles génériques, de l'héritage, et de la représentation orientée objet. Il est souvent utilisé comme une forme de représentation des connaissances, où les concepts représentés par les noeuds sont connectés l'un à l'autre en utilisant les liens relationnels représentés par des arcs. Le Réseau sémantique est construit avec l'aide d'experts de la connaissance et la compréhension d'un domaine. Il est donc principalement construit par les hommes avec une très bonne précision.

Le Chapitre 6 détaille le réseau sémantique étendu (Extended Semantic Network). La deuxième phase consiste à examiner attentivement et de manière efficace les différentes possibilités d'intégrer les informations obtenues à partir de notre modèle mathématique et à partir du modèle cognitif développé manuellement. Cette phase se base sur une méthode heuristique développée dans l’extension des réseaux et utilisant les résultats de la méthode mathématique. Cette phase se termine en considérant le modèle humain (développé manuellement) comme le point d’entré de notre réseau de concepts.

L’idée principale est de développer une approche novatrice combinant les caractéristiques humaines et la théorie des concepts utilisée par la machine. Les résultats peuvent présenter un grand intérêt dans différents champs de recherche tels que la représentation des connaissances, la classification, l'extraction, le filtrage des données ainsi que dans la recherche sur le développement d’ontologies. Dans le cadre de ce travail de thèse, nous avons discuté et nous avons mis en lumière des méthodes novatrices concernant le traitement et l’intégration d'information. Cette recherche présente une nouvelle méthode de travail de collaboration s’appliquant particulièrement au contexte de la représentation des connaissances et à la recherche d’information.

Le chapitre 7 illustre quelques des expériences réalisées à l'aide de notre réseau sémantique étendu et ouvre des orientations pour les perspectives d'avenir. Les questions concernant la représentation des connaissances, la gestion, le partage et l'extraction d'information sont passionnantes et complexes. Cet attrait est en toute évidence essentiellement du aux rapports entre l'homme et la machine.

Le fait que nous essayons de combiner les résultats de deux aspects différents constitue l'une des caractéristiques les plus intéressantes de notre recherche actuelle. Notre proposition consiste à essayer de construire des ontologies de manière plus rapide et plus simple. L'avantage de notre méthodologie par rapport aux travaux précédent est que notre approche est novatrice par le fait qu'elle combine les calculs de la machine avec le raisonnement humain. Le réseau ainsi obtenu peut alors être utilisé par des outils comme par exemple, un classificateur de documents.

Nous considérons notre résultat comme étant structuré par l'esprit et calculé par la machine. L'une des principales perspectives serait de trouver le juste milieu pour combiner le concept de réseau sémantique avec le mot obtenu à partir du réseau de proximité. D'autres perspectives à ce travail de recherche seraient d'identifier cette combinaison entre les deux grandes méthodes et de mettre en place un benchmark afin de mesurer l'efficacité de notre prototype.

Table of Contents:

List of Figures:

1. Introduction

1.1. Machine intelligence: brief history

Mankind has long been curious about how the mind works and fascinated by intelligent machines. One can see people's desire to understand and even to create intelligence. With today's ever accelerating advances in science and technology, it is becoming increasingly achievable that we may soon gain a complete understanding of human intelligence and consciousness. Intelligence can be described as the computational part of the ability to achieve goals in the surrounding world. Varying levels and types of intelligence exist in all people, many animals and few machines.

Artificial Intelligence is one such concept, which defines the science and engineering of making intelligent machines, especially intelligent computer programs capable of understanding and imitating human intelligence. AI [McCarthy, 1963] is the area of computer science focusing in creating machines that can engage on behaviors that humans consider intelligent. With this understanding it seems reasonable to assume that it will then be possible to build artificial machines whose intelligence matches, and possibly even exceeds, that of humans. The ability to create intelligent machines has intrigued humans since ancient times and today with the advent of the computer and 50 years of research into AI programming techniques, the dream of smart machines is becoming a reality. Researchers are creating systems which can mimic human thought, understand speech, beat the best human chess player, and countless other feats never before possible.

It wasn't until the post-war period (1945-1956) that Artificial Intelligence would emerge as a widely-discussed field. What impelled the birth of Artificial Intelligence were the arrival of modern computer technology and arise of a critical mass. Researchers such as Marvin Minsky, John McCarthy [McCarthy, 1959], Allen Newell, and Herbert Simon led their students in defining the new and promising field. The development of the modern computer technology affected the AI research immensely. Although the computer provided the technology necessary for AI, it was not until the early 1950's that the link between human intelligence and machines was really observed. Many pioneers of AI broke away from the traditional approach of artificial neurons and decided that the human thought could be more efficiently emulated with modern digital computer.

The term artificial intelligence was first introduced in 1956, at the Dartmouth conference headed by John McCarthy regarded as the father of AI, and since then Artificial Intelligence has expanded because of the theories and principles developed by its dedicated researchers. Through its short modern history, advancement in the fields of AI have been slower than first estimated, progress continues to be made. Since its birth 4 decades ago, there have been a variety of AI programs, and they have constructively impacted other technological advancements. Researchers like Marvin Minsky, John McCarthy played very significant role in the development of AI. Marvin Minsky went a step further to declare that, there may come one day, when our nanotechnology may even make us immortal. The mid 60’s saw AI arrive in every field from Military operations to computer games. AI became the common goal of thousands of different studies. Researchers used various AI techniques and improved the capability of computers in pursuing various projects.

Various AI-related studies had developed into recognizable specialties during the 70’s. Researchers like Edward Feigenbaum pioneered the research on expert systems; Roger Schank promoted language analysis with a new way of interpreting the meaning of words; Marvin Minksy propelled the field of knowledge representation a step further with his new structures for representing mental constructs; Douglas Lenat explored automatic learning and the nature of heuristics; David Marr improved computer vision; the authors of PROLOG language presented a convenient higher language for AI researches. The specialization of AI in the 70's greatly strengthened the backbone of AI theories.

1.2. Research context

Since the past decade the World Wide Web (WWW) has played a pivotal role in information diffusion and sharing, leading to tremendous upsurge in information availability in the electronic form. For many people, the World Wide Web has become an essential means of providing and searching for information leading to large amount of data accumulation. Searching information on the web is soon becoming an infuriating experience due to the fact that the data available is both superfluous and diverse. Web users end up finding huge number of answers to their simple queries, consequentially investing more time in analyzing the output results due to its immenseness. Yet many results here turn out to be irrelevant and one can find some of the more interesting links left out from the result set.

Most of the existing machine models find it difficult to analyze information independently. One of the principal explanations for such a condition is the reason that majority of the existing data resources in its present form are designed for human comprehension. When using these data with machines, it becomes highly infeasible to obtain good results without human interventions at different levels. It becomes essential to involve human expertise to achieve dependable results.

1.3. Problems and objectives

One such widely accepted approach that represents information in a machine readable form is to build knowledge domain ontology that can be used in machine analysis. Several researches have been carried out in this direction and some of the interesting solutions proposed are the semantic web based ontology to facilitate data understanding by machines. The objective here is to intelligently represent data, enabling machines to effectively analyze and read existing information.

Nevertheless human involvement still largely remains for the simple fact that ontology design and development requires extensive domain knowledge provided by human experts. So how to find methods that can be both effective as well as productive? What is the method that will require minimum human support for development and functioning? Can an automated knowledge representation be developed based on mathematical models alone?

The constraints in our research domain require answers to questions such as:

- Information analysis and retrieval: It is very important to find a method that is most efficient in the present context for fast and efficient information retrieval to match the with the resource size.
- Developing knowledge representation techniques that do not require expert intervention, possible automated/semi-automated approaches.
- Replacing classical ontology models with automated models.

Our responses to these research problems are based on the past models and findings that suggest:

- Exploring innovative approach that can build machine understandable knowledge representation techniques involving minimal cost.
- Identifying methods that can combine different models to achieve a common goal of fast and efficient information analysis for retrieval and classification.

In response to the following research needs we propose a solution by:

- Exploiting different mathematical models and techniques that can propose easy and effective methods of knowledge representation.
- Replacing human developed ontology with automated word networks.

1.4. Our contribution

In response to the above constraints we propose to develop an innovative approach that combines the human modeling with automated word networks. We propose to develop word networks that require minimum human intervention which can eventually replace ontology. The model we propose is the extended semantic network which is a large word network developed by combining human expert knowledge used in our semantic network model with that of machine results in our proximal prototype. We advocate using automated models in knowledge representation techniques for effective analysis of large textual information.

Our proposal is to construct a network of concepts on similar lines of an ontology but using a method where minimal human intervention is required. We compare this to a semi-supervised ontology, representing certain qualities of ontology and this is later expatiated by adding the information obtained from the automatically developed proximal network. We recommend through our experiments that this method will produce similar output as any traditional ontology but will greatly decrease the construction time, attributed to its mathematically modeled extension method. Some of the major points we hope to achieve through our approach are:

- To exploit techniques based on theories of proximity that will enable automatic construction of knowledge networks.
- To make construction cost effective and productive by encouraging minimum human intervention.
- To avoid the difficulty involved in coordinating cooperation between experts and a way to avoid their disagreements.

1.5. Report plan:

This thesis report composes of 7 chapters. We begin by presenting our context of research and listing the existing approaches to tackle the problem. We then move on to introduce our methods and models and their roles in the context. The last part of the report basically details the prototypes developed using our methods and techniques and illustrate the experimental results that can persuade future work in the field.

Chapter1 Introduces our motivation behind the research: One of the principal explanations for the unsatisfactory condition in information retrieval is due to the reason that majority of the existing data resources in its present form are designed for human comprehension.

When using these data with machines, it becomes highly infeasible to obtain good results without human interventions at regular levels. So, one of the major challenges faced by the users as providers and consumers of web era is to imagine intelligent tools and theories in knowledge representation and processing for making the present data, machine understandable.

Chapter 2 evaluates and studies the existing methods and their short falls: Several researches have been carried out in enabling machines to understand data and some of the most interesting solutions proposed are the semantic web based ontology to incorporate data understanding by machines. The objective here is to intelligently represent data, enabling machines to better understand and enhance capture of existing information. Here the main emphasis is given to the thought for constructing meaning related concept networks for knowledge representation. Eventually the idea is to direct machines in providing output results of high quality with minimum or no human intervention. In recent years the development of ontology is fast gaining attention from various research groups across the globe. There are several definitions of ontology purely contingent on the application or task it is intended for.

Chapter 3 presents the platform ToxNuc-E and positioning of our research around this platform: Given the practical and theoretical importance of ontology development, it is not surprising to find a large number of enthusiastic and committed research groups in this field. Extended Semantic Network is one such innovative approach proposed by us for knowledge representation and ontology like network construction, which looks for sets of associations between nodes semantically and proximally. Our objective here is to achieve semi-supervised knowledge representation technique with good accuracy and minimum human intervention, using the heuristically developed information processing and integration methods. The main goal of our research is to find an approach for automatic knowledge representation that can eventually be used in classification and search algorithms of the platform ToxNuc-E.

Chapter 4 elaborates on the concept of Proximal Network modeling, generated by mathematical models: As stated earlier the basic idea of Extended Semantic Network is to identify an efficient knowledge representation and ontology construction method to overcome the existing constraints in information retrieval and classification problems. To realize this we put our ideas into practice via a two phase approach. The first phase consists in processing large amount of textual information using mathematical models to make our proposal of automatic ontology construction scalable. This phase of our proposal is carried out by realizing a network of words mathematically computed using different statistical and clustering algorithms. Thus creating a proximal network computationally developed, depending essentially on word proximity in documents. The proximal network is basically representing the recall part of our approach.

Chapter 5 investigates the semantic network modeling and introduces a design model proposed by us to enable efficient cost effective design: Semantic Network is basically a labeled, directed graph permitting the use of generic rules, inheritance, and object-oriented programming. It is often used as a form of knowledge representation where concepts represented by nodes are connected to one another using the relational links represented by arcs. Semantic network is constructed with the help of expert knowledge and understanding of a domain. Hence it is mainly a human constructed network with very good precision.

Chapter 6 in effect details the Extended Semantic Network: The second phase of our research mainly consists in examining carefully and efficiently the various possibilities of integrating information obtained from our mathematical model with that of the manually developed mind model. This phase is ensured by a heuristically developed method of network extension using the outputs from the mathematical approach. This is achieved by considering the manually developed semantic mind model as the entry point of our concept network.

Here, the primary idea is to develop an innovative approach obtained by combining the features of man and machine theory of concepts, whose results can be of enormous use in the latest knowledge representation, classification, retrieval, pattern matching and ontology development research fields. In this research work we illustrate the methods used by us for information processing and integration aimed at visualizing a novel method for knowledge representation and ontology construction.

Chapter 7 illustrates some of the experiments carried out using our Extended Semantic Network and opens directions for future perspectives: The question on knowledge representation, management, sharing and retrieval are both fascinating and complex, essentially with the co-emergence between man and machine. This research presents a novel collaborative working method, specifically in the context of knowledge representation and retrieval. The proposal is to attempt at making ontology construction faster and easier. The advantages of our methodology with respect to the previous work, is our innovative approach of integrating machine calculations with human reasoning abilities. The resulting network so obtained is later used in several tools ex: document classifier to illustrate our research approach.

We use the precise, non estimated results provided by human expertise in case of semantic network and then merge it with the machine calculated knowledge from proximal results. The fact that we try to combine results from two different aspects forms one of the most interesting features of our current research. We view our result as structured by mind and calculated by machines. One of the main future perspectives of this research is finding the right balance for combining the concept networks of semantic network with the word network obtained from the proximal network. Our future work would be to identify this accurate combination between the two vast methods and setting up a benchmark to measure our prototype efficiency.

We conclude our research report with the Bibliography and the List of Publication carried out during my PhD.

2. State of the art: knowledge representation, management and retrieval

2.1. Introduction

The initial developments in technology enabled creation of machines and robots that work for humans mostly on a physical capacity. There have been instances where machine robots were created to help humans in their day to day activities. Some have even proved to be very beneficial like the robots built to help disabled persons with their day to day activities thus making them more independent. Humans developing machines that can help men in physical tasks have been for more than few decades and one can find their trails from farming and agriculture to rockets and space stations. However with advancement in technology these machines are becoming more sophisticated with time and is believed to soon match if not exceed human intelligence.

Mining for valuable and applicable information from data was the task of our computers over the past fifty years in the aptly termed "Information Age". However, the focus of the 21st Century will be a shift from the computer as simply a provider of basic information. Computers of the near future will be designed to be used to extract knowledge from information. Rapid advancements in technology, increased volume and complexity, and the wide and easy access to information create a new demand for the computer. The main focus of humankind in the current century is to utilize technology for intellectual activities in the emerging knowledge Age.

The technologies of the current age are transitioning our focus from individual, isolated information systems and repositories to an expanded exchange and sharing of information in order to broaden the size and depth of knowledge available to individuals and activities. It is expected that by the year 2010, more than one trillion intelligent computing devices will be utilized in all aspects of the commercial environment.

One of the most promising and important area of research currently carried out is creating intelligent machines that can actually understand and interpret information like human beings. The concept of Intelligence is built upon four fundamental principles, which include: Data, Information, Knowledge, and Wisdom.

For this to be a reality it is very important that we develop appropriate techniques that will actually help computer programs to interpret information similar to humans. This requires that the information be made available to these machines in formats that can be interpreted by them. One such field which actually helps computer programs and automated machines to interpret and understand information is called knowledge modeling. This is a field of Artificial intelligence where different techniques and methods are developed for representing information in different forms that can be easily understood and interpreted by machines.

2.2. Knowledge modeling

Knowledge Capture and Modeling (KCM) – or in short Knowledge Modeling – is a cross disciplinary approach to capturing and modeling knowledge. Knowledge Modeling packages combinations of data or information into a reusable format for the purpose of preserving, improving, sharing, aggregating and processing knowledge to simulate intelligence [Mach et al., 1999].

Expanding beyond Knowledge-based Reasoning (KBS) and Case-based Reasoning (CBR) systems, Knowledge Modeling offers a shift from local proprietary solutions to actually produce and circulate embedded knowledge models into larger computational solutions in an effort to create applied knowledge. Applied knowledge is very important to the immerging age of knowledge and information that it contributes to scores of intellectual activities, from continuous improvement to automated decision-making or problem-solving, and hence increases intellectual capital for generations of humankind to come.

The fundamental goal of KCM is to bring methodologies and technologies together in an implementation neutral framework as a practical solution for maximizing the influence of knowledge. The core difference between working with information and knowledge is that in addition to facts that information provides, a knowledge model includes enactment of sense or meaning and has the ability to subjectivity of experts and/or users [Makhfi, 2003].

As stated in his work by Makhfi, in everyday situations, people make a variety of decisions to act upon. In turn, these decisions vary based on one's preferences, objectives and habits. The following example – Situational Effects, highlights how gender and age play a role in the decision-making process.

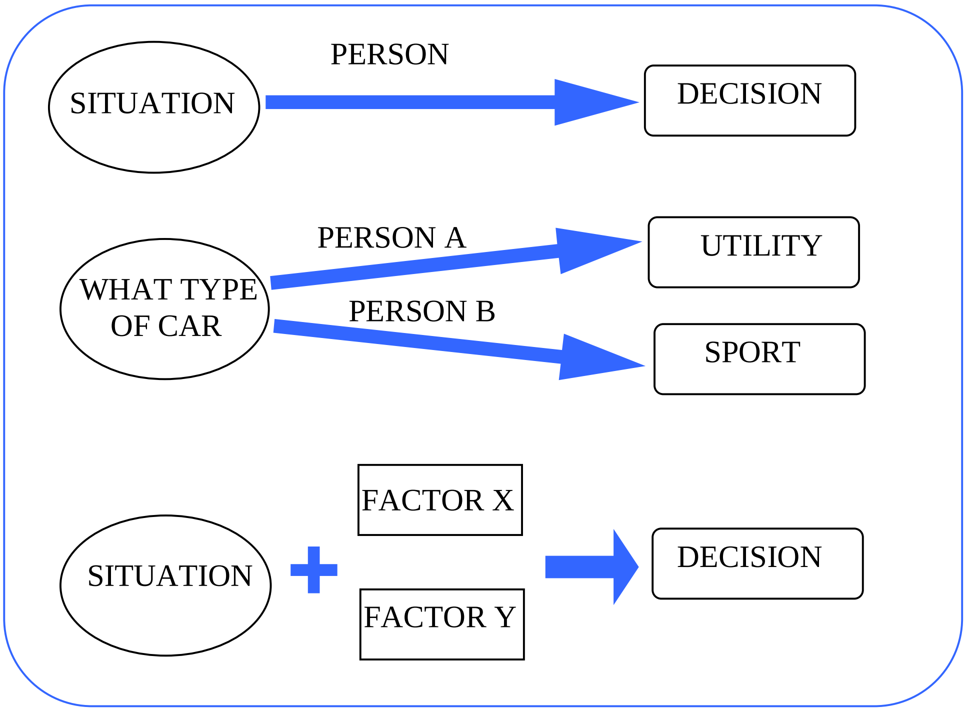


Figure 1: Situational effects

As such, many models, like the example of Person A (Female) and Person B(Male), can only be executed after having a profile assigned. A profile is defined as the personnel interpretation of inputs to a model. KCM incorporate the quantitative and qualitative use of information, and processes tangible and intangible attributes that contribute to end result, such as Person B's decision of buy a sports car. The bridging together of quantitative and qualitative methods enables KCM to incorporate subjectivity, which is the main differentiator between information and knowledge.

Each model can have data, information or outputs from other models as input. As such, models can be chained, nested or aggregated. For consistency all inputs to a model are considered as “information”. As such the output of a model would be referred to as information, when used as input to another model.

Among its benefits, a Knowledge Model has the ability to be constantly monitored and improved. Furthermore, Knowledge Models help us to learn from past decisions, to assess present activities and, just as important, to preserve domain expertise. KCM saves time and overhead costs, and reduces the mistakes from overlooks. Knowledge Models are very valuable and often outlive a particular implementation and/or project. Accordingly, the challenge of KCM is that this process must be designed not only as an abstract idea, but as an implementable process with the ability to aggregate and disseminate applied knowledge for the purpose of creating intellectual capital for generations of humankind to come.

2.3. What is knowledge representation?

As a basic definition knowledge representation can be defined as a subject in cognitive science as well as in artificial intelligence. In cognitive science knowledge representation is largely concerned with how people store and process information. However, in artificial intelligence mainly under knowledge modeling it is a way to store knowledge so that programs can process it and use it for example to support computer-aided design or to emulate human intelligence.

There are representation techniques such as frames, rules and semantic networks which have originated from theories of human information processing. Since knowledge is used to achieve intelligent behavior, the fundamental goal of knowledge representation is to represent knowledge in a manner as to facilitate drawing inference from knowledge.

Some issues that arise in knowledge representation from an AI perspective are questions like:

- How do people represent knowledge?
- What is the nature of knowledge and how do we represent it?

- Should a representation scheme deal with a particular domain or should it be general purpose?
- How expressive is a representation model or language?

There has been very little top-down discussion of the knowledge representation issues and research in this area is well aged. There are well known problems such as spreading activation where one faces problems in navigating a network of nodes, subsumption concerned with selective inheritance; for example an ATV can be thought of as a specialization of a car but it inherits only particular characteristics and classification problems like a tomato could be classified both as a fruit and a vegetable. In the field of artificial intelligence, problem solving can be simplified by an appropriate choice of knowledge representation and representing knowledge in some ways makes certain problems easier to solve.

2.3.1. History - knowledge representation (KR)

In computer science, particularly artificial intelligence, a number of representations have been devised to structure information. Knowledge representation is most commonly used to refer to representations that are intended to be processed by modern computers, and in particular, for representations consisting of explicit objects, and of assertions or claims about them. Representing knowledge in such explicit form enables computers to draw conclusions from knowledge already stored.

Many KR methods were tried in the 1970s and early 1980s, such as heuristic question-answering, neural networks, theorem proving, and expert systems, with varying success. However, medical diagnosis was a major application area, as were games such as chess. In the 1980s formal computer knowledge representation languages and systems arose. Major projects attempted to encode wide bodies of general knowledge [Ramachandran et al, 2005]; for example the Cyc project went through a large encyclopedia, encoding not the information itself, but the information a reader would need in order to understand the encyclopedia: naive physics; notions of time, causality, motivation; commonplace objects and classes of objects. The Cyc project is managed by Cycorp, Inc.; much but not all of the data is now freely available.

Through such work, the difficulty of KR came to be better appreciated. In computational linguistics, meanwhile, much larger databases of language information were being built, and these, along with great increases in computer speed and capacity, made deeper KR more feasible. Several programming languages have been developed that are oriented to KR. Prolog developed in 1972 [Michael et al., 1996] and [Bratko, 2000] but popularized much later, represents propositions and basic logic, and can derive conclusions from known premises. KL-ONE (1980s) [Brachman and Schmolze, 1985] is more specifically aimed at knowledge representation itself.

In the electronic document world, languages were being developed to represent the structure of documents more explicitly, such as SGML and later XML. These facilitated information retrieval and data mining efforts, which have in recent years begun to relate to KR. The Web community is now especially interested in the Semantic Web, in which XML-based KR languages such as RDF, Topic Maps, Gellish English [Van Renssen, 2005] and others can be used to make KR information available to Web systems.

2.3.2. Topics in Knowledge Representation

2.3.2.1. Language and notation

Many researchers think it would be best to represent knowledge in the same way that it is represented in the human mind, which is the only known working intelligence so far, or to represent knowledge in the form of human language. Richard L. Ballard [Ballard, 2004], for example, has developed a theory-based semantics system that is language independent, which claims to capture and reason with the same concepts and theory as people. The formula underlying theory-based semantics is:

Knowledge = Theory + Information

Most of the conventional applications and database systems are language-based. Unfortunately, we don't know how knowledge is represented in the human mind, or how to manipulate human languages the same way that the human mind does it. One clue is that primates know how to use point and click user interfaces; thus the gesture-based interface appears to be part of our cognitive apparatus, a modality which is not tied to verbal language, and which exists in other animals besides humans.

For this reason, various artificial languages and notations have been proposed for representing knowledge. They are typically based on logic and mathematics, and have easily parsed grammars to ease machine processing. They usually fall into the broad domain of ontologies.

2.3.2.2. Ontology languages

Most of the ontology languages developed are declarative languages, and are either frame languages, or are based on first-order logic. Most of these languages only define an upper ontology with generic concepts, whereas the domain concepts are not part of the language definition. Gellish English is an example of an ontological language that includes a full engineering English Dictionary.

Gellish English is a variant of Gellish and is a formal language, which means that it is structured and formalized subset of natural English that is computer interpretable. Its definition includes an English dictionary of concepts that is arranged in a taxonomy and that is extended into an ontology. From an information technology perspective Gellish English is a standard data model for information modeling and for knowledge representation. It is a data exchange language for the Semantic Web and can be used as a successor of electronic data interchange technologies. In principle, for every natural language there is a variant that is specific for that language. For example, Gellish Dutch (Gellish Nederlands), Gellish German (Gellish Deutsch), etc.

2.3.2.3. Knowledge representation languages

XML

The Extensible Markup Language (XML) is a general-purpose specification for creating custom markup languages. It is classified as an extensible language because it allows its users to define their own elements. Its primary purpose is to help information systems share structured data, particularly via the Internet, and it is used both to encode documents and to serialize data.

It is basically a meta-language proposed by the W3C that permits the representation of a text document as a tree using a marking system. This language was developed to simplify the exchange, sharing and publication of data on the web. Majority of the languages and models used in semantic web are expressed in XML.

XML makes it possible to structure a document defining their own tags according to the needs and without considering the significance this structure holds to the computer systems that will use it. The standards like XPath [Clark, 99] et XQuery [Boag, 2004] were developed after considerable research on a tree representation of the XML document, that provides the ability to navigate around the tree, selecting nodes by a variety of criteria.

RDF/RDFS

The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications, originally designed as a metadata data model, which has come to be used as a general method of modeling information through a variety of syntax formats.

The RDF metadata model is based upon the idea of making statements about Web resources in the form of subject-predicate-object expressions, called triples in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. For example, one way to represent the notion "The sky has the color blue" in RDF is as the triple: a subject denoting "the sky", a predicate denoting "has the color", and an object denoting "blue". RDF is an abstract model with several serialization formats (i.e., file formats), and so the particular way in which a resource or triple is encoded varies from format to format.

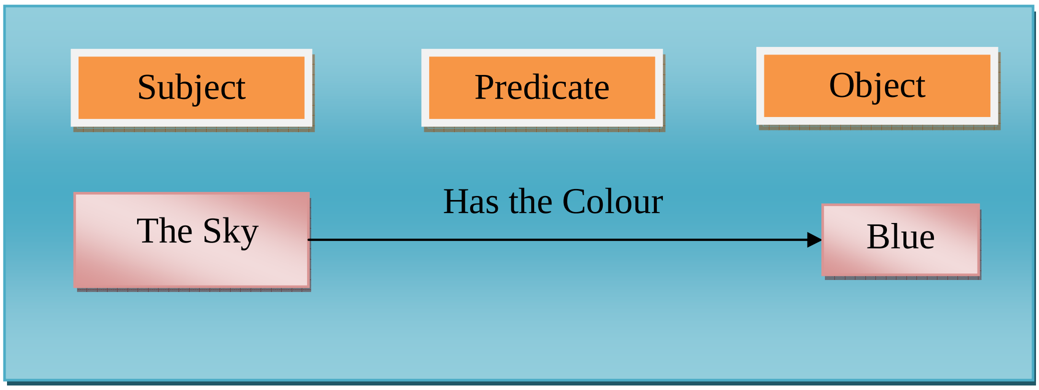


Figure 2: Example showing an RDF model

This mechanism for describing resources is a major component in what is proposed by the W3C's Semantic Web activity: an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the Web, in turn enabling users to deal with the information with greater efficiency and certainty.

RDFS [Lassila and Swick, 1999] is a meta model proposed by the W3C. RDFS or RDF Schema is an extensible knowledge representation language, providing basic elements for the description of ontologies, otherwise called RDF vocabularies, intended to structure RDF resources. The first version was published by W3C in April 1998, and the final W3C recommendation was released in February 2004. Main RDFS components are included in the more expressive language OWL. The main constructs are as follows:

rdfs:Class allows to declare a resource as a class for other resources. Typical example of an rdfs:Class is foaf:Person in the FOAF vocabulary. An instance of foaf:Person is a resource linked to the class using an rdf:type predicate, such as in the following formal expression of the natural language sentence : 'John is a Person'. “ex:John rdf:type foaf:Person”, The definition of rdfs:Class is recursive: rdfs:Class is the rdfs:Class of any rdfs:Class.

rdfs:subClassOf allows to declare hierarchies of classes. For example, the following declares that 'Every Person is an Agent': “foaf:Person rdfs:subClassOf foaf:Agent”. Hierarchies of classes support inheritance of a property domain and range (from a class to its subclasses.

DAML+OIL

DAML+OIL is a tag language for representing ontology. DAML+OIL is a successor language to DAML and OIL [Fensel et al, 2001] that combines features of both the parent languages. DAML stands for DARPA Agent Markup Language [DAML, 2000], and DARPA in turn stands for Defense Advanced Research Projects Agency and is the central research and development organization for the Department of Defense. The DAML program ended in early 2006. OIL stands for Ontology Inference Layer or Ontology Interchange Language. DAML+OIL build on the languages RDF and RDF Schema by enriching it with new primitives. One can generally refer to DAML+OIL as a very expressive logical description language. The expressiveness of the language is determined by the types of supported constructors which permit the definition of classes and the properties of its axioms.

OWL

The W3C was looking to propose a standard known as the OWL a web ontology language [Dean et al, 2003], derived from the DAML+OIL [Horrocks et al, 2001], a language built based on description logic is a family of knowledge representation languages for authoring ontologies, and is endorsed by the World Wide Web Consortium. This family of languages is based on two (largely, but not entirely, compatible) semantics: OWL DL and OWL Lite semantics are based on Description Logics, which have attractive and well-understood computational properties, while OWL Full uses a novel semantic model intended to provide compatibility with RDF Schema. OWL ontologies are most commonly serialized using RDF/XML syntax. OWL is considered one of the fundamental technologies underpinning the Semantic Web.

The data described by OWL ontology is interpreted as a set of "individuals" and a set of "property assertions" which relate these individuals to each other. An OWL ontology consists of a set of axioms which place constraints on sets of individuals (called "classes") and the types of relationships permitted between them. These axioms provide semantics by allowing systems to infer additional information based on the data explicitly provided. For example, an ontology describing families might include axioms stating that a "hasMother" property is only present between two individuals when "hasParent" is also present, and individuals of class "HasTypeOBlood" are never related via "hasParent" to members of the "HasTypeABBlood" class. If it is stated that the individual Harriet is related via "hasMother" to the individual Sue, and that Harriet is a member of the "HasTypeOBlood" class, then it can be inferred that Sue is not a member of "HasTypeABBlood".

Some existing OWL ontologies may be browsed using an editor such as Protégé-OWL to edit the ontologies posted at the Protégé web site. There is a large collection of biomedical ontologies available through the OBO Foundry, which are available on their download page, as well a number of others hosted at the NCBO BioPortal. Other ontologies can be found by searching for appropriate search terms with the filetype set to ".owl" or ".rdf" or by using the Swoogle semantic web search engine.

2.3.2.4. Links and structures

While hyperlinks have come into widespread use, the closely related semantic link is not yet widely used. The mathematical table has been used since Babylonian times. More recently, these tables have been used to represent the outcomes of logic operations, such as truth tables, which were used to study and model Boolean logic, for example. Spreadsheets are yet another tabular representation of knowledge. Other knowledge representations are trees, by means of which the connections among fundamental concepts and derivative concepts can be shown.

Visual representations, called a plex as developed by The Brain Technologies are relatively new in the field of knowledge management but give the user a way to visualize how one thought or idea is connected to other ideas enabling the possibility of moving from one thought to another in order to locate required information. The approach is not without its competitors [Amaravadi, C. S, 2005].

2.3.2.5. Notation

The recent fashion in knowledge representation languages is to use XML as the low-level syntax. This tends to make the output of these KR languages easy for machines to parse, at the expense of human readability and often space-efficiency.

First-order predicate calculus is commonly used as a mathematical basis for these systems, to avoid excessive complexity. However, even simple systems based on this simple logic can be used to represent data that is well beyond the processing capability of current computer systems

Examples of notations:

- DATR is an example for representing lexical knowledge
- RDF is a simple notation for representing relationships between and among objects

For the semantic web, one of the most important aspects is the ability to manipulate the semantic information of a web document such that the machines are able to understand the semantic data of the document. The notations normally describe the actual contents of the document by associating it with semantic descriptions. One could consider this more like the meta-data of documents.

2.3.2.6. Storage and manipulation

One problem in knowledge representation consists of how to store and manipulate knowledge in an information system in a formal way so that it may be used by mechanisms to accomplish a given task. Examples of applications are expert systems, machine translation systems, computer-aided maintenance systems and information retrieval systems (including database front-ends).

Semantic networks may be used to represent knowledge. Each node represents a concept and arcs are used to define relations between the concepts. One of the most expressive and comprehensively described knowledge representation paradigms along the lines of semantic networks is MultiNet (an acronym for Multilayered Extended Semantic Networks).

From the 1960s, the knowledge frame or just frame has been used. Each frame has its own name and a set of attributes, or slots which contain values; for instance, the frame for house might contain a color slot, number of floors slot, etc.

Using frames for expert systems is an application of object-oriented programming, with inheritance of features described by the "is-a" link. However, there has been no small amount of inconsistency in the usage of the "is-a" link: Ronald J. Brachman wrote a paper titled "What IS-A is and isn't" [Brachman, 1983], wherein 29 different semantics were found in projects whose knowledge representation schemes involved an "is-a" link. Other links include the "has-part" link.

Frame structures are well-suited for the representation of schematic knowledge and stereotypical cognitive patterns. The elements of such schematic patterns are weighted unequally, attributing higher weights to the more typical elements of a schema. A pattern is activated by certain expectations: If a person sees a big bird, he or she will classify it rather as a sea eagle than a golden eagle, assuming that his or her "sea-scheme" is currently activated and his "land-scheme" is not.

Frame representations are object-centered in the same sense as semantic networks are: All the facts and properties connected with a concept are located in one place - there is no need for costly search processes in the database. A behavioral script is a type of frame that describes what happens temporally; the usual example given is that of describing going to a restaurant. The steps include waiting to be seated, receiving a menu, ordering, etc. The different solutions can be arranged in a so-called semantic spectrum with respect to their semantic expressivity.

2.4. What is ontology

In a nutshell Ontologies can be defined as tools that allow to store domain knowledge in a much more sophisticated form than thesauri. We therefore assume that by using ontologies in information retrieval (IR) systems a significant gain in retrieval effectiveness can be measured. The better (more precise) an ontology models the application domain, the more gain is achieved in retrieval effectiveness. It is possible to diminish the negative effect of ontology imperfection on search results by combining different ontology-based heuristics during the search process which are immune against different kinds of ontology errors.

It is a well-known fact that there is a trade-off between algorithm complexity and performance. This insight is also true for ontologies: most of the ontology formalisms do not have tractable reasoning procedures. Still, the assumption is that by combining ontologies with traditional IR methods, it is possible to provide results with acceptable performance for real-world size document repositories.

Ontology in philosophy is defined as the study of being or existence and the basic categories and relationships involved, to determine what entities and what types of entities exist. Ontology is considered to be the most fundamental branch of metaphysics. Ontology thus has strong implications for conceptions of reality. However the world of computer science and information science uses its own jargon to define ontology as a formal representation of a set of concepts within a domain and the relationships expressed between those concepts. It is used to reason about the properties of that domain, and also be used to define the domain.

2.4.1. State of the art

Despite its fundamental importance, the accumulation of ontologies has only just begun. Techniques for organizing ontologies, combining smaller ontologies to form larger systems, and using this knowledge effectively are all in their infancy. There are few collections of ontologies in existence; almost all are still under development, and currently none of them are widely used.

Efforts are under way to create ontologies for a variety of central commonsense phenomena, including time, space, motion, process, and quantity. Research in qualitative reasoning has led to the creation of techniques for organizing large bodies of knowledge for engineering domains and automatic model-formulation algorithms that can select what subset of this knowledge is relevant for certain tasks [Jean et al, 2006]. Although these efforts are promising, they are only in the preliminary stages of development. The natural language community has invested in a different form of ontological development. WordNet [Rao, 2008], [Collins and Quillian, 1972] is a simple but comprehensive taxonomy of about 70,000 interrelated concepts that is being used in machine translation systems, health care applications, and World Wide Web interfaces.

Another important development has been the creation of easy-to-use tools for creating, evaluating, accessing, using, and maintaining reusable ontologies by both individuals and groups. The motivation is that ontology construction is difficult and time consuming and is a major barrier to the building of large-scale intelligent systems and software agents. Because many conceptualizations are intended to be useful for a wide variety of tasks, an important means of removing this barrier is to encode ontologies in a reusable form so that large portions of an ontology for a given application can be assembled from smaller ontologies, that are drawn from repositories. This work is also only in the preliminary stages of development.

2.4.2. Why is an ontology built?

Some of the most basic reasons [Natalya F Noy et al, 2001] for building an ontology are as follows:

- **Sharing common understanding of the structure of information** among humans or machines is one of the most primary goals in developing ontologies [Musen M.

A. 1992]; [Gruber 1993]. To make this concept clearer let us consider an example of several web sites containing travel information. If these Web sites share and publish the same underlying ontology of the terms they all use, then computer agents can extract and aggregate information from these different sites. The agents can use this aggregated information to answer user queries or as input data in other similar applications.

- **Enabling reuse of domain knowledge** was another important driving force behind recent surge in ontology research. For example, models for many different domains need to represent the notion of Date. This representation includes the notions of different date format, and so on. If one group of researchers develops such an ontology in detail, others can simply reuse it for their domains. Additionally, if we need to build a large ontology, we can integrate several existing ontologies describing portions of the large domain. Similarly one can also reuse a general ontology, and extend it to describe one's domain of interest.

- **Making explicit domain assumptions** underlying an implementation makes it possible to change these assumptions easily if our knowledge about the domain changes. In addition, explicit specifications of domain knowledge are useful for new users who must learn what terms in the domain mean.

- **Separating the domain knowledge** from the operational knowledge is another common use of ontologies. We can describe a task of configuring a product from its components according to a required specification and implement a program that does this configuration independent of the products and components themselves [McGuinness and Wright 1998]. We can then develop an ontology of PC-components and characteristics and apply the algorithm to configure made-to-order PCs [Rothenfluh et al. 1996].

- **Analyzing domain knowledge** is possible once a declarative specification of the terms is available. Formal analysis of terms is extremely valuable when both attempting to reuse existing ontologies and extending them [McGuinness et al. 2000].

Developing an ontology is similar to defining a set of data and their structure for other programs to use. Problem-solving methods, domain-independent applications, and software agents use ontologies and knowledge bases built from ontologies as data.

2.4.3. Ontology : definitions

There are several definitions describing an ontology, this section states few the most widely known definitions of ontology.

According to many dictionary definitions ontology can be defined as the science or study of being: specifically, a branch of metaphysics relating to the nature and relations of being; a particular system according to which problems of the nature of being are investigated; first philosophy. It is also stated as a theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system.

In modern philosophy, formal ontology has been developed in two principal ways. The first approach has been to study formal ontology as a part of ontology, and to analyze it using the tools and approach of formal logic: from this point of view formal ontology examines the logical features of predication and of the various theories of universals. The use of the specific paradigm of the set theory applied to predication, moreover, conditions its interpretation.

The second line of development returns to its Husserlian origins and analyses the fundamental categories of object, state of affairs, part, whole, and so forth, as well as the relations between parts and the whole and their laws of dependence - once all material concepts have been replaced by their correlative form concepts relative to the pure 'something'. This kind of analysis does not deal with the problem of the relationship between formal ontology and material ontology [Liliana Albertazzi, 1996].

A more widely know definition of ontology is by Gruber, where an ontology is defined as an explicit specification of a conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of existence. For knowledge-based systems, what “exists” is exactly that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge.

Thus, we can describe the ontology of a program by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names are meant to denote, and formal axioms that constrain the interpretation and well-formed use of these terms [Gruber, 1993].

In the philosophical sense, one may refer to an ontology as a particular system of categories accounting for a certain vision of the world. As such, this system does not depend on a particular language: Aristotle's ontology is always the same, independent of the language used to describe it. On the other hand, in its most prevalent use in AI, an ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words.

In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships; in more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation [Guarino 1998].

For Swartout [Swartout W. R. 1996], an ontology is a structured assembly of terms describing a field and, which can be used as a nucleus of a knowledge base.

Alternatively Bachimont [Bachimont, 2001] described ontology as an outcome of modelisation. According to him the aim of ontologies is to define which primitives, provided with their associated semantics, are necessary for knowledge representation in a given context.

All these different definitions provide diverse and complimentary views of an ontology mostly depending on the field (Artificial intelligence, Philisophy,) where ontology is used. Hence the three most widely accepted definitions of ontology based on their filed of use are as follows.

- Ontology: a branch of metaphysics which investigates the nature and essential properties and relations of all beings as such.

- ontology: a logical theory which gives an explicit, partial account of a conceptualization [Guarino and Giaretta, 1995] [Gruber, 1993]; the aim of ontologies is to define which primitives, provided with their associated semantics, are necessary for knowledge representation in a given context. [Bachimont, 2001]

- Formal ontology: the systematic, formal, axiomatic development of the logic of all forms and modes of being [Guarino and Giaretta, 1995].

2.4.4. Ontology classification

This section briefly comments on the classification of ontology proposed [Van Heijst et al., 1996]. They distinguish ontology based on two dimensions, which are as follows:

- The amount and type of structure of the conceptualization and
- The subject of the conceptualization.

The first category stating the amount and type of structure of the conceptualization mainly distinguishes 3 categories namely,

1. The ontology on terminology (lexical, glossaries etc)

2. The ontology on information (data base schema) and
3. The ontology for knowledge modeling.

Similarly, the second dimension of ontology called the subject of conceptualization is mainly categorized into the following four distinctions namely:

- Application ontologies: basically contains all the information necessary in developing a knowledge model for a particular application.

- Domain ontologies: provides an assembly of concepts and their relations describing the knowledge for a particular domain.

- Generic ontologies: cribbed by generic ontologies are more similar to domain ontologies, the only difference being that the concepts defined here are more of a generic in nature which actually describes the knowledge expressed by state, action, space and components. Generally the concepts of an ontology representing any domain are the specific concepts representing the domain, thus making it a specialized ontology for that specific domain.

- Representation ontologies: furnished primitive of formalization for knowledge representation. These are generally used representing domain ontologies. In this case, the underlying conceptualization addresses representation primitives, like those defined in Ontolingua's Frame Ontology [Gruber 1993]. Accordingly the representation ontology is therefore an example of meta-level ontology hence is sometimes called as Meta ontologies.

The drawbacks of the above distinctions are that the first dimension is far from being clear as it is hard to see how “information ontologies" can be considered as ontologies. This is because of the fact that one is not sure if a specification of the record structure of a database can be considered as an ontology according to the definition given by the authors, since it belongs to the symbol level.

A database schema can be seen as an ontology as long as it is a conceptual database schema, while a logical database schema belongs again to the symbol level. Considering this as an ontology would violate the distinction made by the authors between domain knowledge and domain ontology. Rather, what constitutes an ontology is the vocabulary used to describe, but this collapses into what have been called "terminological ontologies".

Consecutively, the distinction between terminological and knowledge-modeling ontologies is also not clear. Due to the problems of the information ontologies, the contrast between them and knowledge-modeling ontologies is misleading, and the meaning of the "richer internal structure" of the latter remains vague.

In conclusion, as stated by Guarino, there is no reason to hypothesize a distinction among ontologies on the basis of "the amount and type of structure of their conceptualization" [Guarino, 1997]. Maybe, as suggested, a distinction can be made among different ontologies on the basis of the degree of detail used to characterize a conceptualization.

A very detailed ontology gets closer to specifying the intended conceptualization (and therefore may be used to establish consensus about the utility of sharing a particular knowledge base which commits to that ontology), but it pays the price of a richer language. A very simple ontology, on the other hand, may be developed with particular inferences in mind, in order to be shared among users which already agree on the underlying conceptualization. Hence one may distinguish between reference ontologies and implemented (shareable) ontologies, or maybe off-line and on-line ontologies.

Very simple ontologies like lexicons can be kept on-line, while sophisticated theories accounting for the meaning of the terms used in a lexicon can be kept off-line. The second dimension is much clearer: depending on the subject of the conceptualization, the authors distinguish between application ontologies, domain ontologies, generic ontologies and representation ontologies.

2.4.5. Ontology construction and its life cycle process:

Ontologies are normally constructed to be utilized as components in software tools of different operational systems. Their development is similar to a life cycle of any software engineering tool. Particularly ontologies are considered as technical objects that can evolve and posses a life cycle worth describing.

Although there is some collective experience in developing and using ontologies, there is no field of ontological engineering comparable to knowledge engineering. In particular, as yet, there are no standardized methodologies for building ontologies. Such a methodology would include a set of stages that occur when building ontologies, guidelines and principles to assist in the different stages, and an ontology life-cycle which indicates the relationships among stages [Uschold et al., 1998]. The most well known ontology construction guidelines were developed by Gruber [Gruber, 1993], to encourage the development of more re-usable ontologies. Recently, there has been increased effort in trying to develop a comprehensive ontology methodology [Fernandez M. et al., 1997], [Gruninger and Fox, 1995], [Uschold and Gruninger, 1996].

These methodologies are broadly divided into those that are stage-based [Uschold and Gruninger, 1996] and those that rely on iterative evolving prototypes [Gomez-Perez, 1994]. These are in fact complementary techniques. Most distinguish between an informal stage, where the ontology is sketched out using either natural language descriptions or some diagram technique, and a formal stage where the ontology is encoded in a formal knowledge representation language that is machine computable. As an ontology should ideally be communicated to people and unambiguously interpreted by software, the informal representation helps the former and the formal the latter.

The stages involved in a life cycle of an ontology can be identified as listed below:

- **Identify purpose and scope:** This actually falls under the category where the administration of the project is involved. Developing a requirements specification for the ontology by identifying the intended scope and purpose of the ontology is the fundamental stage in the design process. A well-characterized requirements specification is important to the design, evaluation and re-use of an ontology.

- **Knowledge Acquisition:** mainly involves the process of acquiring domain knowledge from which the ontology will be built. Sources span the complete range of knowledge holders: Specialist biologists; database metadata; standard text books; research papers and other existing ontologies. Motivating scenarios are collected and informal competency questions formed [Uschold and Gruninger, 1996] - these are informal questions that the ontology developed must be able to answer and the one’s which will be used to check if the ontology is fit for purpose.

- **Conceptualization:** deals with identifying the key concepts that exist in the domain, their properties and the relationships that hold between them; identifying natural language terms to refer to such concepts, relations and attributes; and structuring domain knowledge into explicit conceptual models. This is the process where the concepts and relationships describing the domain are captured. The ontology is usually described using some informal terminology. Gruber [Gruber, 1993] suggests writing lists of the concepts to be contained within the ontology and exploring other ontologies to re-use all or part of their conceptualizations and terminologies. At this stage it is important to bear the results of the first step, that of requirements gathering, in mind.

- **Integrating:** is nothing but use or specialize an existing ontology: a task frequently hindered by the inadequate documentation of existing ontologies, notably their implicit assumptions. Using a generic ontology, gives a deeper definition of the concepts in the chosen domain.

- **Encoding:** mainly involves representing the conceptualization in some formal language, e.g. frames, object models or logic. This includes the creation of formal competency questions in terms of the terminological specification language chosen (usually first order logic).

- **Documentation:** it is clear that informal and formal complete definitions, assumptions and examples are essential to promote the appropriate use and re-use of an ontology. Documentation is important for defining, more expansively than is possible within the ontology, the exact meaning of terms within the ontology.

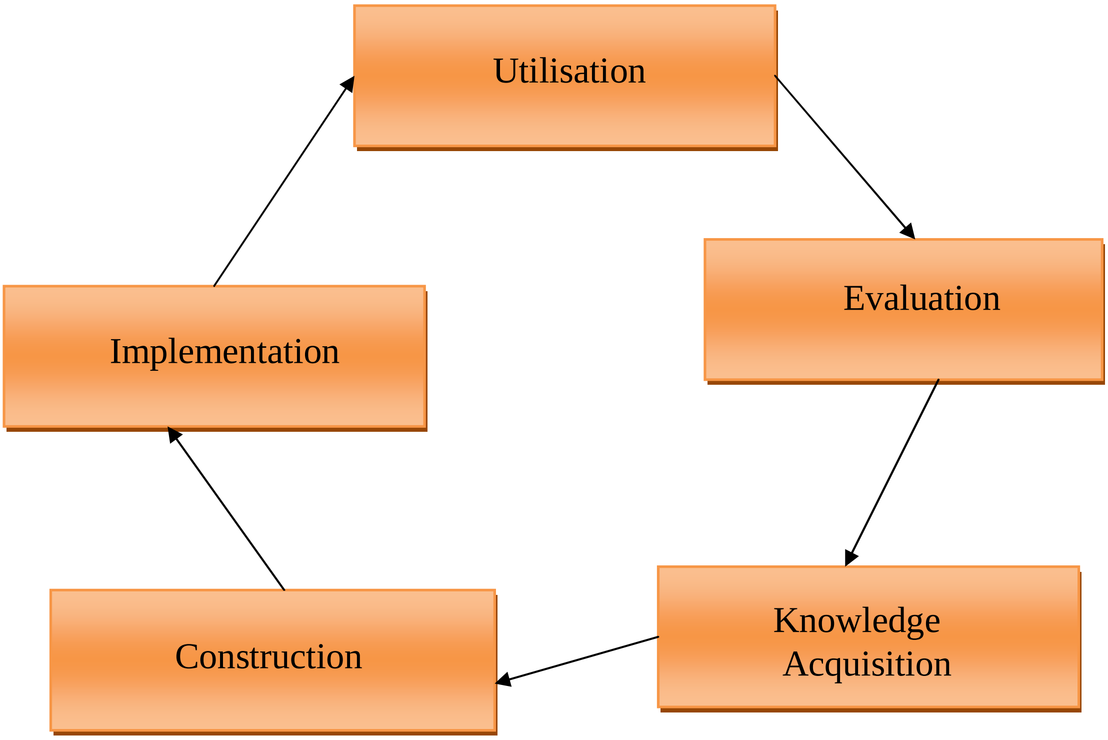


Figure 3: The ontology building life cycle

- **Evaluation:** is determining the appropriateness of an ontology for its intended application. Evaluation is done pragmatically, by assessing the competency of the ontology to satisfy the requirements of its application, including determining the consistency, completeness and conciseness of an ontology [Gomez-Perez, 1994]. Conciseness implies an absence of redundancy in the definitions of an ontology and an appropriate granularity**.**

Although there are different methodologies that can be followed for successfully constructing an ontology one is also aware of the fact that none of these methodologies developed so far actually cover all the proposed factors, principles and criteria required to construct an ontology. Nevertheless all the proposed methodology ensures that it does list the most important and majority of the existing processes such that its users are guided efficiently in the construction process. However, the most commonly referred methods is the Methontology of [Gomez-Perez, 1998] and [Fernandez et al., 1997] that establishes the stages through which the ontology moves during its life time and the activities to be performed in each stage.

The other methodologies are the enterprise ontology by [Uschold et King, 1995], a collection of terms and definitions relevant to business enterprises. The methodology used in building ontology for Tove project. Here the goal of the project is to develop a set of integrated ontologies for the modeling of both commercial and public enterprises. Some ontologies focus on how to bifurcate different stages of knowledge representation [Jasper and Uschold, 1999]. The method On-to-knowledge [Sure et al., 1999], was developed as a solution to the problem faced in developing web ontologies.

However, whatever the methodologie adopted but the process of constructing an ontology is mainly a collaboration that requires co-operation of domain knowledge experts, engineers and the future users of the ontology [Farquhar et al., 2000]. The environment for developing ontologies generally requires tools such as an ontology editor which is believed to help construction of ontology and also a knowledge representation language, providing basic elements for the description of ontologies like RDFS or OWL.

Protégé [Noy et al., 2001] is one of the most well known ontology editor, with an architecture that supports integration of pluggins thus permitting the editor to use new functionalities. Ontoedit [Sure et al., 2002] is another ontology editor on the line of On-to-knowledge. Webode [Arpirez et al., 2003] is based on client server environment and offers tools and functionalities that supports the complete life cycle of an ontology and operates based on the method Methontology.

2.5. Natural language processing

Natural language processing (NLP) is a subfield of artificial intelligence and computational linguistics. It studies the problems of automated generation and understanding of natural human languages. Natural-language-generation systems convert information from computer databases into normal-sounding human language. Natural-language-understanding systems convert samples of human language into more formal representations that are easier for computer programs to manipulate.

The goal of the Natural Language Processing (NLP) [Christopher et al., 1999] is to design and build software that will analyze, understand, and generate languages that humans use naturally, so that eventually one will be able to address computer as similarly as addressing another person.

This goal is not easy to reach. "Understanding" language means, among other things, knowing what concepts a word or phrase stands for and knowing how to link those concepts together in a meaningful way. It's ironic that natural language, the symbol system that is easiest for humans to learn and use, is hardest for a computer to master. Long after machines have proven capable of inverting large matrices with speed and grace, they still fail to master the basics of our spoken and written languages.

The challenges one faces stem from the highly ambiguous nature of natural language. As an English speaker you effortlessly understand a sentence like "Flying planes can be dangerous". Yet this sentence presents difficulties to a software program that lacks both your knowledge of the world and your experience with linguistic structures. Is the more plausible interpretation that the pilot is at risk, or that the danger is to people on the ground? Should "can" be analyzed as a verb or as a noun? Which of the many possible meanings of "plane" is relevant? Depending on context, "plane" could refer to, among other things, an airplane, a geometric object, or a woodworking tool. How much and what sort of context needs to be brought to bear on these questions in order to adequately disambiguate the sentence? [Bates, 1995].

In NLP these problems are addressed using a mix of knowledge-engineered and statistical/machine-learning techniques to disambiguate and respond to natural language input. In theory, natural-language processing is a very attractive method of human-computer interaction. Early systems such as SHRDLU, working in restricted "blocks worlds" with restricted vocabularies, worked extremely well, leading researchers to excessive optimism, which was soon lost when the systems were extended to more realistic situations with real-world ambiguity and complexity. Natural-language understanding sometimes referred to as an AI-complete problem require extensive knowledge about the outside world and the ability to manipulate it. The definition of "understanding" is one of the major problems in natural-language processing.

One example of knowledge representation method using Natural language processing techniques is Latent Semantic analysis (LSA). LSA is a technique in natural language processing, in particular in vectorial semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA [Landauer et al., 1998] can use a term-document matrix which describes the occurrences of terms in documents; it is a sparse matrix whose rows correspond to terms and whose columns correspond to documents, typically stemmed words that appear in the documents.

A typical example of the weighting of the elements of the matrix is tf-idf (term frequency–inverse document frequency): the element of the matrix is proportional to the number of times the terms appear in each document, where rare terms are upweighted to reflect their relative importance. This matrix is also common to standard semantic models, though it is not necessarily explicitly expressed as a matrix, since the mathematical properties of matrices are not always used. LSA transforms the occurrence matrix into a relation between the terms and some *concepts*, and a relation between those concepts and the documents. Thus the terms and documents are now indirectly related through the concepts.

3. ToxcNuc-E platform- a wide framework

3.1. Introduction

Collaboration literally means an action or a work completed in common with two or several persons. It is a group activity where individuals unite to form groups or unions with an intention to attain an objective. We can find evidence of group activity in many living beings ensuring early completion of tasks and better security against possible dangers. Every member in the group experiences better results when tasks are being accomplished in a co-ordinate group then attaining it individually.

This definition of collaboration is best defined in [Penalva and Commandre, 2006] where collaboration is defined as a hypothesis of collective intelligence relative to the capacity of a group of cognitive actors and artificial agents to attain a superior performance as compared to the addition of individual performances. Hence, one can conclude that in every group activity, self interest forms a deciding factor to motivate collaboration.

Collaboration and sharing is relatively a developing area in research introducing a methodology for the planned capture and re-use of organizational knowledge. Successful application of collaboration practices involves the understanding and constructive use of organizational learning and information flows within the organization. Co-operation and collaboration is becoming more important in the evolving context of global network, thus placing the user at the centre of a collective device [Shetty et al., 2006].

Collaborative work can either be of the nature, where each group member is involved in every activity with the work being highly interactive or where each group member is given an individual task.

In this chapter we introduce a collaborative platform called ToxNuc-E a brain child of CEA in collaboration with other research laboratories in France. The ToxNuc-E is a platform completely dedicated to research related to nuclear toxicology in living beings. This is a user friendly platform with built-in applications and features which not only guides the platform members in efficiently using the platform for information retrieval and management but also encourage the users to share information about their research activities with the rest of the registered members on the platform belonging to different communities.

3.2. Toxicologie nucléaire environnemental plateforme (ToxNuc-E)

A multi-field inciting program was set up to primarily stimulate the emergence of a community of experts and young researchers around a stake touching the public health and the environment. This program mainly handled the question of including/understanding the mechanisms of actions of heavy metals and radio nuclides on the various levels of organization of the living beings. Hence was aptly named Toxicologie Nuclear Environnemental Platforme. The research program in fundamental was a multi-field project which implied on a great number of researchers.

Some of the current activities in the platform are mainly concerning the disciplines such as biology, chemistry, medicine and physics. The main tasks of the committee of this program are to manage and provide all the necessary tools and applications for easy interaction among the vast community of researchers involved in the program. This will in turn favor and support communication leading to information exchange between actors (researchers) of the Program.

The Program first initiated in the year 2001 at the CEA (Commissariat à l'Energie Atomique), is now extended to four organizations of research partners: the CEA, CNRS, Inra and Inserm some of the well known research laboratories in France.

The idea of this platform raised from several questions on nuclear toxicology like: What are the effects on the living organisms from the elements such as the radio nuclides or heavy metals and metalloids used in medicine, research or for industrial activities? How a toxic element does reach its molecular target? Why certain cells of a body are more sensitive? How certain cells or organizations resist some of the elements which can be toxic?

The answer to these questions will actually make it possible to have a thorough knowledge of the impact of the entropic activities on human health and its environment. The “law of 91” on the nuclear waste, has in addition led to a certain number of technical solutions in the year 2006. But none of law asks for study of toxicology or impact of this waste on human health or the environment around them.

The recent studies dedicated to studying the extent of nuclear toxicology on living beings are very few and scattered in France and as well as abroad. Some of the field and methodologies used integrate very little the projections of the revolutionary techniques of genomic and biotechnologies. Contradictory to these observations, research in biology and genetics develops at a vertiginous speed and all the resources of post-genomic that are available to renew the field of toxicology are highly neglected in biology.

In order to contribute to this society and human health related questions, fresh impulse was given to this research within the framework of an inciting multi-field program titled “Nuclear Toxicology”. This program set up at the CEA was extended to the national community (CEA, CNRS, Inra and Inserm) over the period 2004-2007 and is entitled “Program Environmental-ToxNuc-E Nuclear Toxicology” [Ménager, 2004].

3.2.1. Scientific objectives of the program ToxNuc-E

It is a question of including/understanding the mechanisms of actions of heavy metals and radio nuclides on the various levels of organization from living organisms (molecular, cellular, bodies and fabrics, whole organizations) in order to propose preventive technical solutions, provisions of effective monitoring and solutions to decontaminate these elements distributed in certain compartments of the tropic chain.

The chemical elements were identified in dialogue with various actors involved in the nuclear die; in industry and in research, and a list of interesting elements was brought out. These elements are listed as follows: tritium, beryllium, boron, carbon, cobalt, selenium, strontium, technetium, cadmium, iodine, cesium, lead, uranium, plutonium, and americium. From the year 2004, zinc, copper and nickel were also included in the study list.

However, the state of the art on this domain was very weak to be used in helping the researchers in actually identifying these elements. This resulted in focusing the studies in two different fields called environmental toxicology and human toxicology. In these two fields, it is mainly a question of being interested: with the biological effects of these substances and the molecular and cellular mechanisms of transport, of toxicity and detoxification.

- In case of environmental toxicology, it helps to study the mechanisms of transfer of the geo-sphere towards the biosphere by the means of the bacteria and the plants and also to imagine applications to decontaminate the terrestrial or water environments.

- Similarly in case of human toxicology, it will be beneficial to imagine applications for treatment of contamination by targeting the studies on the elements uranium and plutonium. The organizations around which these studies are focused are preferentially those whose genome is sequenced i.e.: bacteria, yeast, arabidopsis, human cells, mouse, rats etc to name a few.

The above approach allows the massive use of the methods of genomic (transcriptome, protéome, métabolome).

3.2.2. The mobilization and the organization of the Program

The human means always does exist: it is only a question of mobilizing them based on some clearly defined scientific objectives. The CEA organized meetings to include some of the major researchers in the biological, chemical, physical and informative fields. Committees were organized and co-coordinators or heads for each research project were chosen and several researchers geographically dispersed were brought into contact through this platform.

The registered members on the platform ToxNuc-E were over 700 researchers from diverse fields, working on topics related to nuclear toxicology. In very short period vast information were collected on the platform. Once these tasks were completed the focus shifted on other problems like efficient data management, easy information retrieval and safety about confining one's research results to the other members of the platform only known professionally due to similar research interests were needed to be resolved. These solutions would automatically encouraged researchers on the platform to exchange information and discuss the research requirements and observation with other existing members of the community. Thus leading to a collaborative proceeding to resolve issues concerning to nuclear toxicology.

3.2.3. Development and evolution of ToxNuc-E program

The program committee and its management responsible for the first part of the program (2001-2003) identified twelve scientific projects to be selected for the period. Each of these projects controlled by one or more coordinators includes various specialists such as biologists, chemists, doctors, physicists, pharmacists. The program mobilizes 99 men per year of statutory personnel primarily CEA but also personnel from CNRS, Inra, Inserm which are the partners of of the CEA in the project. The program also finances 30 post doctorates and 15 doctorates.

These 150 men per year correspond in fact to more than 250 researchers established in several areas and concerning various operational directions of the CEA. These source data encouraged the members of the Management of Program to install communication and management tools making it possible to create a community around the Program.

Some of the tools developed by the program for its members are as follows:

The newsletter - the Letter of the Nuclear Program Toxicology is a monthly recto-back which is used as a bond between the researchers of the projects and allows a fast circulation of information useful to all. It is also an external tool of communication with the committee of the CEA and also with our scientific and industrial partners.

A cycle of continuous training - the people to be used in the existing research projects are the statutory doctorates, post-doctorates, technicians and researchers (CEA, CNRS, Inserm, Inra) intervening within the framework of the program of Nuclear Toxicology, in whole or part of their diversified initial formation, working time (biologists, chemists, physicists, pharmacists, doctors…) .

Some of the additional awaited development results of the action are as follows: integration of common concepts, improvement of the capacity of interaction between scientists resulting from various disciplines for the realization of the program. The trainees acquire a base of common knowledge in toxicology (general, target concepts of the poisons, experimental methods,); the biologists supplement their knowledge in chemistry and analytical chemistry; the chemists supplement their knowledge in cellular biology and molecular biology; the chemists and the biologists supplement their knowledge in: statistics applied to toxicology; epidemiology; physiology of target bodies to name a few.

3.2.4. Assessment and prospects

The scientific assessment of the period 2001-2003 is as follows: 79 publications with a factor of average impact of 4,17 including 6 publications with a factor of impact higher than 10 (average: 14,70); 4 articles of synthesis in referred works; 8 cards of synthesis summarizing the principal results by chemical element studied and 4 patents deposited. The human assessment is more difficult to quantify however a bringing together very Net between biologists and chemists, in the broad sense, is observed.

This for example led to the appropriation of the analytical step by the biologists and the taking into account of the complexity of the world of living organisms by the chemists. Without the detailed attention of all the scientific components, technical and administrative support of the CEA, this program could not have become a success. On this same set of themes, and with the same rigor in the selection of the projects and their follow-up, we will continue the adventure with our colleagues of CNRS, Inra and Inserm. The program "Environmental Nuclear Toxicology" was initiated in 2004 per one three years duration by selecting the best teams of the four organizations of research partners.

3.2.5. Building the collaborative platform

The reference frames of knowledge of the nuclear program Toxicology Nuclear Environmental are mainly developed and are managed by the LGI2P within the framework of URC EMA-CEA contract between the two labs.

The URC is a mixed structure created by the CEA and the School of the Mines of Alès, which carried out the technical development of platform using the content management system developed by LGI2P called the GSITE. Several platforms currently are supported by GSITE especially in the European networks of excellence or national networks.

A platform of collaborative work of the type "Reference frame of knowledge" is intended to help a scientific community to develop its collective processes: presentation of the researchers and the teams, presentation of the program, capitalization of information and results, shares knowledge, internal communication, filing of institutional documents, joint workspaces, forums of exchanges, specialized transport, diffusion of information to general public. Advanced functions of dynamic cartography of the contents (evolutionary trees and matrices) make it possible to follow the evolution of the data bases.

Each researcher registered in the program is a contributor authorized to deposit documents, to consult the filed documents, to communicate with the other researchers. A system of management of the confidentiality makes it possible to protect the diffusion of information to the centre among the existing research communities inside the platform.

A follow-up meeting and a report/ratio of different stages is maintained once in every six months. During these meetings, each project group writes a progress report and presents its results to the management of program. Following these meetings, the management of the program organizes a session of restitution on the advancement of all the projects involved.

3.2.6. Technical requirements on the platform

With every passing year the platform is experiencing a tremendous surge in the number of users registering on the platform thus leading to a growing database of information. This is mainly because the platform being one of its kinds provides a secure and common space for researchers to actually interact with the other existing researchers on the platform. It helps research groups geographically dispersed to actual connect with other similar research groups and thus exchange their research details and information through the platform.

It is currently assumed that this platform is one of the largest platforms dedicated to the research area of nuclear toxicology with a large database of information. The information are the data diffused by the users of the platform concerning their research groups, projects and the innovative developed in the domain.

However it is becoming very difficult to manage all the data that is being input into the platform by its existing and new users. This is for the simple reason that the amount of data flow on the platform in so high that it is clearly difficult to be managed by humans.

This is mainly because as and when the data flow increases it becomes necessary that more personals are employed to manage these data flows. This mainly leads to 2 major problems

- A steep increase in the costs involved in the program, as a result of employing more engineers to manually tackle the problem of data overflow in the platform.
- There is a high possible risk of data mismanagement while employing manual management of data. This is a very important concern especially when highly confidential and important information are involved.

Thus it became very evident that the platform employs tools and techniques to manage and tackle the information overflow. The committee responsible for the platform approached its collaborative partners LGI2P who were initially responsible of all the technical features of the platform. This is when our research team which was actually involved with a development of an innovative knowledge representation technique was associated with this platform.

As a member of this research team I was handed the task of analyzing and developing a prototype to be used on ToxNuc-E based on my research finding. The following chapters entail the detailed approach adopted by us to tackle this problem. The entire document sets and data used in the different stages of my research were provided by the ToxNuc-E platform. These documents were the most recent research activities and innovation/findings in the domain of nuclear toxicology. My phd are as listed below:

3.3. Graph editor

During the initial stages of my I realized the requirement of an application that can help us visualize results stored in tables of our database. Since my work predominantly dealt with modeling graph networks, it was very important that I visualize graphs constructed every time a new model is built. This basically helped me understand the developments as well as the requirements of my design model.

To address this requirement our research group decided to develop an editor application called graph editor which can be used for both visualization and editing purpose. The application was programmed using mainly the java and php coding. The functionality of the applications was primarily designed to support visualization of network structures and also in editing them where ever needed.

While initially designed largely for visualizing the constructed results, the graph editor also offered functionalities and features using which the user of the application would actually build his/her own graph network. The visualization of results was a very important parameter of my research in order to understand the different stages of the development of designs of the model. It also made the editing of results for testing very convenient. As per the users requirement one could easily make changes to the existing results and if the changes were a desirable result than the user could save these changes into database by erasing the previous entries.

The editing functionality of the graph editor became of enormous importance in the initial stages of the research work. This is mainly because of the fact that the initial research involved building some models along with experts from varying domains to develop concept networks. The graph editor made this task easy as it was very convenient to build new networks using this application as well as make changes to existing networks.

Below is the design window of the graph editor integrated into Toxnuc-e website:

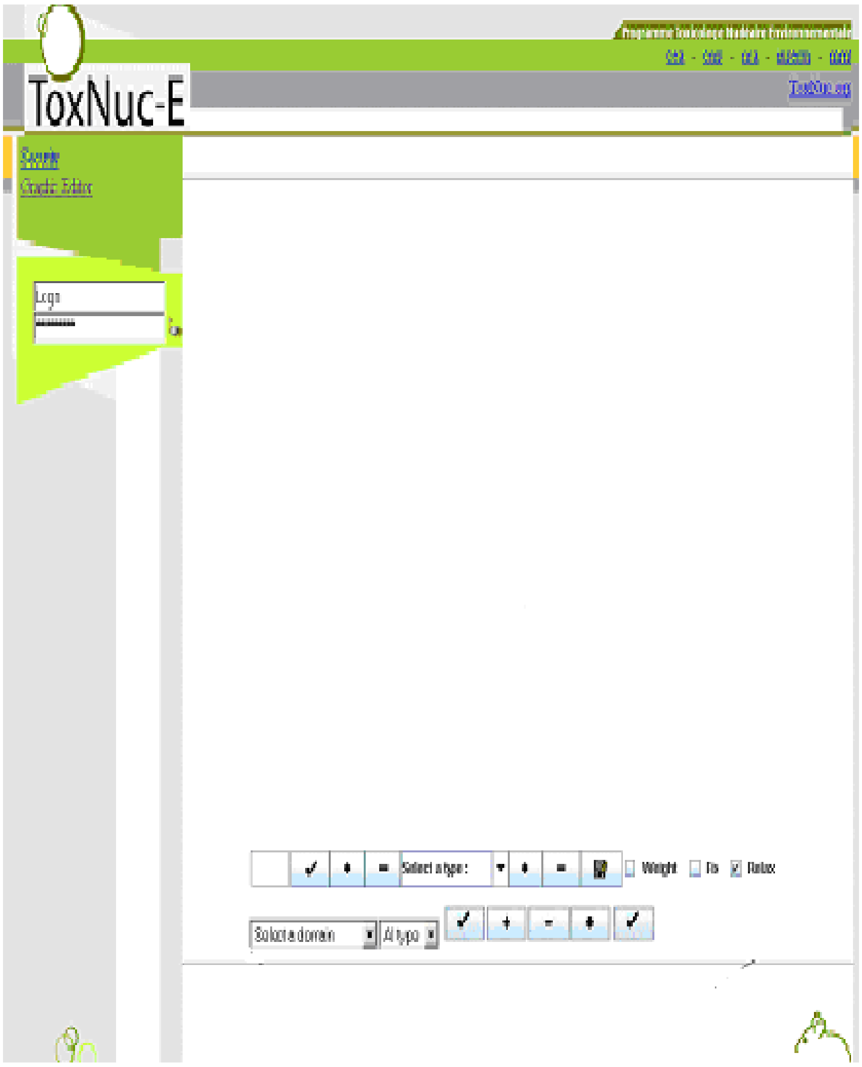


Figure 4: Snapshot of graph-editor ToxNuc-E platform

3.3.1. Design specifications

Graph editor is mainly designed using a main panel that supports the graphical visualization and a second panel that contains the tool bar with all the functionality buttons. This design was adopted mainly to separate the visual panel from the tool bar as the functionalities were constantly evolving and hence the tool bar required frequent redesigning.

The interface of this application is designed using mostly the features from java swing and all the functional features built in the graph editor tool are coded using the php language. The figure below shows the tool bar of the graph editor with its functional features.

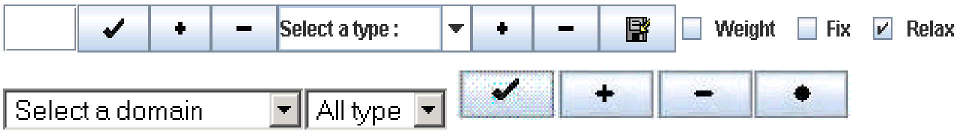


Figure 5: Graph editor tool bar

The buttons in the tool bar helps the application user to create and construct new graphs, visualize and edit existing graphs by saving any changes made on the existing graphs. It also contains functions that can be used to view graphs in different forms.

.

The second line of the tool bar helped the user of the graph editor to select a domain under which the user would like to build a new graph or view existing graphs. If the user decided to select an existing graph then the rest of the functionality buttons here helped the user carry out any required changes to the existing network.

In order to save space buttons are visualized using symbol language. All the buttons that necessary validates an action is represented by the symbol ✓. This symbol replaces the letter **OK** which is commonly used in many interfaces. Similarly the symbols + and − represents functionalities used to either add or delete nodes from a table respectively. Also the button with the symbol ● is used to identify the node which will be considered as the centre point of a network.

These buttons allow us to create new graphs using nodes and links. It also gives an option of saving the changes made by the user on to the database using the button for backup with a save symbol 💾 . The nodes created can also be deleted if the user in not satisfied with his creation. We can also drop tables from the database using this editor. This is done by selecting a particular table from the domain and then selecting the button with a subtraction symbol from the second line.

Similarly the first line of the tool bar is predominantly used to create new tables, firstly by naming the new table and then creating nodes and arcs under this table. The arcs to be drawn between nodes are chosen from the drop down menu chose a type which holds a set of relational links predefined by us. The types of links used here are similar to those already explained in the semantic network prototype. These links have been chosen to depict the actual relation between the nodes in the constructed graph. The symbols + and − next to the chose a type is used to add or remove an arc between any tow nodes of the network.

The check boxes shown □ Weight □ Fix ☑ Relax help the user in visualizing the weights of nodes; fixing the position of certain nodes and the relax button for relaxing the nodes thus making the network spread out for proper visualization.

The main programming of graph editor is done using the java language due to its simplicity in usage and integration with different platforms. Certain parts of the editor are programmed using php for easy communication with the php server.

Some of the main advantages we envisage using graph editor are:

- Easy and efficient visualization of any constructed prototype of our models. The visualization gives a clear picture of the design model as one is able to view a real time visual of a graph and thus easily identify fallouts.
- The second most important utilization of the graph editor is for building new graphs. This will greatly minimize the time and efforts used to build new networks, particularly in case of semantic networks where experts are involved. This feature also makes it very easy to make change to existing models.

Once the graph editor was developed we first integrated it into locally loaded version of the Toxnuc-e platform. This helped us greatly is analyzing our results during the development stages of our models.

4. Proximal network prototype

4.1. Introduction

Classifying documents and data is essential to the efficient management and retrieval of knowledge. The classical approach followed in document classifications are the methods where the classification is typically assigned to humans knowledgeable in the subject who actually read the entire document sets. In many large organizations, huge volumes of textual information are both created and examined, and some form of categorization of this textual information flow is always required. One major problem in document retrieval is of determining whether a document is relevant to the query. This determination is inherently inaccurate, since human experts can differ on their judgments with respect to the same document and query pair, even with the whole document available and a considerable range of background information on which to draw.

The document and query representations available to computer programs are less in quality; hence the results may be less precise. Nevertheless, the number of documents of potential interest to a human searcher far exceeds what one could hope to read. This has in a way helped to limit a search to relevant topics by assigning one or more subject codes from a prearranged list. One can find a large number of such classification systems available for document collections, excluding the fact that manually assigning these codes to documents is time consuming and expensive.

This chapter presents an alternative method that can be used for automatic classification of documents. The proposal is to use the proximity theory to estimate the existing proximal relation between documents. The idea is to develop a model that can actually calculate the commonality between the document contents by analyzing their proximal nature with one another. This relation between documents can then be used to classify them into different categories.

Effective machine-generated solutions would obviously increase efficiency and productivity. A computer can process information much faster than humans. With the explosion of electronically stored text, efficiency and productivity is of increasing importance. Beyond the immediate gains, however, is the great promise of enabling machines to analyze and examine free text and make correct decisions. We see a greater picture where such automated models can be incorporated with models based on human decision making, to form what we call hybrid models making efficient and productive knowledge representation feasible.

4.2. Understanding and definition

Proximity in general is defined as the distance between objects or entities in consideration. It can also be defined as the ability of a person or thing to tell when it is near an object, or when something is near it. This sense of proximity keeps us (living beings) from running into things in our everyday life. The above stated definition of proximity can also be applied when measuring the distance from one object to another object.

The simplest proximity calculations can be employed to calculate distance between entities, thereby avoiding a person away from things he can hit. Hence, proximity basically defines how far or near an entity is from/to another entity. The basic and the most important parameter in calculating proximity between two entities will be the measure of distance separating the entities.

The concept of proximity is largely used in medical fields to describe human anatomy with respect to positioning of organs. When the physical distance of internal organs is defined from one another they use proximity as the defining parameter. In these cases, one can state the distance of organs from each other by simply stating their proximity.

Alternatively, proximity can also be defined as closeness between two entities. In this case the parameter for closeness of the entities can either be the state, quality, sense or the fact of being physically near one another. This sense of being proximally closer to one another can be used in data analyses for processing and categorization of word entities in any given textual information. Here word entities can be grouped and or related based on their position and occurrence in any given textual data.

This categorization can be used to understand the relative proximity of these words in a given sentence, paragraph or even a document. This feature of proximity in analyzing textual information makes it possible to understand the positioning distance of the word entities occurring in the processed textual information. This will enable understanding the score of the relation it shares with the other word entities occurring in the same document or text.

The basic theory of proximity is concerned with the arrangement or categorization of entities that relate to one another. Proximity between entities is often believed to favor interactive learning, knowledge creation and innovation. It is necessary to understand why; when entities of a similar nature are grouped together the information becomes a unit.

This provides an observer with a clue to the concept you are communicating rather than being confronted with unrelated entities. When a number of entities are close in proximity a relationship is implied. If entities are logically positioned they connect to form a Structural Hierarchy.

The proximal prototype model is built based on the structural hierarchy characteristic of proximity where proximity between words, documents and information is used to for the logical positioning of words.

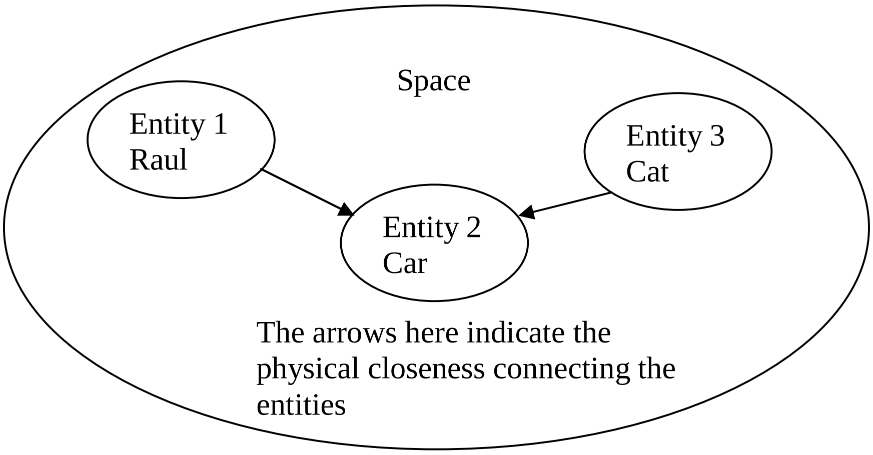

Space
Entity 1
Raul
Entity 3
Cat
Entity 2
Car
The arrows here indicate the
physical closeness connecting the
entities

Figure 6: Entities connected proximally

The proposed model is largely based on the distance aspect of proximity, where word entities are measured for closeness in a given space. In a text which reads “Raul owns a beautiful car. A white cat is on the car.”, one can easily identify that the word “Car” and “Cat” match best in terms of proximity. But the same word pair fails to make a match when seen in terms of sense or meaning, a Car and a Cat does not make the best word pair. This particular example clearly exemplifies the facet of proximity based on their occurrence in a given phrase or document.

The proximal prototype model built based on the concept of proximity between entities; is primarily used in analyzing huge amounts of textual data or information. The main focus of this model lies in optimizing the time required to analyze large amounts of textual information. Here the emphasis is on attaining quantity (recall) of information processed in a given time span. The proposed model enables a fast and efficient system for data analysis and representation. The model basically processes textual information for word entities and their proximity. Each word pair thus obtained during this processing is given a proximal value P.

The above information obtained from our proximal prototype model is then used to create a network of word pairs, called the proximal network. The proximal network is fundamentally a colossal word network in which word entities represent the nodes of the network and the proximal values they share between them represent the arcs joining these nodes (word entities) in the network.

The main functionality of this model as mentioned earlier lies in its ability to process huge amounts of textual data and create a list of processed word entities. These word entities are then fed into the mathematical programs which designate a proximity value for each word pair created by the model. The proximity parameter is a numerical value assigned to each word pair and is stored in the word matrix obtained as an output result from our model.

This parameter simply defines how close a word pair is in any given network after analyzing a given document. This feature of our model enables easy and quick processing of huge amounts of textual information in comparatively negligible time. The model thus enables efficient processing, retrieval and management of information from any set of documents.

4.3. Proximal prototype model

The past decade has witnessed tremendous upsurge in information availability in electronic form, leading to large amounts of data accumulation. The availability of too many information resources has made data management a cumbersome task for all companies and organizations. It has become very important for organizations to come up with innovative techniques to help them manage and maintain their ever growing information database. The major issue here is the difficulty faced in processing this huge database for specific information. It has become increasingly important to arrive with newer processing techniques with features to process information efficiently and more rapidly.

It is very important to understand that when processing for specific information from a pool of available information resources; it is very natural to have a higher quality of resource data when the information has been extracted by covering various different resource channels. Similarly it is also highly essential that when we search for data the processing agent should be able to identify most number of possible good information resources supporting our search. This attribute of recall is a very important factor to be considered while developing any processing technique for information retrieval.

The main concern that we try addressing here is the speed and recall factor, while processing large amount of information or data. We try to emphasize on the importance of the above factors in any information management and retrieval tool. These features will distinguishingly help develop processing tools which are capable of managing and retrieving information from vast data resources very efficiently without actually slowing down its speed.

Addressing these issues is our data processing technique called Proximal Prototype model. This model firstly aims at managing data by processing through large amounts of information quickly for efficient data management. The other issue that this model addresses in the recall factor to improve the quality of data retrieved using the technique. The objective here is to intelligently represent data, enabling machines to better understand and enhance capture of existing information.

Here the main emphasis is given to the thought for constructing closely related word networks for knowledge representation used for information management and retrieval using software programs. Eventually the idea is to direct machines in providing output results of high quality with minimum or no human intervention.

4.3.1. Architecture and design

The proximal prototype primarily helps in processing the information resources to form a network of word entities which can be used in information management and retrieval. The network of word entities thus formed called as proximal network is a completely automated network of words derived from the information resources fed into the proximal prototype model.

This network primarily represents the possible word pairs (extracted from the given information resource) and the proximity value of these word pairs. These word pairs are created by the statistical calculations that we employ during the various processing stages involved. Below is a block diagram of the proximal prototype model.

The information resources that are given as input to the proximal prototype model are basically a set of documents containing textual information. This model is designed to process data that are textual only and therefore cannot process any graphical images or designs. Although the prototype is built to process text format documents only, the actual input document given to the prototype in the initial stage at input can be of any format.

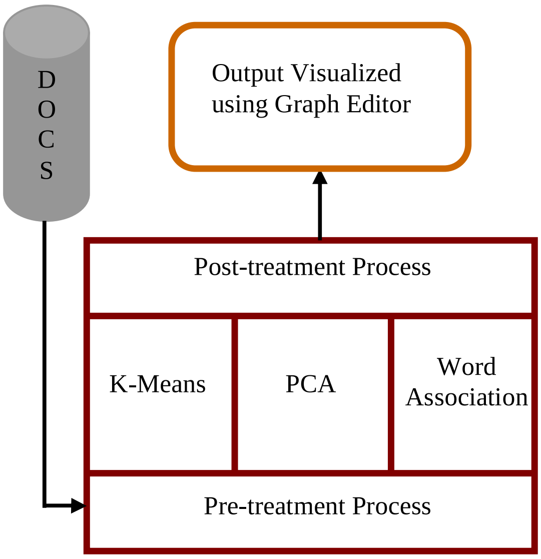


Figure 7: Block diagram representing proximal prototype model

This is because of the fact that all input documents will be automatically converted into text format by the external document converter included in the prototype before entering the processing stages that follow. The entire prototype model is built using java as the programming language.

Our proximal network model can be largely compared to Latent Semantic Analysis (LSA) model [Landauer, 1998] a technique in natural language processing. The LSA can use a term-document matrix which describes the occurrences of terms in documents; it is a sparse matrix whose rows correspond to terms and whose columns correspond to documents, typically stemmed words that appear in the documents. A typical example of the weighting of the elements of the matrix is tf-idf (term frequency–inverse document frequency): the element of the matrix is proportional to the number of times the terms appear in each document, where rare terms are upweighted to reflect their relative importance.

This matrix is also common to standard semantic models, though it is not necessarily explicitly expressed as a matrix, since the mathematical properties of matrices are not always used. LSA transforms the occurrence matrix into a relation between the terms and some concepts, and a relation between those concepts and the documents. Thus the terms and documents are now indirectly related through the concepts. LSA technique finds its primary use in:

- Compare the documents in the concept space (data clustering, document classification).
- Find similar documents across languages, after analyzing a base set of translated documents (cross language retrieval).
- Find relations between terms (synonymy and polysemy).
- Given a query of terms, translate it into the concept space, and find matching documents.

The Proximal prototype model is also built on a model very close to the LSA model. In the Proximal prototype model we have incorporated 3 important processing stages namely

1. Pre-treatment process
2. Mathematical modeling process and
3. Post treatment process

The output of the post treatment process is the word pair matrix with proximity value between them representing their closeness in a given space. This matrix is actually visualized using the graph editor model developed by us detailed under chapter 3.

The textual documents input to our proximal model can be of any size and format. These texts are considered as the information pool by the prototype. The prototype through its processing stages actually gathers and collects all the information to be identified and selected to form the proximal network. This is purely based on the proximity weight give to the links (detailed later in the chapter) connecting each word pair by our processing agents.

4.3.1.1. Pre-treatment process

This processing stage is mainly involved in preparing input documents for the mathematical processing stage. Here, in this process the input document is processed in several stages and an output of word frequency matrix is created with rows representing the words and columns representing the document name.

We basically carry out 2 different pre-treatment processes in the proximal prototype model namely the

1) Mots-creux pre-treatment process
2) Mots-pleins pre-treatment process

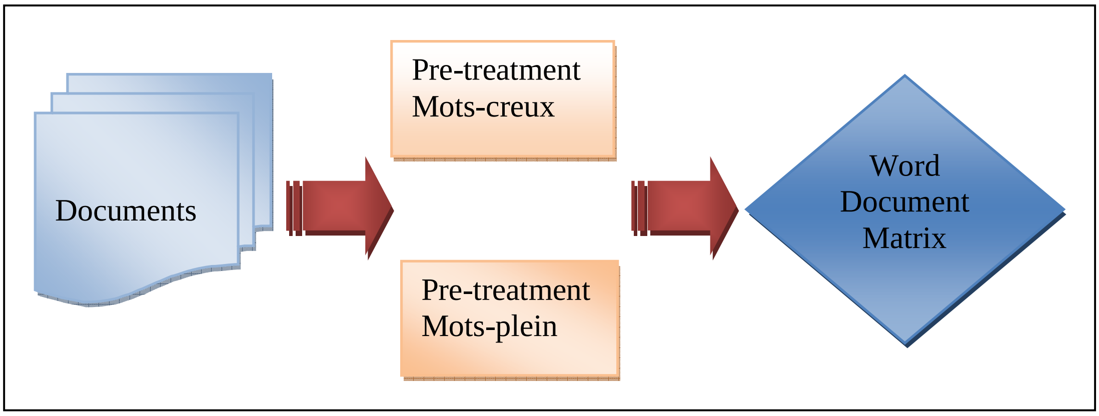
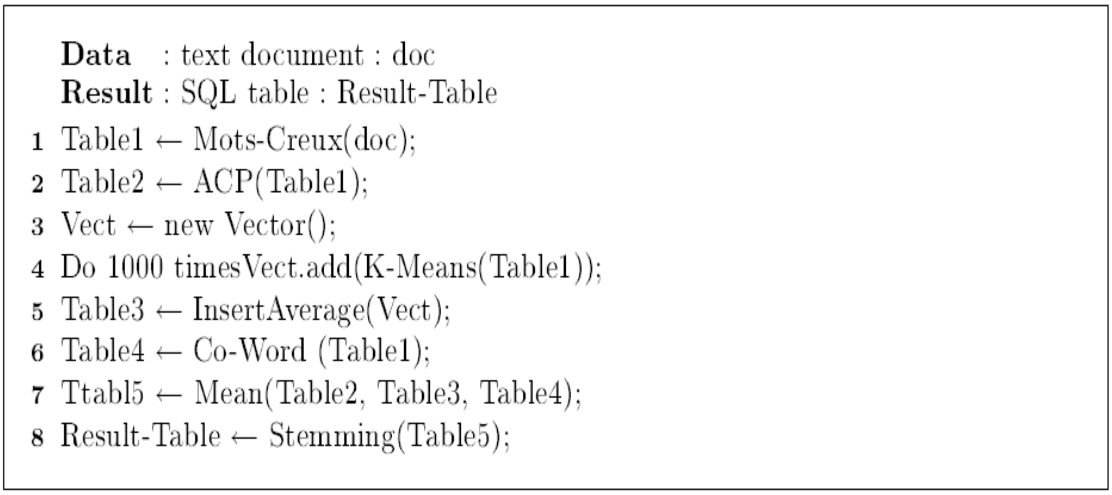


Figure 8: Proximal network pre-treatment process

Mots-creux pre-treatment process

In the mots-creux pre-treatment processing technique the input documents are primarily processed for any hollow words present in the document. This is carried out by feeding the input document into a java program which basically looks for pre-defined and pre-identified hollow or empty words present in the document. Once these hollow words are identified by the java program they are systematically deleted from the input document. This document is then fed into a program and treated to extract the rest of the words retained in the document. These extracted words are actually built into a word document matrix which forms the output stage of the pre-treatment process. This is stored as an input for the later processing stages in the proximal prototype model.



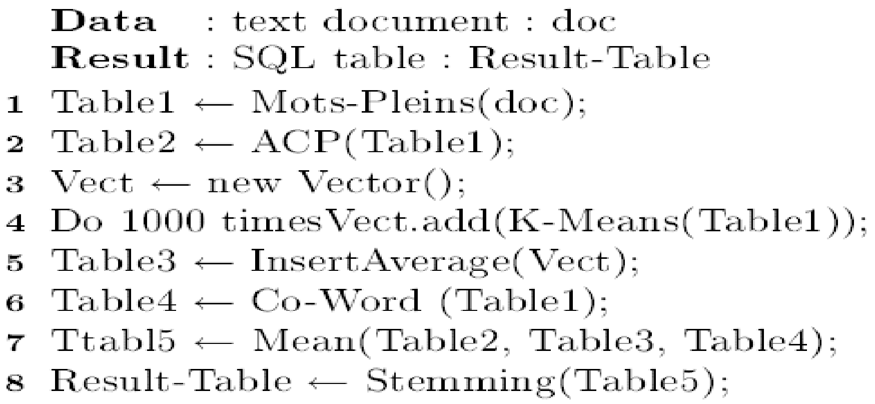
```
  Data   : text document : doc
  Result : SQL table : Result-Table
1 Table1 ← Mots-Creux(doc);
2 Table2 ← ACP(Table1);
3 Vect ← new Vector();
4 Do 1000 timesVect.add(K-Means(Table1));
5 Table3 ← InsertAverage(Vect);
6 Table4 ← Co-Word (Table1);
7 Ttabl5 ← Mean(Table2, Table3, Table4);
8 Result-Table ← Stemming(Table5);
```

Algorithm 1: General algorithm: mots-creux

The above algorithm represents our proximal prototype using the Mots-creux pre-treatment process as represented in line 1. The InsertAverage in line 5 does the average calculation on the results stored in Vect and inserts the averaged result into the data. The Vect contains all the result obtained while doing 1000 times K-Means.

Mots-Pleins Process

Similar to mots-creux pre-treatment this process is designed to identify the word entities in the input document which are later used to form the output of word document matrix. But unlike in mots-creux process here we process the input documents to actually extract word entities which are identified and considered the most important in the given context or topic the document represents. The input document is passed through the java program which is designed to identify and extract a list of words from the document. The java program actually matches the words of the input document with that present in the predefined list provided and extracts them in the process.

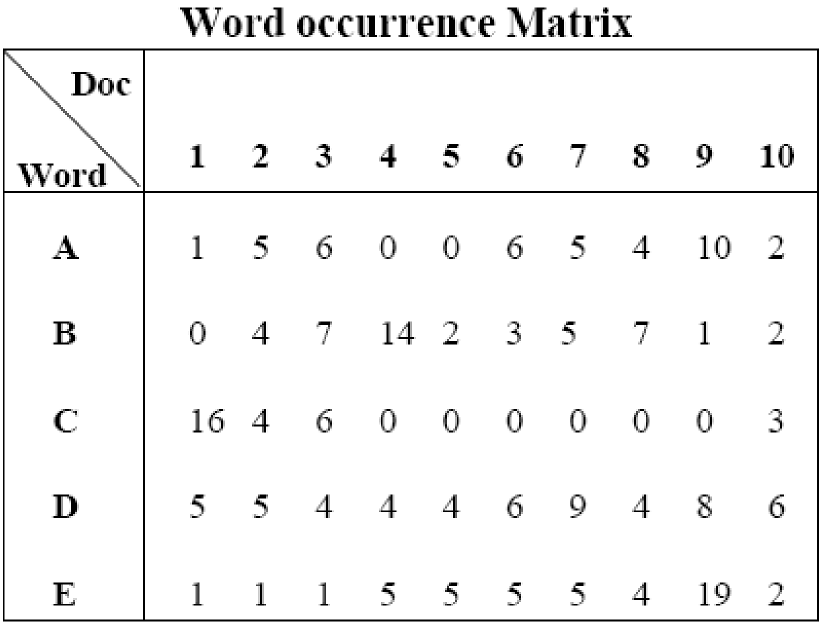


```
Data   : text document : doc
Result : SQL table : Result-Table
Table1 ← Mots-Pleins(doc);
Table2 ← ACP(Table1);
Vect ← new Vector();
Do 1000 timesVect.add(K-Means(Table1));
Table3 ← InsertAverage(Vect);
Table4 ← Co-Word (Table1);
Ttabl5 ← Mean(Table2, Table3, Table4);
Result-Table ← Stemming(Table5);
```

Algorithm 2: General algorithm: mots-pleins

The program therefore extracts all the words that match with its predefined list and stores them on to a separate data base. The pre-defined word list actually defines a set of words that most represent a domain that is under consideration. This list was generated after consulting and following guidance from the experts of the domains we are currently focusing on for experimenting our prototype. Once these words are extracted using the java program they are then transformed and stored as a word document matrix which forms the input for the mathematical modeling process.

The results obtained by the above 2 methods of pre-treatment process is actually stored into a Mysql database. These database results are then used as input information for the mathematical processing model. The result at each stage can be visualized using the graph editor tool.



Word occurrence Matrix

Doc Word	1	2	3	4	5	6	7	8	9	10
A	1	5	6	0	0	6	5	4	10	2
B	0	4	7	14	2	3	5	7	1	2
C	16	4	6	0	0	0	0	0	0	3
D	5	5	4	4	4	6	9	4	8	6
E	1	1	1	5	5	5	5	4	19	2

Figure 9: Word document matrix

4.3.1.2. Mathematical modeling process

Mathematical modeling process is the second stage of processing in our proximal prototype. The input given to this model is the result matrix obtained from the previous processing stage. In this pre-treatment process the actual proximity between word pair entities are calculated based on several statistical and mathematical models. The 3 different models used for calculating the proximity between word pairs occurring in the input document are as follows

- Principle Component Analysis
- K - Means Clustering and
- Word Association

Our processing model in actual is based on these mathematical models but are slightly customized to satisfy our processing criteria. This is essentially achieved by altering and adding certain simple computations where ever needed. Java has been used as the programming language to build all these mathematical models. Each of the above mathematical models, process the input data and provides an output in the form of a word pair matrix with a numerical value assigned to each of these word pairs.

The assigned values are computed based on the closeness or proximity of word pairs in the projected space. In our model all projections are made in a 2 dimensional space. Every word projected in the given space is compared with all the other projected words in the same space for closeness or proximity. This measure of proximity ranges between 0 and 1 and is later scaled to 0 and 100 for computational ease. Here, 0 represents the least value of proximity indicating least matched word pair and 100 indicating the best match for a word pair in terms of proximity they share.

However it is not mandatory that every word in the given space shares a proximal relationship with every other projected word in the same space. In fact if the proximity between a word pair is lower than 25 (0.25) we have designed our algorithms to discard such word pairs. This is mainly because in our later computations word pairs with proximity value lower than 25 does not make a significant change in the final result. This decision was made based of trials that we conducted using word pairs with proximal values lower than 25. We noticed that networks formed using word pairs with proximal values lower than 25 did not provide any additional information when being used in our data retrieval techniques.

Each of the above 3 algorithms will provide us a numerical proximity value between each word pairs. Each of this value has been computed differently based on the 3 different algorithms we employed. It is not necessary that all the 3 outputs obtained from the above algorithms have similar word pairs. Since each of this algorithm employs completely different approaches in calculating the proximity values it is highly probable that one technique might assign a higher value to a word pair as compared to the value obtained by the word pair through other methods.

However the outputs obtained from each of these algorithms are compared and the common word pairs prevalent across the output matrix obtained from each of these 3 different methods are extracted and then stored into the database for further computations. The outputs from each of these algorithms are later combined using the simple calculation of mean derivation and a single value for each word pair is estimated.

Principle Component Analysis (PCA)

PCA [Pearson, 1901] is a technique used to reduce multidimensional data sets to lower dimensions for analysis. It is mostly used in exploratory data analysis and for making predictive models. PCA generally involves the calculation of the eigenvalue decomposition [Jolliffe, 2002] of a data covariance matrix or singularvalue decomposition of a data matrix, usually after mean centering the data for each attribute. The results of a PCA are usually discussed in terms of component scores and loadings.

PCA can also be defined as a way of identifying patterns in data and expressing the data in pattern to highlight their similarities and differences. Since patterns in data can be hard to find especially in data of high dimension, where graphical representation in either difficult or impossible, PCA emerges as a powerful tool for analyzing such data.

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA is theoretically the optimum transform for a given data in least square terms.

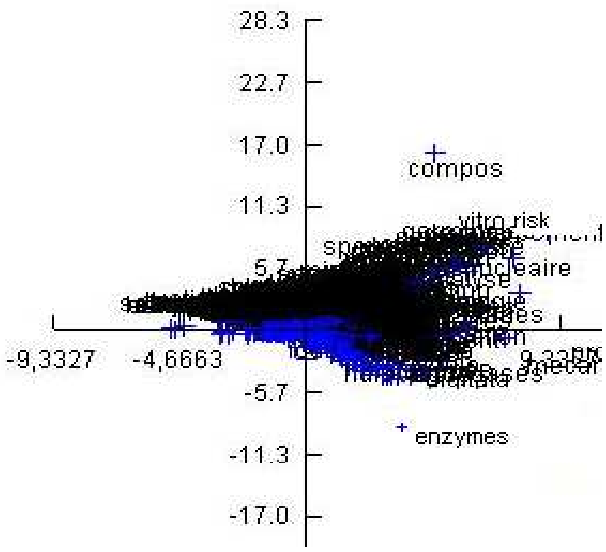


Figure 10: Data from Arabidopsis projected using PCA

PCA can be used for dimensionality reduction [Jolliffe, 2002] in a data set by retaining those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" aspects of the data. However, depending on the application this may not always be the case.

For a data matrix, XT, with zero empirical mean (the empirical mean of the distribution has been subtracted from the data set), where each row represents a different repetition of the experiment, and each column gives the results from a particular probe, the PCA transformation is given by:

$$\mathbf{Y}^T = \mathbf{X}^T\mathbf{W}$$

$$= \mathbf{V}\mathbf{\Sigma}$$

where, V Σ WT is the singular value decomposition (svd) of XT.

PCA has the distinction of being the optimal linear transformation for keeping the subspace that has largest variance. This advantage, however, comes at the price of greater computational requirement if compared, for example, to the discrete cosine transform.

```
Data   : SQL Table : table
Result : SQL Table : result-table
vect1 ← Covariance(table);
vect2 ← Svd(vect1);
vect3 ← Do-distance(vect2);
vect4 ← Filtre(vect3, filtre_value);
InsertTable(result-table, vect4);
Return result-table;
```

Algorithm 3: PCA

In the algorithm the Svd projects the words in a two dimensional space using the algorithm: NumericalTools.svdcmp.

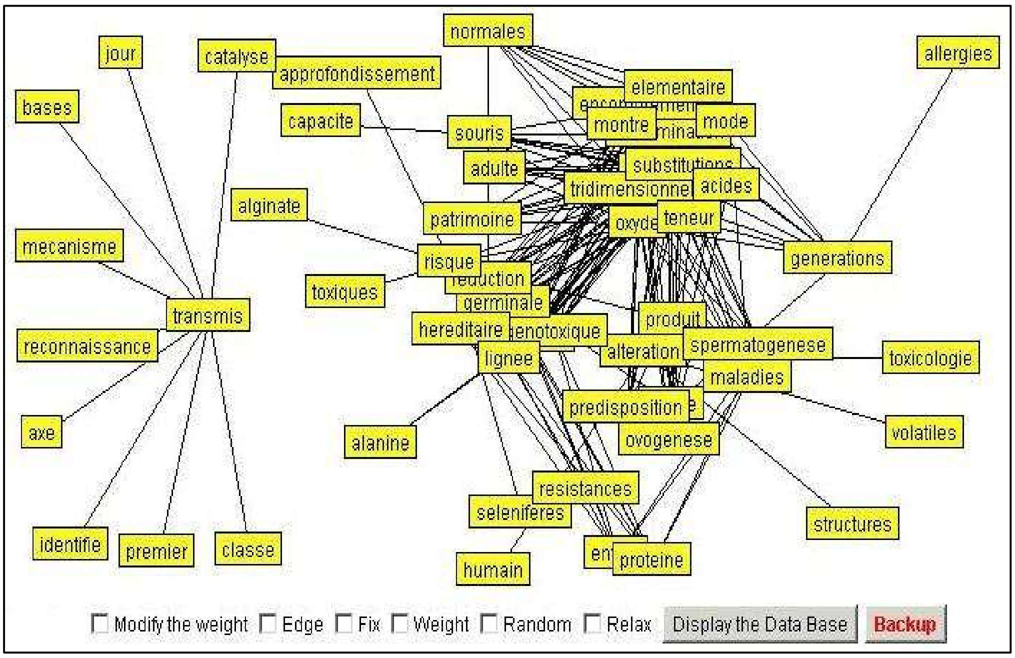


Figure 11: PCA results visualized using graph editor

In our model we utilize the functionalities of PCA to plot the word pair network to calculate the overall proximity in the chosen documents. We first input the word frequency matrix into the PCA Java program. The program performs all the PCA mathematical calculations on the input data and return results in a word to word matrix. Once the word pair calculation is completed, the program plots value of all the word pairs thus calculating the word proximity. The result matrix produced by the PCA program is then fed into the graph editor which in turn plots the word network obtained using PCA analysis.

We currently are using the singular value decomposition function called the SVDCmp( ) function [Golub and Kahan, 1965] which calculates the position of the word processed in a given dimensional space. Based on this projection proximity between these projected word pairs are calculated, which is inverse of the actual distance using the Euclidian formula. The values obtained by this algorithm range between the value of 0 and 1. These values are then pre-treated and all values between 0 and 0.25 are eliminated or discarded and the rest of the matrix is stored in the database.

These pre-treated values are then used to obtain the network of words based on their values shared between each word pairing. Therefore, each word is connected or linked to every other proximally related word in the network. This in turn forms a word network which can be viewed in the graphical display shown. The above algorithm can be applied on documents to connect them based on their proximity to produce document network.

K- Means Clustering

K-means is one of the most famous clustering algorithms. It is an algorithm used to classify or to group objects based on attributes/features into K number of group. K is a positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Thus the most important purpose of K-mean clustering is to classify the data. K-means utilizes the Euclidian method to calculate the number of clustering required and to decide which data falls into what cluster.

K-means [MacQueen, 1967], is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of the fact that different location cause different result. So, the best choice is to place them as much as possible far away from each other.

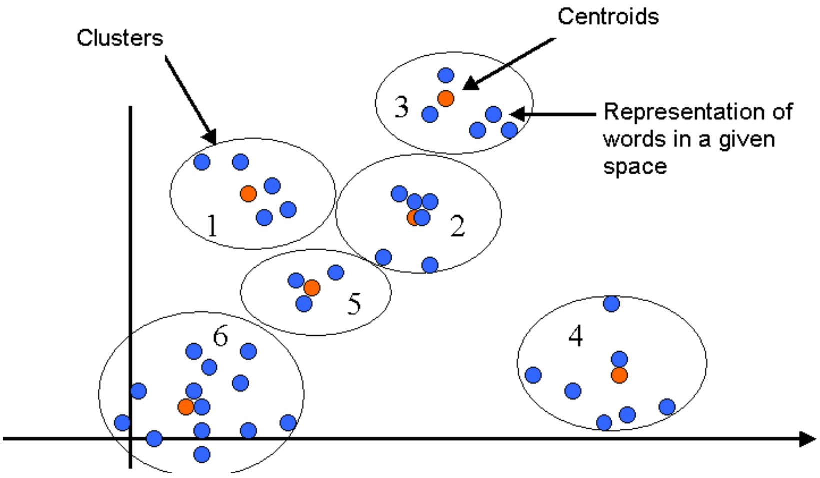


Figure 12: A sample of K-Means projection

The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early grouping is done. At this point we need to re-calculate k new centroids as barycentre of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop is thus generated and as a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move anymore and stays stable even when iteration is repeated.

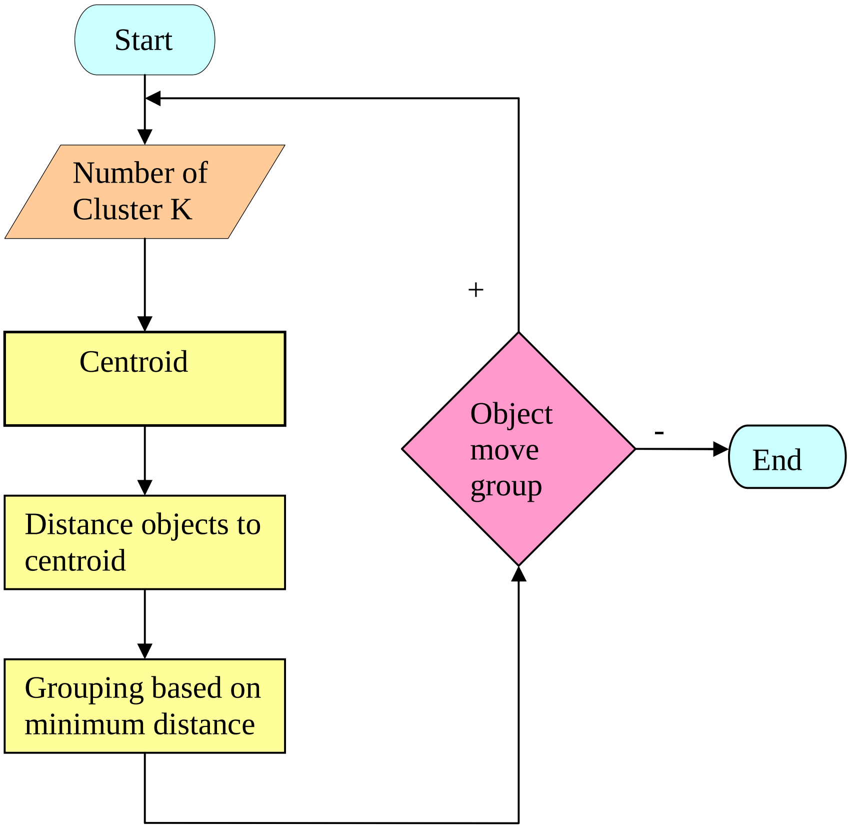

Start
Number of Cluster K
Centroid
Distance objects to centroid
Grouping based on minimum distance
Object move group
+
-
End

Figure 13: Flow chart illustration of K-Means clustering algorithm

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect. The diagram shows a flow chart depicting the typical functioning of K-Means clustering. As can be seen in the above flow chart, in K- means algorithm, several numbers of iterations are carried out in the loop until a data no more changes its cluster and pending there is no room for new cluster formation.

K-means is a simple algorithm [Moore, 2003] that has been adapted to many problem domains. As we are going to see, it is a good algorithm to work with for distance computations in our proximal prototype model. The primary mathematical equation that K-Means employs is the Euclidian distance metrics. The formula for the Euclidian distance between a point X (X1, X2, etc.) and a point Y (Y1, Y2, etc.) is:

$$d= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

In our prototype we employ k-means algorithm with a simple aim of calculating and identifying word entities that tend to group together and thus forming a cluster. We pass the word document matrix obtained from our previous processing into the K-means algorithm. This algorithm begins its iterations until word entities form stable clusters. The initial value for K has been determined by us on a random basis. This was primarily due to the fact that the value of K in our experimentations did not play a significant part in determining the actual grouping of word entities. However, we have decided to maintain a uniform value throughout all iterations for all different data mainly for consistency purpose.

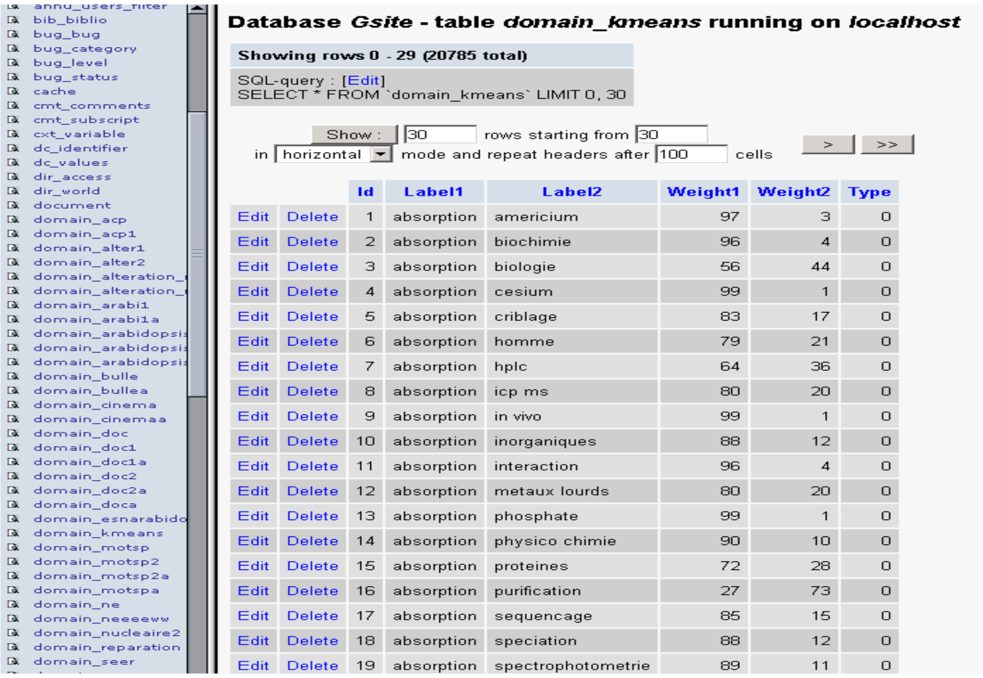
```
  Data   : table sql : Table
  Result : table sql : Vector res_vect
1 Clusters ← new Vector();
2 Clusters.init;
3 forall (word w ∈ Table) do
4     while (unstable(Clusters)) do
5         Do-Euclian(w, Clusters);
      end
  end
6 forall (Word Pair(w1,w2) such as w1, w2 ∈ getWords(Ci)) do
7     vect.add((w1 w2 1));
  end
8 forall (Word Pair(w1,w2) such as w1 ∈ getWords(Ci), w2 ∈ getWords(Cj),
  where i!=j) do
      vect.add((w1 w2 0));
  end
9 Return vect;
```

Algorithm 4: K-Means

In the algorithm line 2 presents all the words with its co-ordinates in one single cluster e.g. (C1,…..Ci…, Cn), where C1= ((w1 2 3) (w 4 5 6)). The Do-Euclidian in line 5 calculates the distance between the words using the Euclidian equation. GetWords, in line 6 returns the words from the formed clusters. Once the algorithm is activated it begins the iterations to form word clusters. These iterations are continued until the number of clusters remains stable and there is no more possibility of words changing clusters. Once we obtain the final result of our k-means algorithm, we then assign values to each of the word pairs.

We employ a simple Boolean method to actually determine the proximity of these word entities. Here we do not intend to compute the actual distance of word entities from one another, but what are significant to our research interest are the word entities appearing in the same cluster. We simply assume that word entities occurring in the same cluster form a good word pair and assign a value of 1 and similarly all word entities that do not occur in same cluster are assumed to make bad word pairs and hence assigned a value of 0.

Table 1: Snapshot of K-means result database



Database *Gsite* - table *domain_kmeans* running on *localhost*

Showing rows 0 - 29 (20785 total)

SQL-query : [Edit]
SELECT * FROM `domain_kmeans` LIMIT 0, 30

Show : 30 rows starting from 30
in horizontal mode and repeat headers after 100 cells

		Id	Label1	Label2	Weight1	Weight2	Type
Edit	Delete	1	absorption	americium	97	3	0
Edit	Delete	2	absorption	biochimie	96	4	0
Edit	Delete	3	absorption	biologie	56	44	0
Edit	Delete	4	absorption	cesium	99	1	0
Edit	Delete	5	absorption	criblage	83	17	0
Edit	Delete	6	absorption	homme	79	21	0
Edit	Delete	7	absorption	hplc	64	36	0
Edit	Delete	8	absorption	icp ms	80	20	0
Edit	Delete	9	absorption	in vivo	99	1	0
Edit	Delete	10	absorption	inorganiques	88	12	0
Edit	Delete	11	absorption	interaction	96	4	0
Edit	Delete	12	absorption	metaux lourds	80	20	0
Edit	Delete	13	absorption	phosphate	99	1	0
Edit	Delete	14	absorption	physico chimie	90	10	0
Edit	Delete	15	absorption	proteines	72	28	0
Edit	Delete	16	absorption	purification	27	73	0
Edit	Delete	17	absorption	sequencage	85	15	0
Edit	Delete	18	absorption	speciation	88	12	0
Edit	Delete	19	absorption	spectrophotometrie	89	11	0

During our initial K-means algorithm testing we noticed that some word entities tend to behave differently by changing their cluster group each time K-means algorithm is initiated. We termed these word entities as indecisive word entities which always wobble between 2 clusters. This was basically because of the fact that they were the border word entities of these 2 clusters. This resulted in word entities sometimes appearing in either of these different clusters in different K-means processing.

To solve this problem we decided to carry out K-means over several times to obtain more consistency. We realized after several trial and error runs that 1000 K-means run gave fair enough understanding of which cluster maximum times a indecisive word entity belonged too. Hence for this reason we decided to run k-mean algorithm for over a 1000 number of runs. However, each run began with a different starting point in the given space. Each time the k-mean was run its final data was recorded in a separate database linked with the run number. Once all 1000 runs were completed we then averaged out the value for all the word entities and stored the end result in the k-means output database.

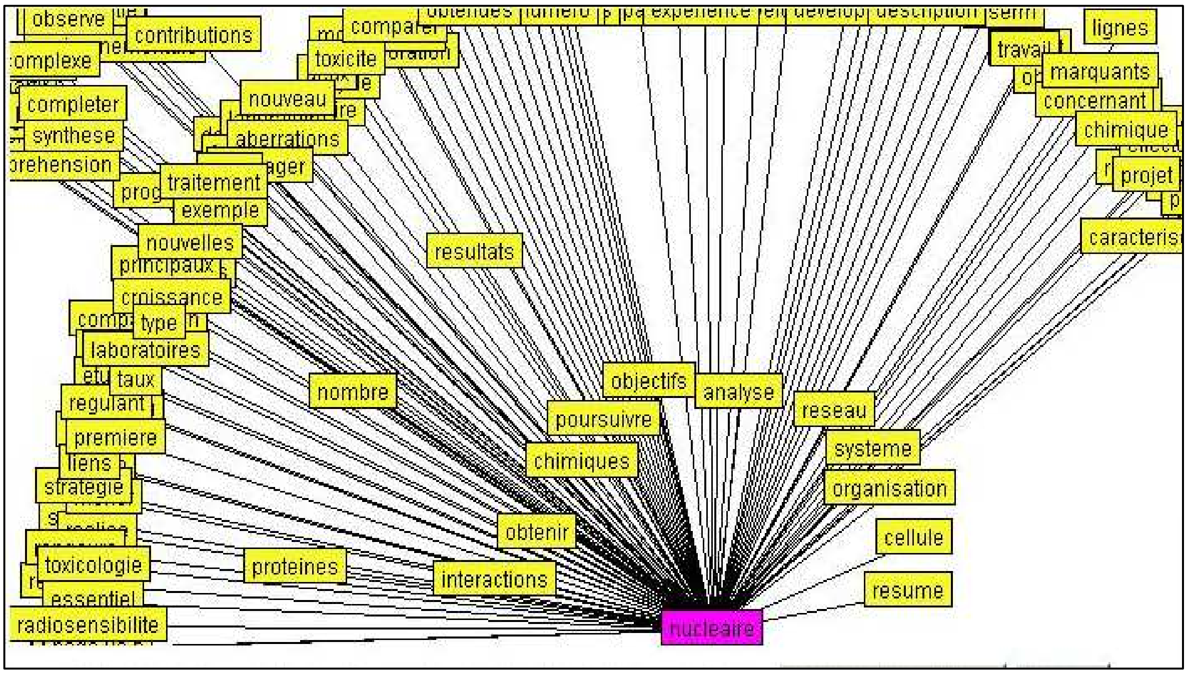


Figure 14: An illustration of one of the K-Means clusters

The basic idea of using K-means in our proximity calculations was mainly to involve clustering perspective to project our data and apply the Euclidian distance metrics in calculating the proximity between our word entities. In K means we actually project data to form different groups. Word entities occurring in each of these groups or cluster are considered to be proximally close while word entities occurring in different clusters are assumed to be proximally a bad match.

The figure displays the word network graphical representation obtained using K-means clustering. Here the K-means algorithm separates each word and using the Euclidian's calculation forms different clusters. Words seeming proximally closer are put into the same cluster and then a Boolean value is provided depending on whether a word is listed in a given cluster. The resulting values of words present in clusters are then compared and averaged to obtain a more precise result value. This result is stored on to the database and then utilized by the graph editor to showcase the network obtained using the K-means clustering as detailed earlier.

Word Association

Word association is a method in which a person says the first word they think of when a particular word is said, which may help to discover about how parts of the mind work. In the proximal prototype model we try to calculate the proximity of word entities using approximation method of word association algorithm.

Word association is a common word game [Packard, 1961] involving an exchange of words that are normally associated with one another. Once an original word has been chosen, usually randomly or arbitrarily, a player will find a word that they associate with it and make it known to all the players, usually by saying it aloud or writing it down as the next item on a list of words so far used. The next player must then do the same with this previous word. This continues in turns for any length of time, but often word limits are set, so that the game is agreed to end after, for instance, 400 words.

Usually, players write down the next word by merely using the first word that comes to their mind after they hear the previous one. Sometimes however they may put in more thought to find a more creative connection between the words. Exchanges are often fast and sometimes unpredictable (though logical patterns can usually be found without difficulty). Sometimes, a lot of the game's fun can arise from the seemingly strange or amusing associations that people make between words. It is also found amusing what you can get from an original word, and how they contrast distinctly, for example, from the word "tea" you could get the word "murder".

It is believed by some that the word association game can reveal something of a person's subconscious mind (as it shows what things they associate together); however some are skeptical of how effective such a technique could be in psychology. However, more often than not, most of the fun of the game comes from observing the erratic links between words, where the amusement comes from wondering how someone else's mind managed to make such an association.

Certain popular psychologists have shown an ability to predict people's word associations, and some suggest that humans actually find it very difficult to disassociate words such that they become more predictable when told to do so. Word association has been used by market researchers to ensure the proper message is conveyed by names or adjectives used in promoting company's products. Word association dates back to Avicenna, who developed a system for associating changes in the pulse rate with inner feelings, which is seen as an anticipation of the word association test. In the early years of psychology, many doctors noted that patients exhibited behavior that they were not in control of. Some part of the personality seemed to have an influence on that person's behavior that was not in his/her conscious control. This part was, by function, unconscious, and became so named the Unconscious.

Carl Jung theorized that people connect ideas, feelings, experiences and information by way of associations [Jung and Jaffé, 1965], that ideas and experiences are linked, or grouped. For instance given the word 'volcano', a common word people might submit would be 'lava', and this would result in a very strong connection between 'volcano' and 'lava'. On the other hand, given the word 'volcano', fewer people might associate it with something like 'birthday party', resulting in a very weak connection or no connection at all.

In some of the association experiment used during early experimentation by Galton [Forrest, 1974], the subjects were asked to respond to a stimulus word with the first word that comes to their mind. These associative responses are therefore explained using the principle of learning by contiguity. It is noticed that objects once experienced together tend to become associated in the imagination, so that when any one of them is thought of the others are likely to be thought of as well, in the same order of sequence or co-existence as before. This is clearly related to the mental association by contiguity.

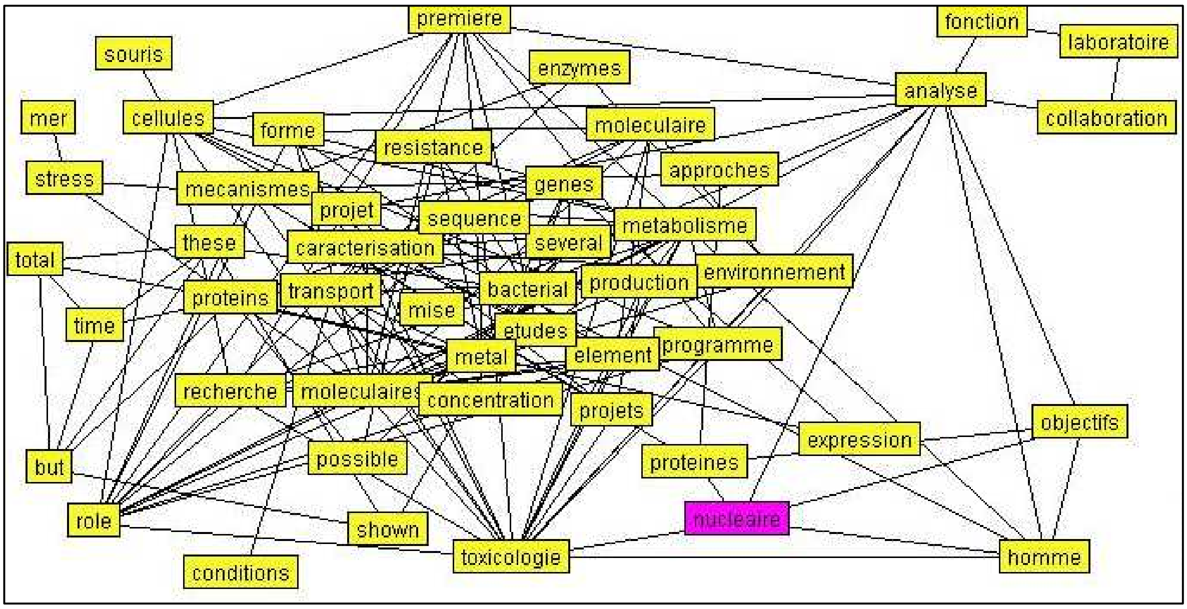
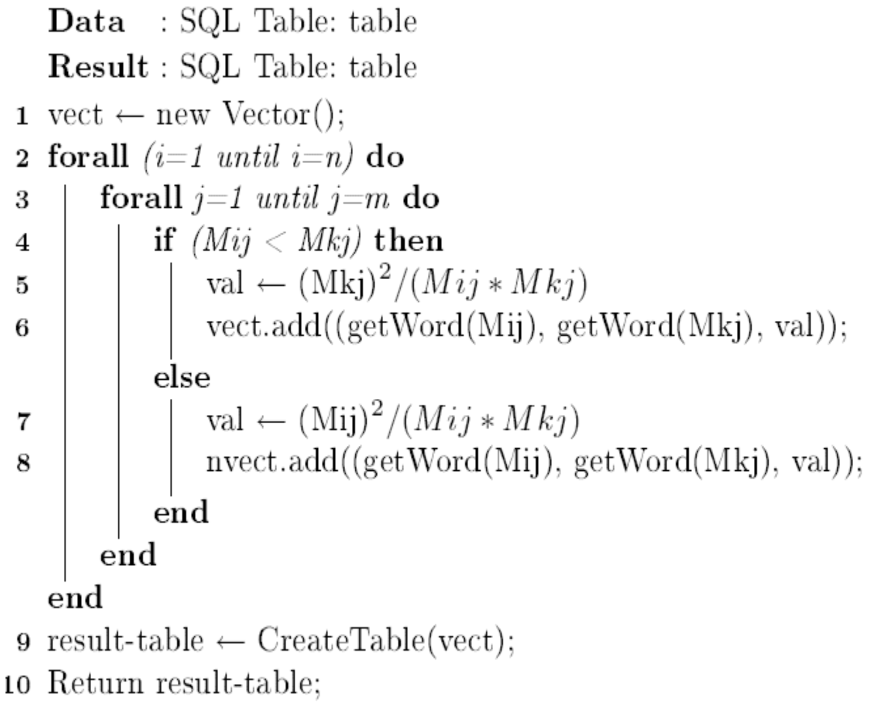


Figure 15: Word association using co-occurrence

According to the law of contiguity, the association strength between two words should be a function of the relative frequency of the two words being perceived together, i.e. the relative frequency of the two words occurring together. With these assumptions holding true it should be possible to predict word associations from the common occurrences of words in texts.

In our word association algorithm the co-occurrence of a word pair entity in a given set of documents is analyzed. Here very simple and basic mathematical calculations (i.e. covariance) are used which provide a word pair a value ranging from 0 to a maximum of 1.

In our algorithm we utilize combination of word association and co-occurrence. That is we count the number of instances (frequency of occurrence) a word pair co-occur in a given phrase of a document under consideration using the assumptions detailed above.



```
Data   : SQL Table: table
Result : SQL Table: table
vect ← new Vector();
forall (i=1 until i=n) do
    forall j=1 until j=m do
        if (Mij < Mkj) then
            val ← $(Mkj)^2/(Mij * Mkj)$
            vect.add((getWord(Mij), getWord(Mkj), val));
        else
            val ← $(Mij)^2/(Mij * Mkj)$
            nvect.add((getWord(Mij), getWord(Mkj), val));
        end
    end
end
result-table ← CreateTable(vect);
Return result-table;
```

Algorithm 5: Word association

The value for each word pair is obtained using the formula below where Cij and Ckj represent the value of the words they represent. These values are being compared for word association and the following formula is used to obtain a value representing the word association of co-occurrence between these pairs.

if Cij <Ckj

then

$(Ckj)^2/(Cij*Ckj)$

else

$(Cij)^2/(Cij*Ckj)$

The values thus obtained for each word pair is then stored into the data base as a word pair matrix. These matrixes can be graphically represented using the graph editor as shown in the figure.

The result matrixes obtained using the above three algorithms are then combined using the mean equation. Here word entity pairs present in all the three results are extracted and a mean value is calculated for each of these word pair entities. This result is stored as the final result matrix which is then used by the graph editor to produce a graphical view of the proximal network.

The combination of the above results is currently done by finding the mean of the three values. We do not rule out the possibility of using more sophisticated algorithms that can give a more precise result when combining the above result. More precisely algorithms that would minimize the information loss occurring at this stage using our current method. One such method that we can suggest using and are presently working on is the linear approximation calculus.

4.3.1.3. Post treatment process

This is the final processing stage in our proximal prototype model. In this stage the output matrix obtained from the previous process is subjected to partial/ selective stemming using an external stemming algorithm. In the stemming process the inflected or derived words are reduced to its base or root form using the stemming algorithm. However we decide on which words in the results matrix requires stemming. We have chosen to carry out partial stemming as we believe that, by applying complete stemming we might subject our data to the possibility of losing useful information needed to build our proximal network. Hence we have pre-defined a set of words we consider are necessary to be subjected to stemming process in each knowledge domain.

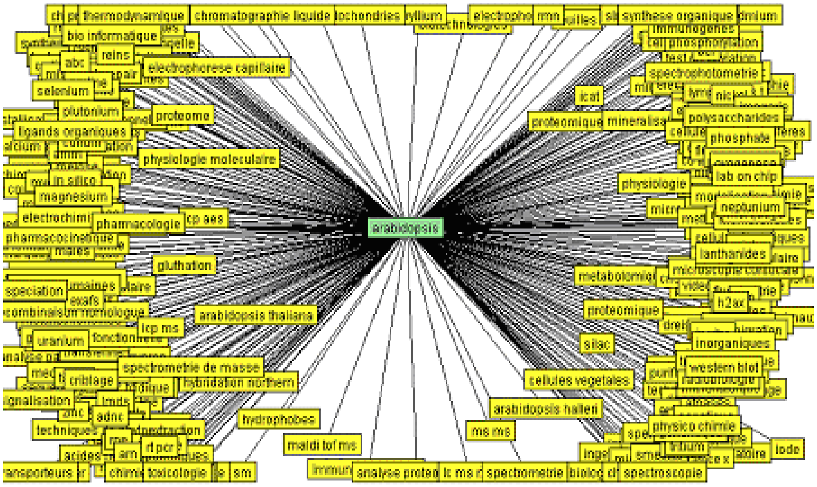


Figure 16: An extract of proximal network

The output from this process is then stored into a Mysql database. This data can be later visualized using the java application graph editor for visualization as detailed in the earlier chapters. In our proximal network we construct word network around a central word entity which actually represents the domain subject on which the entire network is constructed. Figure 16 illustrates an example of a proximal network visualized using the graph editor program. Here the proximal network is built on the subject Arabidopsis. This means that the documents we initially chose to be used in our pre-processing stages were all related to the Arabidopsis domain. However the figure illustrates the proximity of word entity Arabidopsis with all the other word entity it is proximally related to in our network.

Hence the main idea here is to analyze large amounts of data very quickly and then develop a representational network which can depict the relationship shared by the word entities in these documents which can be machine readable.

We initially started with 3 research topics for which we considered building the proximal network. Once the initial results proved very satisfactory we decided to extend from 3 research topics to 15 topics. Hence the documents processed are relating to the research activities carried out in the chosen 15 fields from the ToxNuc-E project namely

- Altération réparation

- Arabidopsis
- Bactéries
- Chélation biologique
- Cibles moléculaires
- Décorporation
- Génotoxicologie
- Iode
- Levure
- Méthodologie et spéciation
- Nephro et toxicocancérogenèse
- Stress oxydant
- Toxicogénomique
- Transfert sol plantes
- Transporteurs

This proximal network primarily evaluates word entities based on the physical distance that separates word entities formed after using our processing models. Currently, we have successfully processed around 3423 words computing their actual physical occurrence. We have been able to successfully build a proximal network of over 50,000 word pair, an extract of which is seen in the above figure depicting the Arabidopsis proximal network. Each of these word pair is related using the value obtained from the prototype and is connected using the simple UML link of association detailed in the following chapter.

This data processing method in itself can be independently used for processing large number of documents in an efficient and productive way. The fact that the small time taken for processing huge amounts of data makes it an important aspect in ontology construction for multiple domain scalable.

4.4. Limitations

One of the major limitations of our proximal prototype is that the end result is completely dependent on the type of into provided to the model. Hence the quality of documents used as an input has direct impact on the quality of the results obtained. Hence it is very important that we ensure that the documents are well representing the knowledge domain for which the proximal prototype is built.

The other possible areas for future research are exploring new algorithms that might be used in our mathematical modeling process. Currently for simplicity we have restricted our use to the three classical algorithms but it would be definitely interesting to see the results when more varied mathematical models are employed for the calculations.

It is also very evident that this model will enable us to automate data analysis and representation by a large extent but however its accuracy is largely limited by the input data. Over main idea for developing this model was the fact that the large amounts of research materials present on the ToxNuc-E platform needed systematic processing and classification. These documents were too large and many in number for manual classification. Since this was a virtual platform with 700 more researchers' connected, it became increasingly difficult to maintain the documents uploaded by the members. It became necessary that we employ models that would enable easy and fast processing of such documents. Proximal network does precisely this when used with our other models in our knowledge representation approach.

5. Semantic network

5.1. Introduction

Semantic Network can be defined as a labeled, directed graph with nodes representing physical or conceptual objects and labeled arcs representing relations between objects. This permits the use of generic rules, inheritance, and object-oriented programming. A semantic network is often used as a form of knowledge representation with directed graph [Collins and Quillian, 1969] consisting of vertices representing concepts and edges or nodes representing semantic relations between these concepts. Semantic network is basically used as a technique to represent knowledge in a machine readable form.

The basic anatomy of a semantic network can be described using the 2 principle elements [Sowa, 1987] representing any semantic network.

1. **Concepts**: They are nothing but ideas or thoughts that have meaning.
2. **Relations:** These mainly describe specific kinds of links or relationships between two concepts.

The figure17 represents a simple semantic network of concepts and relations. Here rose, flower, plant, perfume represent the concept nodes of the network while the arcs connecting them to one another are the; is-a, part-of and has (associative) relations drawn between the concept nodes.

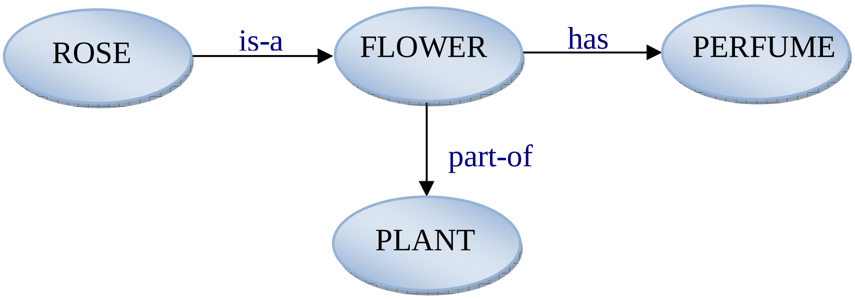


Figure 17: Semantic network depicting relation between nodes

The concept of semantic network is now fairly old in the literature of cognitive science and artificial intelligence, and has been developed in so many ways and for so many purposes in its 20-year history that in many instances the strongest connection between recent systems based on networks is their common ancestry. The term semantic network as it is used now might therefore best be thought of as the name for a family of representational schemes rather than a single formalism.

Semantic network is revolutionizing the way people and organizations visualize, store and communicate knowledge through the practical application of semantic network theory [Quillian, 1968]. Semantic networking is based on over thirty years of research in artificial intelligence, cognitive psychology, mimetic and learning theory, and has been independently proven to be significantly more effective in the transfer of knowledge. Semantic networks can also be termed as a common type of machine readable dictionary as they represent data such that it can be easily interpreted by machines.

5.2. State of the art

The idea of linking concepts together is very old, perhaps dating as far back as Aristotle when he explored and systematized the classical categorization theory initially proposed by Plato the Greek philosopher. Using the classical categorization Aristotle analyzed the differences between classes and objects. He also applied intensively the classical categorization scheme in his approach to the classification of living beings establishing this way the basis for natural taxonomy.

The classical Aristotelian view claims that categories are discrete entities characterized by a set of properties which are shared by their members. In analytic philosophy, these properties are assumed to establish the conditions which are both necessary and sufficient to capture meaning. According to the classical view, categories should be clearly defined, mutually exclusive and collectively exhaustive.

The oldest known semantic network was drawn in the 3rd century AD by the Greek philosopher Porphyry in his commentary on Aristotle’s categories. Porphyry used it to illustrate Aristotle’s method of defining categories by specifying the genus or general type and the differentiae that distinguish different sub types of the same super type. A Tree of Porphyry version drawn by logician Peter of Spain in 1329 illustrates the categories under substance, which is called the supreme genus or the most general category.

An arbor porphyriana or Porphyrian [Jevons, 1870] tree as it is commonly known, created by Porphyry, is a hierarchical (tree structured) ontology, construction in logic consisting of three rows or columns of words; the middlemost whereof contains the series of genus and species, and bears some analogy to the trunk. The extremes, containing the differences, are analogous to the branches of a tree.

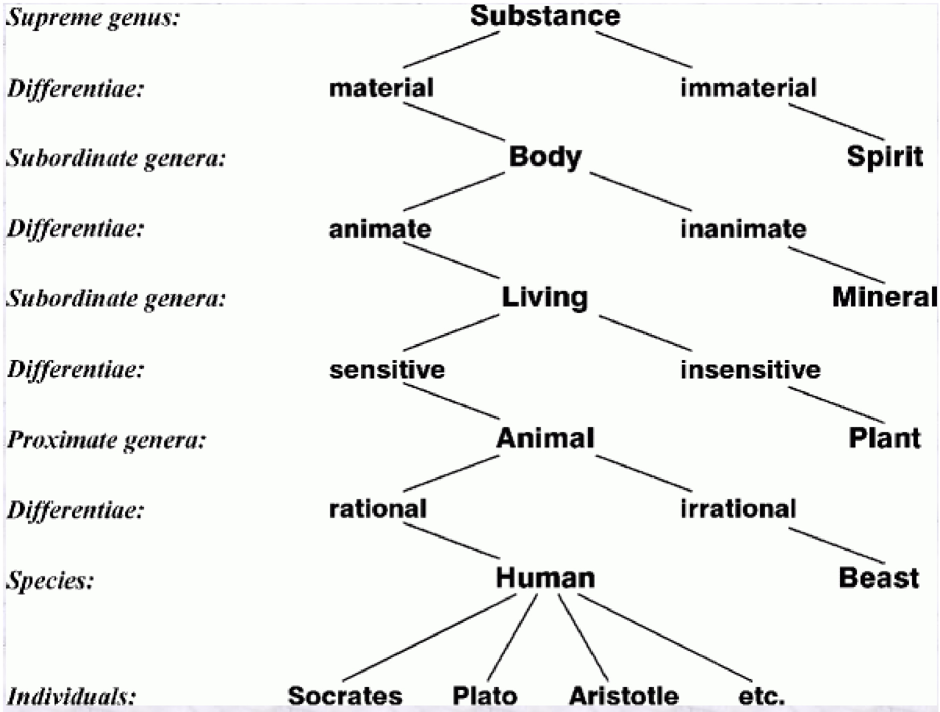


Figure 18: Tree of porphyry

The arbor porphyriana has also been known as scala praedicamentalis. It is a known fact that until the late 19th century, the tree of porphyry was being taught to students of logic. Despite its age the tree of Porphyry represents the common core of all modern hierarchies that are used for defining concept types. The first implementations of semantic networks were used to define concept types and patterns of relations for machine translation systems.

Silvio Ceccato the founder and director of the first Center for Cybernetics in Milan, Italy in 1961 developed co-relational nets, which were based on 56 different relations, including subtype, instance, part-whole, case relations, kinship relations, and various kinds of attributes. He used the correlations as patterns for guiding a parser and resolving syntactic ambiguities.

Margaret Masterman’s (a pioneer in the field of computational linguistics) system at Cambridge University also in 1961, was the first to be called semantic network. She actually developed a list of 100 primitive concept types, such as folk, stuff, thing, do and be. In terms of those primitives, her group defined a conceptual dictionary of 15,000 entries. She organized the concept types into a lattice, which permits inheritance from multiple super types. The basic principles and even many of the primitive concepts have survived in more recent systems of preference semantics [Fass and Wilks, 1983].

Semantic nets for computers were first invented by Richard H. Richens of the Cambridge language research unit in 1956 as an interlingua for machine translation of natural languages. They were developed by Robert F. Simmons at the system development corporation, Santa Monica, California in the early 1960s and later its modern incarnation featured prominently in the work of Ross Quillian in 1966, when Quillian wrote his PhD thesis which described a system for allowing the meaning of words to be modeled on a computer such that computational use of these meanings would be possible. This became the basis for the idea of a semantic network. Since then, several decades of research have refined the idea to its fullest modern expression.

Among current systems, the description logics include the features of the Tree of Porphyry as a minimum, but they may also add various extensions. They are derived from an approach proposed by Woods in 1975 and implemented by Brachman in the year 1979 in a system called Knowledge Language ONE (KL-ONE) [Brachman et al., 1991].

The Tree of Porphyry, KL-ONE, and many versions of description logics are subsets of classical first order logic (FOL). They belong to the class of monotonic logics, in which new information monotonically increases the number of provable theorems, and none of the old information can ever be deleted or modified. Some versions of description logics support non monotonic reasoning, which allows default rules to add optional information and cancelling rules to block inherited information. Such systems can be useful for many applications, but they can also create problems of conflicting defaults.

Although the basic methods of description logics are as old as Aristotle, they remain a vital part of many versions of semantic networks and other kinds of systems. Much of the ongoing research on description logics has been devoted to increasing their expressive power while remaining within an efficiently computable subset of logic [Brachman et al. 1991]; [Woods and Schmolze 1992].

The most established example of semantic network processing approach is the Collins & Quillian Semantic Network Model [Collins and Quillian, 1970]. This approach states that the meanings of words are embedded in networks of other meanings. Knowledge is validated and acquires meaning through correlation with other knowledge [Harley, 1995]. The connections within a semantic network are not only associative in nature and the links within the network have a semantic value. In the Collins and Quillian model semantic nets are composed of simple concepts, concrete-abstract (is-a) relations and part-whole (attribute, is, has, can) relations.

The schema theory of Rumelhart and Ortony [Rumelhart and Ortony, 1977] claims that personal knowledge is stored in information packets or schemas that comprise our mental constructs for ideas. Each schema we construct represents a mini-framework to inter relate elements or attributes of information about a topic into a single conceptual unit. These mini-frameworks are organized by the individual into a larger network of interrelated constructs known as a semantic network. These networks are composed of nodes: representations of schemas. Ordered labeled relationships define the propositional relationship between the nodes.

Two recent description logics are DAML and OIL [Horrocks et al., 2001], which are intended for representing knowledge in the semantic web [Berners-Lee et al., 2001], a giant semantic network that spans the entire Internet.

The Semantic Web is an evolving extension of the World Wide Web in which the semantics of information and services on the web is defined, making it possible for the web to understand and satisfy the requests of people and machines to use the web content. It derives from W3C director Tim Berners-Lee's vision of the Web as a universal medium for data, information, and knowledge exchange [Herman, 2007].

At its core, the semantic web comprises a set of design principles (design issues, W3C), collaborative working groups, and a variety of enabling technologies. Some elements of the semantic web are expressed as prospective future possibilities that are yet to be implemented or realized [W3C, 2008]. Other elements of the semantic web are expressed in formal specifications [Herman, 2007]. Some of these include Resource Description Framework (RDF), a variety of data interchange formats (e.g. RDF/XML, N3, Turtle, N-Triples), and notations such as RDF Schema (RDFS) and the Web Ontology Language (OWL), all of which are intended to provide a formal description of concepts, terms, and relationships within a given knowledge domain.

5.3. Types of semantic networks

There are several elaborate types of semantic networks connected with corresponding sets of software tools used for lexical knowledge engineering, like the Semantic Network Processing System (SNePS) of Stuart C. Shapiro [Shapiro and Rapaport, 1992] or the MultiNet (Multilayered Extended Semantic Network) paradigm of Hermann Helbig which is especially suited for the semantic representation of natural language expressions and used in several NLP applications.

One can consider mind map to be a very free form variant of semantic network. By using colors and pictures the emphasis is on generating semantic net which evokes human creativity. However, a fairly major difference between mind maps and semantic networks is that the structure of a mind map, with nodes propagating from a centre and sub-nodes propagating from nodes, is hierarchical, whereas semantic networks, where any node can be connected to any node, have a more hierarchical structure.

In the 1960s to 1980s the idea of a semantic link was developed within hypertext systems as the most basic unit, or edge, in a semantic network. These ideas were extremely influential, and there have been many attempts to add typed link semantics to HTML and XML.

An example of a semantic network is WordNet [WordNet, 2002], a lexical database of English. Such networks involve fairly loose semantic associations that are nonetheless useful for human browsing. It is possible to represent logical descriptions using semantic networks such as the existential graphs of Charles S. Peirce or the related conceptual graphs by John F. Sowa. These have expressive power equal to or exceeding standard first-order predicate logic. Unlike WordNet or other lexical or browsing networks, semantic networks using these can be used for reliable automated logical deduction. Some automated reasoners exploit the graph-theoretic features of the networks during processing.

Machine implementations of semantic networks were first developed for artificial intelligence and machine translation, but earlier versions have long been used in philosophy, psychology, and linguistics.

The common feature to all semantic networks is a declarative graphical representation that can be used either to represent knowledge or to support automated systems for reasoning about knowledge. Some versions are highly informal, but other versions are formally defined systems of logic. Following are six of the most common kinds of semantic networks [Sowa, 1987], each of which is discussed in detail in the following sections:

- **Definitional Networks:** This emphasizes the subtype or is-a relation between a concept type and a newly defined subtype. The resulting network, also called a generalization or subsumption hierarchy, supports the rule of inheritance for copying properties defined for a super type to all of its subtypes. Since definitions are true by definition, the information in these networks often assumed to be necessarily true.

- **Assertional Networks:** These are designed to assert propositions. Unlike definitional networks, the information in an assertional network is assumed to be contingently true, unless it is explicitly marked with a modal operator. Some assertional networks have been proposed as models of the conceptual structures underlying natural language semantics.

- **Implicational Networks:** These use implication as the primary relation for connecting nodes. They may be used to represent patterns of beliefs, causality, or inferences.

- **Executable Networks:** These include some mechanism, such as marker passing or attached procedures, which can perform inferences, pass messages, or search for patterns and associations.

- **Learning Networks:** These build or extend their representations by acquiring knowledge from examples. The new knowledge may change the old network by adding and deleting nodes and arcs or by modifying numerical values, called weights, associated with the nodes and arcs.

- **Hybrid Networks:** This is basically a combination of two or more of the previous techniques, either in a single network or in separate, but as closely interacting networks.

Some networks have been explicitly designed to implement hypotheses about human cognitive mechanisms, while others have been designed primarily for computer efficiency. Sometimes, computational reasons may lead to the same conclusions as psychological evidence. The distinction between definitional and assertional networks, for example, has a close parallel to Tulving's distinction between semantic memory and episodic memory [Tulving and Donaldson, 1972].

Network notations and linear notations are both capable of expressing equivalent information, but certain representational mechanisms are better suited to one form or the other. Since the boundary lines are vague, it is impossible to give necessary and sufficient conditions that include all semantic networks while excluding other systems that are not usually called semantic networks.

Hence a semantic network can be defined as fundamentally a system for capturing, storing and transferring information that works much the same as the human brain. It is robust, efficient and flexible. It is also the basis for many efforts to produce artificial intelligence. Semantic networks can grow to extraordinary complexity, necessitating a sophisticated approach to knowledge visualization, balancing the need for simplicity with the full expressive power of the network. Semantic networks may be traversed via concept list views, via their relations, or by retracing the user's history.

5.4. Semantic network- general design

Semantic network as described earlier is a labeled, directed graph with nodes representing physical or conceptual objects and labeled arcs representing relations between objects. This permits the use of generic rules, inheritance, and object-oriented programming. Semantic networks are knowledge representation schemes involving nodes and links (arcs or arrows) between nodes. The links are directed and labeled; thus, a semantic network is a directed graph. In print, the nodes are usually represented by circles or boxes and the links are drawn as arrows between the circles as in Figure 19.

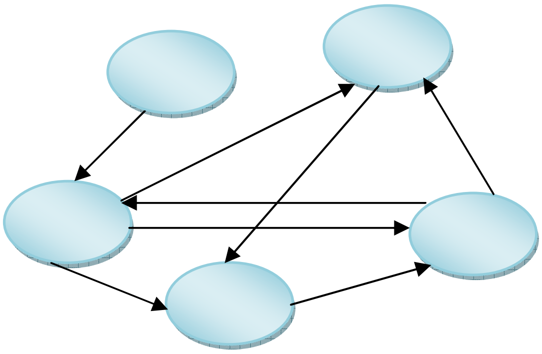


Figure 19: Semantic network structure

The above figure represents the simplest form of a semantic network, a collection of undifferentiated objects and arrows. The structure of the network basically defines its meaning. The meanings are purely which node has a pointer to which other node. The network defines a set of binary relations on a set of nodes.

A Semantic network is basically a node-link structure as nodes in the network represent concepts and the links represent the relationship between these concepts. To move semantic nets from this abstract realm to something more concrete, let us consider an example from the structure of university. To begin simply, let us introduce two nodes and a link.

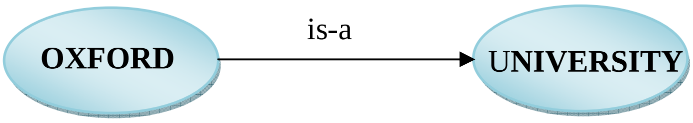


Figure 20: Semantic network depicting the is-a link

The node on the left labeled "Oxford" is linked to the node on the right, labeled "University", and the arrow is labeled "is-a". Oxford is an example of a university. The diagram, in other words represents the fact that there is a binary relation between a university, oxford, and the concept of a university. Another node with the label "Cambridge" and a "is-a" link from this node to the "University" node could be added, again representing that "Cambridge" is a type of "University".

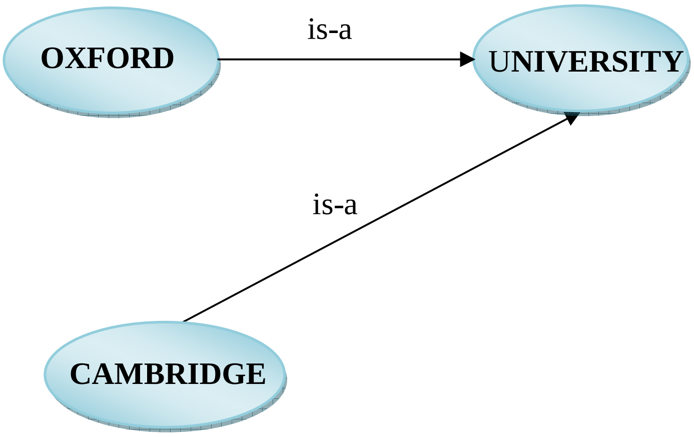


Figure 21: Semantic network depicting is-a link

If a college node is added to Figure 21, the structure of the network becomes apparent as shown in Figure 20. Universities generally contain "COLLEGE" entities. To add an example of a college, add a node labeled "CHRIST CHURCH" and two links - one from the college " CHRIST CHURCH " to "OXFORD" labeled "is-a-college-in" and one from the node " CHRIST CHURCH " to the node "COLLEGE" labeled "is-a". This illustrates that Christ Church is a college in the Oxford University.

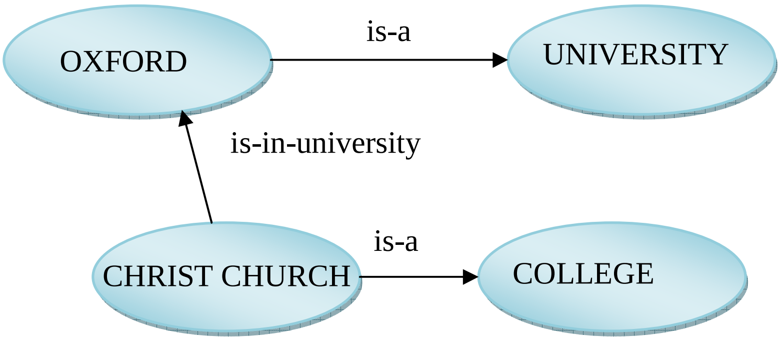


Figure 22: Semantic network showing different levels of is-a relation

It is now important to note a point or two of possible semantic confusion. Notice that the nodes in this small network are not all of the same type. The node labeled "UNIVERSITY" represents the generic or meta or class concept of a university; it represents the abstract concept of a university. It can be thought of as possessing properties common to all universities. The node "OXFORD" represents an individual instance of the node "UNIVERSITY".

The same is true of the relation between the node labeled "COLLEGE" and the node labeled "CHRIST CHURCH". The node "COLLEGE" again represents the concept of a college that is common across all particular colleges. One instance of such a college is the node labeled "CHRIST CHURCH". In order to distinguish between these two types of nodes, the class nodes become boxes and the instance nodes become ellipses, as in Figure23.

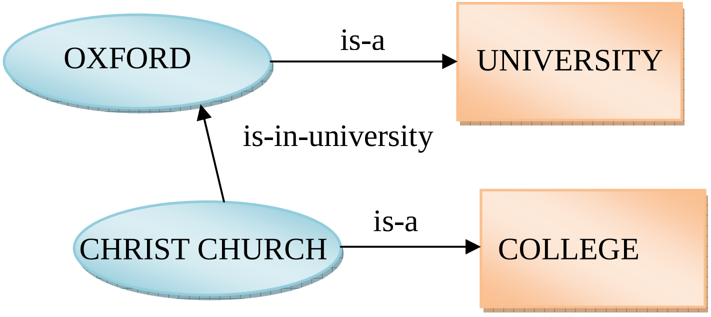


Figure 23: Different types of semantic nodes

Another class node, labeled "STUDENT", that represents the abstraction of items in a category, can now be added. Along with that, an instance “of an”, labelled "DANNY", is added. Thus, another "is-a" link and a new link, "student-enrolled-in", must be added to the node "DANNY" and the node "CHRIST CHURCH" respectively. These new additions are shown in Figure24. The information now being represented is that Christ Church is a college affiliated to the oxford university and that Christ church has student named Danny.

As the nodes proliferate, the meanings of these links need to be considered. It should become apparent that not all links are alike. Some links express only relationships between nodes, and are therefore "assertions" of the nature of the relationship between two different nodes. For example, the link "student-enrolled-in" in Figure24, which illustrates the relationship that the college Christ Church has a student named Danny. For instance, the node labeled "DANNY" is an instantiation of the class node labeled "STUDENT".

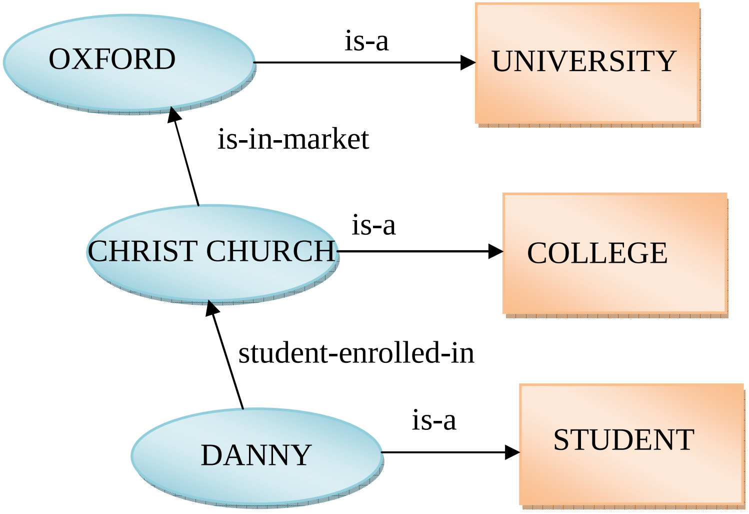


Figure 24: Semantic network different relations

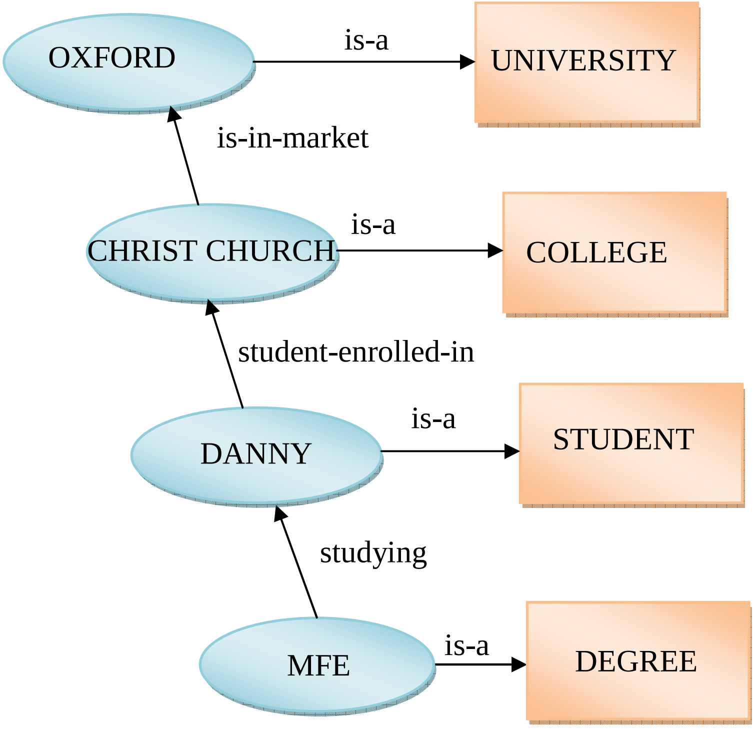


Figure 25: Semantic network with different nodes and relations

In Figure 25, more nodes and links are introduced to the original network. There is now a "DEGREE" class node with an instance node "MFE". The link "studying" conveys the information that student Danny is a student in MFE Course. Our network now has a representation for information about the student node Danny. For instance, the network above conveys the information that Danny is a student studying a degree called MFE offered in Christ Church College affiliated to the Oxford University.

Another import characteristic of the node-link representation is the implicit "inverse" of all relationships represented by the directional arrows. If there is an arrow going from one node to another, this also implies the reverse - that there is an arrow from the second node to the first. In Figure 26, there are the nodes labeled "CHRIST CHURCH" and "DANNY" with the link labeled "is-a-studen-in".

The direction of the relationship is that "DANNY is a Student in CHRIST CHURCH". Further, some linguistic terminology for our binary relationships could be used: "DANNY" is the subject and "CHRIST CHURCH" is the object, and "is-a-student-in" is the verb or action or link between them.

.

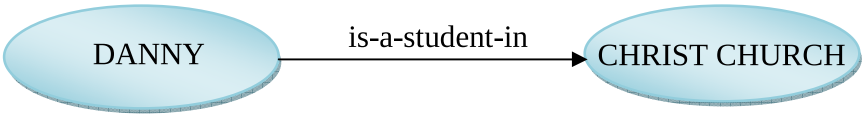


Figure 26: Semantic relationship showing an inheritance relation

This "DANNY is a Student in CHRIST CHURCH" relation implies the inverse relationship that "CHRIST CHURCH has a student named DANNY", as shown in Figure 27.

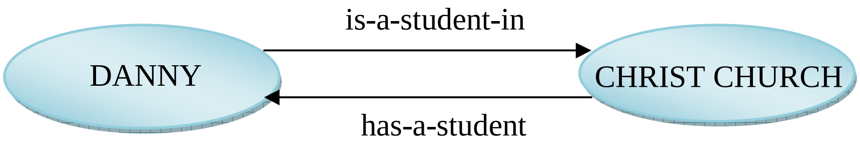


Figure 27: Inverse relations in semantic network

The representational or expressive power of semantic networks has been discussed thus far. As with any kind of knowledge representation scheme, a way of inferring knowledge that is not directly represented by the scheme is needed. The ability to work with incomplete knowledge sets a knowledge representation apart from a database. To give an example of what can be gleaned from the semantic network in Figure 25 that is not directly represented, consider Figure 28. It is an extraction of Figure 25 containing only three nodes and two links.

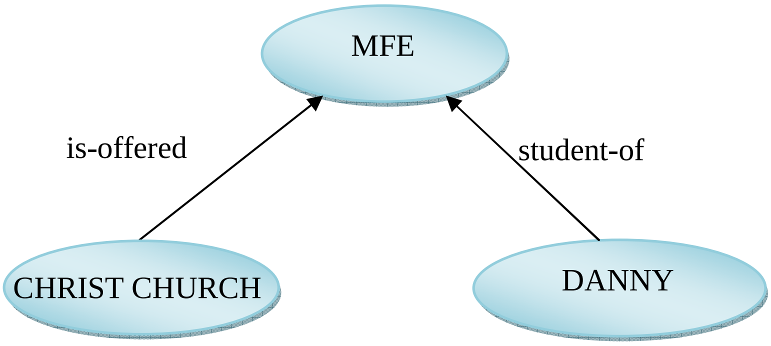


Figure 28: Partial representation

The information explicitly represented is that the student named Danny is a student of the MFE course offered by the Christ church college. The inverse relationship between Danny to MFE, i.e. that MFE is-the degree-studied by Danny is shown in Figure 29.

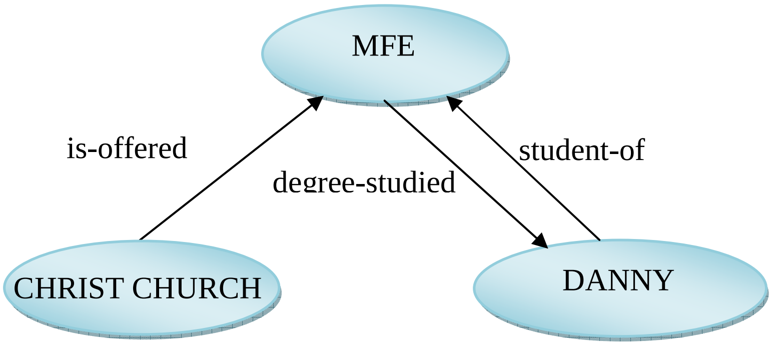


Figure 29: Inverse relation

This discussion has introduced the concept of a semantic network consisting of nodes and links with nodes representing concepts and the links representing relationships between these concepts as described earlier. The discussion also briefs the distinction existing between instance nodes and class nodes: the former representing general notions of the latter; of which there may be many types. The concept of links which extend from the instance node level to the class node level has been detailed in the above sections. It also elaborates on the reversibility feature of the relational links. The method of inferring new relationships between nodes from existing ones is also explained. Thus the discussion provides a detailed explanation on the definition and design of a semantic network.

5.5. Semantic Network Prototype Model:

5.5.1. Introduction

The past decade has witnessed a tremendous upsurge in information availability in various forms, attributed mainly to the ever mounting use of the World Wide Web (WWW). This increase in information availability has made information analysis an extremely difficult and cumbersome task. It is becoming increasingly apparent that we need machines to analyze this information for us humans. But the difficulty here is that not all information is machine understandable and consequently cannot be analyzed using machines. This is simply because of the fact that majority of this information is in the format understandable by humans only. This has made it very important to make information available in a format which can be easily analyzable by machines.

This basically requires good knowledge representation techniques which enable machines to understand and analyze information. It is therefore of paramount importance to represent these large amounts of information using efficient knowledge representation techniques. Knowledge representation is a subject in cognitive science as well as in artificial intelligence and knowledge modeling. In cognitive science it is concerned with how people store and process information. In artificial intelligence (AI) and knowledge modeling (KM) it is a way to store knowledge so that programs can process it and use it for example to support computer-aided design or to emulate human intelligence. AI researchers have borrowed representation theories from cognitive science.

There are representation techniques such as frames, rules and semantic networks which have originated from theories of human information processing. Since knowledge is used to achieve intelligent behavior, the fundamental goal of knowledge representation is to represent knowledge in a manner as to facilitate inference (i.e. drawing conclusions) from knowledge by humans as well as machines.

In the field of artificial intelligence, problem solving can be simplified by an appropriate choice of knowledge representation. Representing knowledge in some ways makes certain problems easier to solve. For example, it is easier to divide numbers represented in Hindu-Arabic numerals than numbers represented as Roman numerals.

One of the widely accepted knowledge representation techniques is the semantic network where knowledge is represented such that it is possible for machine programs to analyze the information represented by the network. A semantic network or net is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs as detailed earlier.

An example of a semantic network is WordNet, as mentioned earlier is a lexical database of English. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. Some of the most common semantic relations defined are meronymy (A is part of B, i.e. B has A as a part of itself), holonymy (B is part of A, i.e. A has B as a part of itself), hyponymy (or troponymy) (A is subordinate of B; A is kind of B), hypernymy (A is superordinate of B), synonymy (A denotes the same as B) and antonymy (A denotes the opposite of B). WordNet properties have been studied from a network theory perspective and compared to other semantic networks created from Roget's Thesaurus and word association tasks respectively yielding the three of them a small world structure.

5.5.2. Model design

In our prototype model we use semantic network as a precision model to obtain a network of concepts representing the knowledge domain or field under consideration. In our research, semantic network is basically used as a tool to increase the overall efficiency of our model. The semantic network in our research approach constitutes for a small network of concepts representing any chosen domain. The design parameters of our semantic network are almost similar to any standard semantic network design with only a few changes in certain design areas and in the relational links used in connecting these concepts.

Our semantic network model mainly contains concepts represented by nodes as is common to any classical semantic network. These concepts are then linked to one another based on the relationship they share with each other thus emerging into a network using our relational links. These links fundamentally form the arcs of our semantic network. The main idea during the design process of our semantic network model was to retain the original features of a semantic network and slightly alter relational links between these concepts. The relational links used in our semantic network model is explained in detail in the following sections.

Another important distinction of our semantic network from any other standard semantic network is the size of the network itself. This is simply because of the fact that we have decided to limit the number of concepts in our semantic network to not more than 100 nodes. This was decided inspired by the schema representation defined by the great philosopher Kant. We rationale this approach based on the thoughts reflected by the empirical concepts and their schemata defined by Kant [Eco, 1999].

Kant defines his empirical concepts as a concept of abstract thought, a thought that can be considered common to several perceptions. When an empirical concept is said to contain an object, whatever is thought in the concept must be intuited in the mental representation of the object. Examples of intuitive perceptions that are the content of empirical concepts are vague images that are imagined in order to connect a concept with the perceptions from which it was derived as their common feature.

On similar ground what we devised is our semantic network concepts as a concept of concrete thought, a thought that can only be possible when attached to a particular subject or domain. What we devised is a set of concepts of any domain that can be considered as most representative of the domain. These concepts can be considered as the heart of the domain representing each and every important aspect of that knowledge domain. We believe that to build an effective semantic network it is actually sufficient to build the network using a set of concepts considered to be most important in representing that particular field or domain.

In our semantic network prototype model we have created an entry point in the network which actually represents the center of the network. This can be compared to the mind maps modeling [Buzan, 2000] where a diagram is used to represent words, ideas, tasks etc that are linked to and arranged in radial position around a central key concept (word) or an idea. The elements in a mind map are arranged intuitively according to the importance of the concepts, and are classified into groupings, branches, or areas, with the goal of representing semantic or other connections between portions of information.

Similarly the center node in our model is considered to be the central and most important concept in the network which is later surrounded by nodes connected semantically. The center concept always bears the name of the domain itself for which the semantic network is being built. In this semantic network prototype the central concept (node) considered to be the most representing concept of the domain is given a numerical value of 1 which is the highest value assigned to any concept in the network. This numerical value is actually useful in calculating the value of each concept in the network and thus determining its importance in the entire network.

The center of our semantic network is then connected to seven different concepts called categories, which operate as the categorizing concepts in the network. These seven concepts are actually predefined concepts which are already existent in the semantic network model irrespective of the knowledge field or domain it is built to represent. These concepts can be varied depending on the topic on which the network is built. These concepts actually helps us subgroup the underlying concepts more clearly and distinctively as sub concepts or concepts that relate in meaning to the over lying categories.

5.5.2.1. Concept categories

The seven concept categories introduced by us basically acts as subdivisions of the main domain or topic for which the semantic network is being built. These seven categorizing concepts were chosen based on the advice and suggestions provided by domain experts. The concepts were chosen such that it covered all the various aspects and information concerning a topic, required is constructing a semantic network for the particular knowledge domain.

These concept categories in point of fact help us in classifying the concepts under seven broad divisions. The examples stated in the following sections are more specific to the topics or domains concerning to the research carried out by the laboratory that we have agreed to collaborate and work with. The topics are mainly related to the research in Nuclear Toxicology on living beings. But nevertheless our model is a generic model i.e., if we need to use this semantic network model on any other research topic it is possible to do so either using the same seven categories or by just modifying these seven predefined concept categories with the categories more pertinent to the knowledge domain in consideration. However, it is very important to note that the seven categories currently defined by us are such that they are largely domain or topic independent hence can generally be used in for all topics.

However, should a need arrive where the categories needs renaming then in that case the user simply needs to chose different categories that in fact summarize all the aspects that are considered important in representing a domain for which the user intends to build a semantic network. These seven categories in fact provide guidance to the user to build the rest of the semantic network even when the user possesses minimum domain knowledge.

This is simply because of the fact that the seven categories help the user to better understand and represent the connectivity/relation/meaning shared between the concepts and thus can connect the concepts appropriately. Even if the user with minimum domain knowledge simply follows the design procedure and rules set by our model, the user is sure to achieve an end result which will produce a fairly strong semantic network for any knowledge domain. This fact will actually make the model increasingly automated by reducing the time and input needed from the user. The seven predefined concepts we currently built into our semantic network prototype model, were based on the assumption that these concepts will be largely suitable as categories representing a knowledge domain, when building semantic network either for the current research collaboration topics or any other arbitrary topic. The so called categorizing concepts are as follows:

- **Disciplines:** This concept mainly groups all the related concepts that correspond to the different disciplines of the knowledge domain for which the network is built. For example, let us consider a semantic network built to represent the domain called history. Here this example is chosen randomly for ease of understanding. While considering the domain history one can easily infer that the topics like American history, Chinese history, Ancient history etc. all fall under the category named disciplines. Hence it is very easy for a user to immediately categorize these concepts as concepts related to the category discipline. It is very important to understand that these predefined topics are not definite and hence can be altered depending on the knowledge domain one is working on. In our prototype example built on Arabidopsis discipline groups concepts like bio-informatique, genetique, metabolatique to name a few.
- **Tools:** This is another subdivision in our model which basically helps in grouping concepts related to the tools and techniques that occur associated to a domain. If we consider an example of Arabidopsis semantic network (one of the networks built by us on the research topic called Arabidopsis), we can easily identify that the categorizing concept tools groups concepts like molecular biology, spectrometry, and speciation which are some of the important tools used in the Arabidopsis research.

- **Molecules:** This is the third type of subdivision and very specific to the research topics on which we are testing our model. This subdivision is mainly targeted to group concepts related to molecules. Some examples are nodes like peptides, enzymes to name a few.
- **Biological Models:** This concept basically enables grouping of all the concepts of the topic or domain which fall under the biological models category. Some example of concepts that come under the biological model for the topic Arabidopsis are genes, mutant germination etc.
- **Organisms:** This division basically groups the nodes related to organism division for example, in the case of Arabidopsis the node named plant is grouped under this category.
- **Types of Study:** This mainly groups all the nodes representing the different fields of study involved related to the research topic. In the semantic network built on Arabidopsis this division mainly groups nodes like Invitro, Invivo to name a few.
- **Technical Interest:** This division mainly identifies and groups together concepts representing all the technical aspects that are indispensable in actually representing the research topic.

The crucial task in our semantic model design is to identify the 100 concepts most representing a research topic for which the network is to be built. Once this task has been accomplished, our next goal is to actually categorize these 100 concepts under the seven predefined category concepts elaborated in the earlier section. These concepts can be actually considered as seven super classes to which all the other concepts in the network are connected based on the relation they share. The concepts are categorized based on the expert advice.

These seven concepts also carry the numerical value of 1 which is similar in value to the center node. This part of our model remains common to all semantic networks built using our model irrespective of the topic. The value 1 assigned to the center node and the seven categorizing nodes will also remain the same irrespective of the domain topic.

It is important to understand that concepts can actually belong to more than one super class concept. This is basically because some concepts represent features or functions inherited from more than one super class. In the example Arabidopsis considered earlier the node representing the concept plant actually belongs to super class organism as well as the super class called biological models.

It is also sometimes possible that the features of few concepts represent more than one super class as they exhibit inherited characteristics that can be found to be derived from different super classes. This aspect of concepts inheriting features from several different classes is termed as multiple inheritances. More specifically multiple inheritance [Meyer, 1988] refers to a feature of some object-oriented programming languages in which a class can inherit behaviors and features from more than one super class. This contrasts with single inheritance, where a class may inherit from at most one super class. Multiple inheritances [Keene, 1989] basically allows a class to take on functionality from multiple other classes thus inheriting functionalities from more than one super class.

5.5.2.2. Relational links

The arcs used to connect the nodes in our semantic network are called the relational links. We fundamentally use 4 types of relational links to represent the different relationships shared by the concepts of the semantic network. The four links chosen to be used in our semantic network model are similar to the links used in the Unified Modeling Language (UML) [UML, 2000]. It is a standardized visual specification language for object modeling. UML is a general-purpose modeling language that includes a graphical notation used to create an abstract model of a system, referred to as a UML model.

Each of these links represents a relationship that the connected nodes share. The links in our semantic network prototype are always pointed towards the super class or concept to which the other concept is connected. For example if concept B is part of concept A then the arrow head of the composition link in our model will be pointed towards concept A. Another important characteristic of the relational links in our semantic network prototype model is the fact that they are unidirectional meaning that the inverse relation does not exist in our semantic network prototype model. This has been done basically to keep the model in its simplest possible form so that the integration of models that we intend to achieve in the later stages of our research becomes less complicated.

The 4 types of relational links used in our semantic network model are as follows:

Association link

Association in UML is a relationship between two classes. Links represents the relationship between objects. Association defines how classes communicate with each other, and link represents a state of the system where an object sends some message to another. In our semantic network prototype the association link is used to represent simple relations that associate different classes with one another. Below is an example of an association link.

An association link is normally a straight line drawn between two concepts believed to be associated with one another. The association link can either be a one to one association or one to many. In the illustrated example we see that a concept named “Person” is associated to the class named “Company” by the association relation of “Employee”. Here the concept “Person” shares a simple employee association with the concept “Company”.

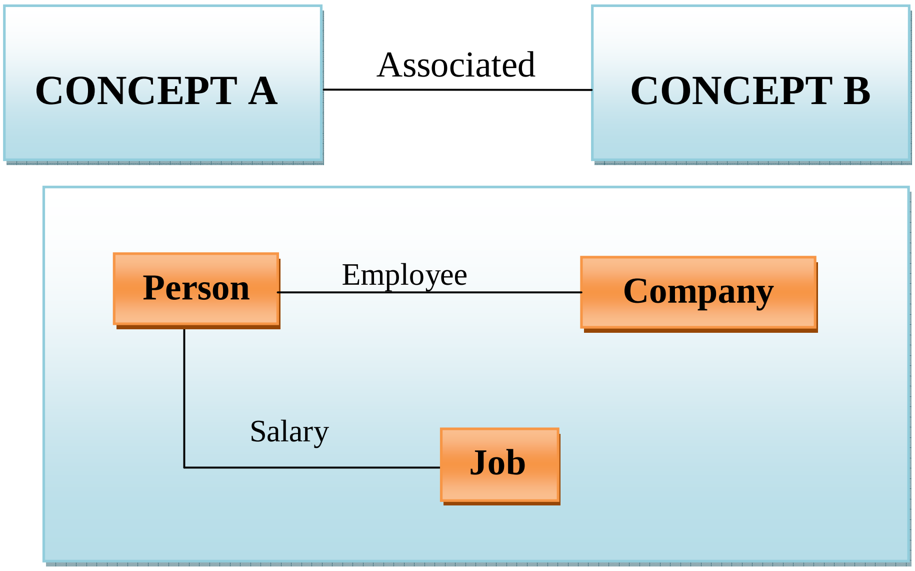


Figure 30: illustration of association relational link

Similarly the same concept “Person” can also be related to another concept named “Job” by the associative relation of “Salary” depicting a relationship where a person takes a job for the salary he or she earns. Hence the relational link of association is used to represent the simple association relation shared by concepts.

Composition link

Composition is a form of aggregation with strong ownership and coincident lifetime of part with the whole. The multiplicity of the aggregate end may not exceed one (it is unshared). The parts of a composition may include classes and associations. The meaning of an association in a composition is that any set of objects connected by a single link must all belong to the same container object.

A composition may be thought of as a collaboration in which all of the participants are parts of a single composite object.

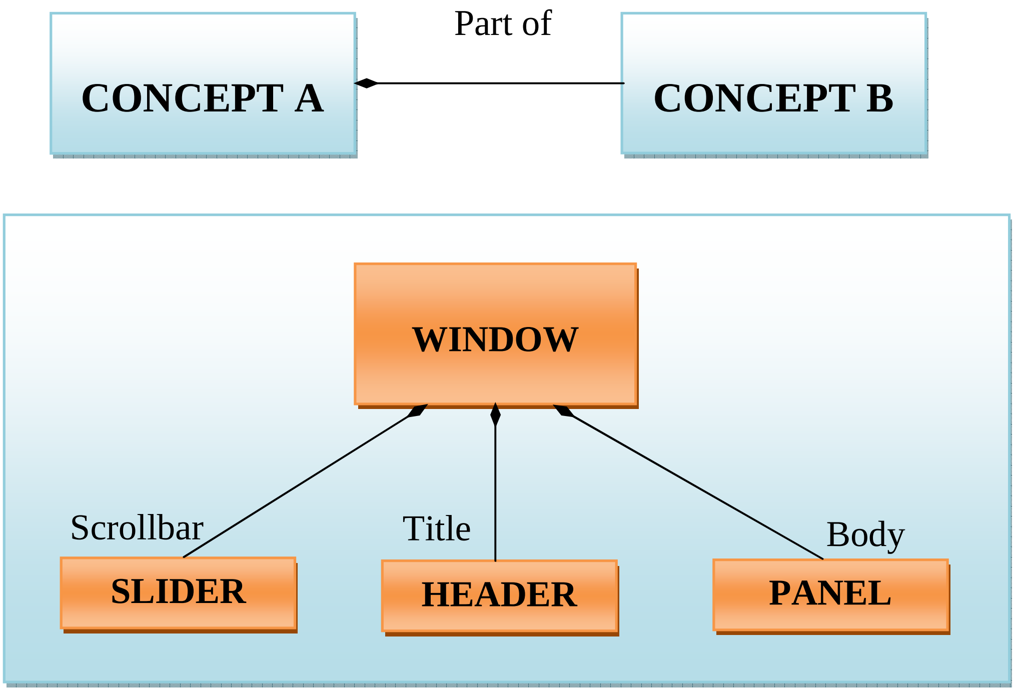


Figure 31: Illustration of composition relational link

The composition link is typically used to represent the relation between all the objects that constitutes to form one complete object. Here even if one part is deleted or moved the whole concept gets affected. Similarly, we use the composition link in our semantic network prototype to represent the relationship between a single or a set of concepts or classes that group to represent another concept or a class. The composition link in our semantic network prototype relates a single or group of concepts to another concept in the network by depicting a part-of relation between them.

This can be better illustrated using the example where a composition link is used to relate classes or concepts. In our prototype a composition is shown by a solid-filled diamond adornment on the end of an association path attached to the element for the whole. This is a widely accepted notation for composition link is many different models.

The figure 31 depicts simple composition relation shared between the concepts. Here one can see that a class “Window” is basically composed of the concepts “Slider” representing the scrollbar, the “Header” representing the title and the “Panel” representing the body. One can see that all these concepts share a part-of relation with the concept “Window” Here even if one of the compositional concepts are moved the concept “Window” losses its original representation.

Instantiation Link

In simple terms this relational link can be defined as a representation of an idea in the form of an instance of it. In programming terms, it can be defined as creating an instance of a variable using a specific value. It can also be defined as the act of creating an ‘instance’ of a generic unit by replacing its formal parameters by a set of matching actual parameters.

Instantiation is basically an identifiable occurrence or occasion of something. In object oriented programming, producing a particular object from its class template is called instantiation. This involves allocation of a structure with the types specified by the template, and initialization of instance variables with either default values or those provided by the class's constructor.

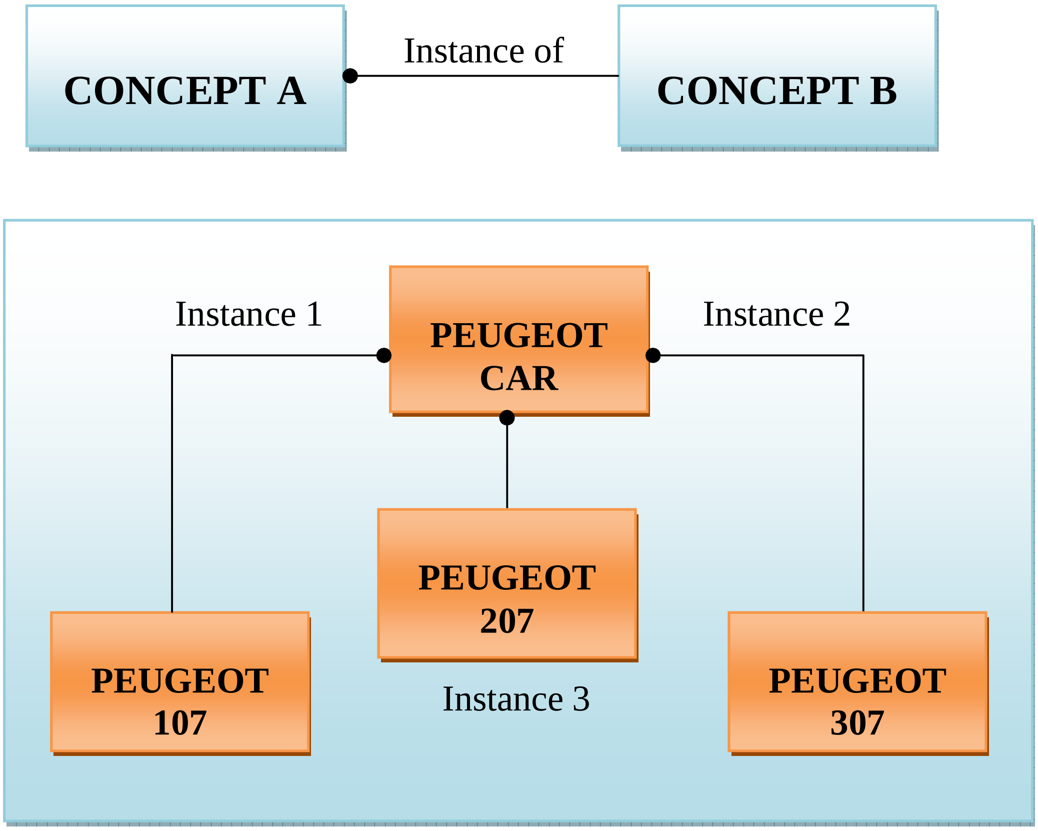


Figure 32: Illustration of instance relational link

Similarly in our prototype model instantiation relation is used in representing a particular concept which is actually an instance of another concept. In the instantiation relation the instance has the same qualities as that of the concept it has been instantiated from. The only difference here is that the instance class is either verbalized or numerated to specifically represent that particular idea at that specific instance. The meaning expressed when instantiation link connects two or more concepts in our semantic network is simply that one concept is just an instance of the other concept under given specifications.

In our prototype model we use an association path ending with a filled circle to represent the link of instantiation as depicted in the figure 32. This can be understood by the following example stated. In the figure we see that the concept “Peugeot” is linked to the concepts “Peugeot 107”, “Peugeot 207” and “Peugeot 307”. One can clearly understand that although all the instant classes are basically cars produced by Peugeot they each individually represent in particular a specific type/model or instance of a Peugeot car.

Inheritance Link

In simple description inheritance can be defined as the reception of genetic qualities by transmission from parent to offspring. In object-oriented programming, inheritance is a way to form new classes (instances of which are called objects) using classes that have already been defined. The new classes, known as derived classes, take over (or inherit) attribute and behavior of the pre-existing classes, which are referred to as base classes (or ancestor classes).

Inheritance is also sometimes called generalization, because the inheritance link actually shows the Is-a relationships which represents a hierarchy between classes of objects. For instance, a "fruit" is a generalization of "apple", "orange", "mango" and many others. One can consider fruit to be an abstraction of apple, orange, etc. Conversely, since apples are fruit (i.e., an apple is-a fruit), apples may naturally inherit all the properties common to all fruit, such as being a fleshy container for the seed of a plant.

In addition to the properties inherited from its super class the inheriting class can also have its own set of features unique to the class only. An advantage of inheritance is that modules with sufficiently similar interfaces can share a lot of functionalities, reducing the complexity of the program. Inheritance is typically accomplished either by overriding (replacing) one or more methods exposed by ancestor, or by adding new methods to those exposed by an ancestor.

There are many different aspects to inheritance. Different uses focus on different properties, such as the external behavior of objects, internal structure of the object, structure of the inheritance hierarchy, or software engineering properties of inheritance.

One common reason to use inheritance is to create specializations of existing classes or objects.

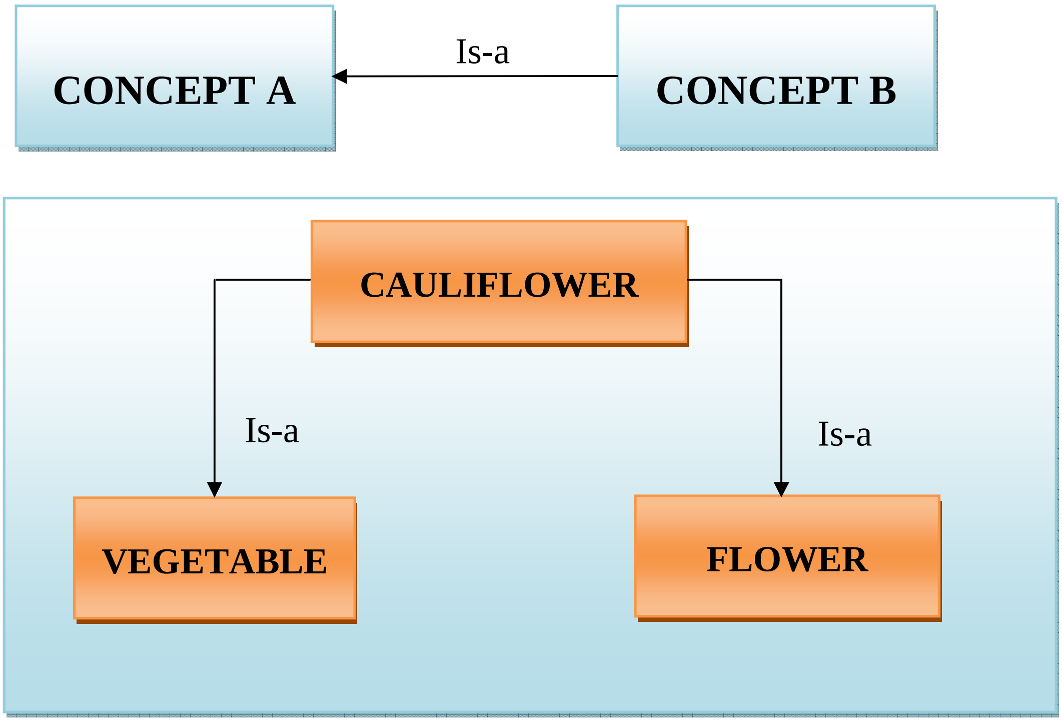


Figure 33: Illustration of inheritance relational link

The inheritance link used in our prototype model shares most functions analogous to a standard inheritance link. Our prototype permits concepts inheriting features from one or more super class concepts. Thus our prototype supports the multiple inheritance features. Multiple inheritances [Keene, 1989] refer to a feature of some object-oriented programming languages in which a class can inherit behaviors and features from more than one super class. In our prototype we use an association path ending with a filled arrow head to indicate an inheritance link showing is-a relation between concepts. The diagram shown depicts concepts inheriting features from one or more super classes.

Here the concept “Cauliflower” inherits features from both the concept “Vegetable” and “Flower”. Hence it is inheriting from more than one super class or concept, therefore depicting multiple inheritances.

Determining Values for each of the relational links:

We decided on the values denoted to each of these relational links on a completely random basis. We chose to value compositional link at 0.85 followed by the instantiation link at 0.80 and inheritance link with a value of 0.75. However, we are considering of exploring this aspect in the future perspective section of our research.

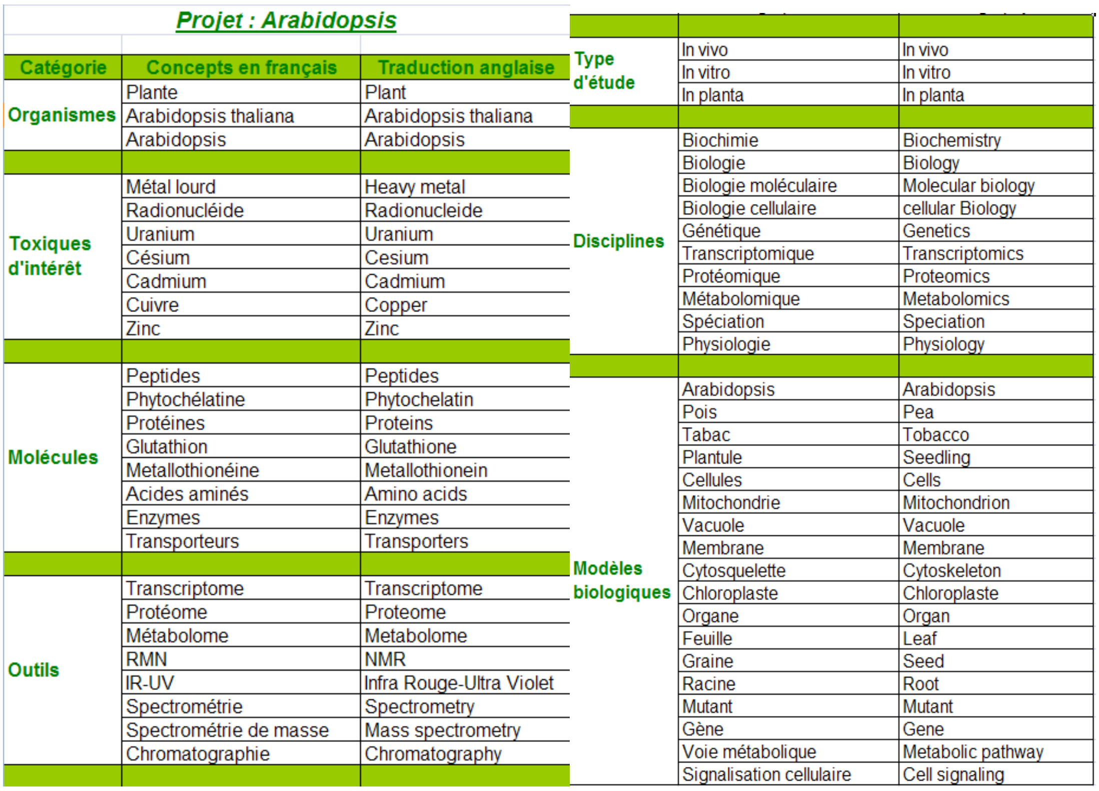
5.5.3. Semantic network construction:

Once the fundamental design of our semantic network has been finalized, the next task in our prototype building is to actually build a semantic network using our design. This is an important stage in our prototype design cycle due to the fact that our design is actually being put to test. During this stage we needed to choose a topic for building the semantic network using our design model. We decided to build our first semantic network using our design model on the topic called Arabidopsis, which is one of the main research areas on the ToxNuc-E platform.

Given that the topic was from the biology domain it was essential that we consult a specialist from the field who can help us in identifying the 100 most important concepts from the field Arabidopsis. We decided to consult the domain specialist from CEA to help us with this task. After several deliberations between our research teams, we decided to split the task into two categories. First part being the task of actual identification of the 100 concepts most representing the domain Arabidopsis and the second task is semantically linking these concepts using our design model.

We initially began with our first task by contacting some of the domain experts from CEA who were ready to spare some time in helping us find the required data to be used in our research project. Our initial task was to actually gather all the data available on the topic Arabidopsis from the ToxNuc-E database. Once we had this data we then began the analysis task. The data obtained from the ToxNuc-E database was actually processed such that it enabled us in identifying the concepts occurring maximum number of times as well as most commonly used in the entire data set.

Table 2: snapshot of the questionnaire: 7 concept categories of Arabidopsis project



Projet : Arabidopsis

Catégorie	Concepts en français	Traduction anglaise
Organismes	Plante	Plant
	Arabidopsis thaliana	Arabidopsis thaliana
	Arabidopsis	Arabidopsis
Toxiques d'intérêt	Métal lourd	Heavy metal
	Radionucléide	Radionucleide
	Uranium	Uranium
	Césium	Cesium
	Cadmium	Cadmium
	Cuivre	Copper
	Zinc	Zinc
Molécules	Peptides	Peptides
	Phytochélatine	Phytochelatin
	Protéines	Proteins
	Glutathion	Glutathione
	Metallothionéine	Metallothionein
	Acides aminés	Amino acids
	Enzymes	Enzymes
	Transporteurs	Transporters
Outils	Transcriptome	Transcriptome
	Protéome	Proteome
	Métabolome	Metabolome
	RMN	NMR
	IR-UV	Infra Rouge-Ultra Violet
	Spectrométrie	Spectrometry
	Spectrométrie de masse	Mass spectrometry
	Chromatographie	Chromatography
Type d'étude	In vivo	In vivo
	In vitro	In vitro
	In planta	In planta
Disciplines	Biochimie	Biochemistry
	Biologie	Biology
	Biologie moléculaire	Molecular biology
	Biologie cellulaire	cellular Biology
	Génétique	Genetics
	Transcriptomique	Transcriptomics
	Protéomique	Proteomics
	Métabolomique	Metabolomics
	Spéciation	Speciation
	Physiologie	Physiology
Modèles biologiques	Arabidopsis	Arabidopsis
	Pois	Pea
	Tabac	Tobacco
	Plantule	Seedling
	Cellules	Cells
	Mitochondrie	Mitochondrion
	Vacuole	Vacuole
	Membrane	Membrane
	Cytosquelette	Cytoskeleton
	Chloroplaste	Chloroplaste
	Organe	Organ
	Feuille	Leaf
	Graine	Seed
	Racine	Root
	Mutant	Mutant
	Gène	Gene
	Voie métabolique	Metabolic pathway
	Signalisation cellulaire	Cell signaling

Once these concepts were identified, they were then stored into a database. This data was later passed to all the domain experts along with a pre-designed ranking sheet as shown in the figure. This sheet actually contains 3 columns. The first column contains a listing of all the concept from the database followed by three columns stating the importance of these concepts in the domain from Highly important, important to Not important. The experts are advised to rate each of these concepts based on their level of importance in representing the knowledge domain. Based on the response of this survey we were able to extract the 100 most important concepts for each project on the ToxNuc-E platform.

Once the concepts were rated by the experts we then chose the top 100 concepts based on its ratings. This was then stored into another database for future use. Now the task was to divide these 100 concepts into the seven predefined classes in our semantic model as described in the earlier section. In order to achieve this we created another survey model where we listed the 100 concepts chosen by us based on the previous concept rating survey response and then sent another survey to all the participating domain experts to identify the class or division they think each concept should fall under.

When this survey response was received, we analyzed these responses with the help of a couple of domain experts who volunteered to help us with our analysis. We along with the domain experts analyzed the responses of the researchers based on which we actually determined the final categorization of concepts into the seven predefined divisions. Once the concepts were separated into the seven categories our next task was to relate these concepts to represent them as a semantic network. This required that we identify the relationship each concept shared with the other concepts in the network, although it is not necessary that every concept be related to every other concept present in the semantic network.

This is the most crucial part of the semantic network construction due to the fact that this stage requires complete human intervention. It is very important for us to establish the correct relations between the concepts in the network. This is the only part of the model where concepts are related based completely on human expert's knowledge about the knowledge domain. Therefore, it is very important that we consult more than one domain expert to accomplish this task. We therefore involved 2 domain experts who with our assistance built the semantic network based on our design model.

We firstly provided the domain experts with the list of concepts to be used in building the network. We later introduced them to the set of relational links that they need to use to represent the relations that they think the concepts share with each other. It was however decided that we use the compositional link to connect the center of the semantic network with the seven predefined categories. This is mainly because of the fact that these seven categorizing concepts were so chosen that they represented different aspects of the centre concept and were actually in way composing the center concept. Once the relation between concepts was established we then stored them in a data base to be used by the graph editor program for visualization of the semantic network.

The figure 34 is the semantic network built on the topic Arabidopsis visualized using the graph editor. We see that the center of the network is always represented by the concept named Arabidopsis which carries the name similar to the domain name which is also called Arabidopsis. The center of the network is then connected to seven concepts using the compositional link as seen in the illustration. One can notice that these seven concepts are the predefined categories designed by us to facilitate users of our model.

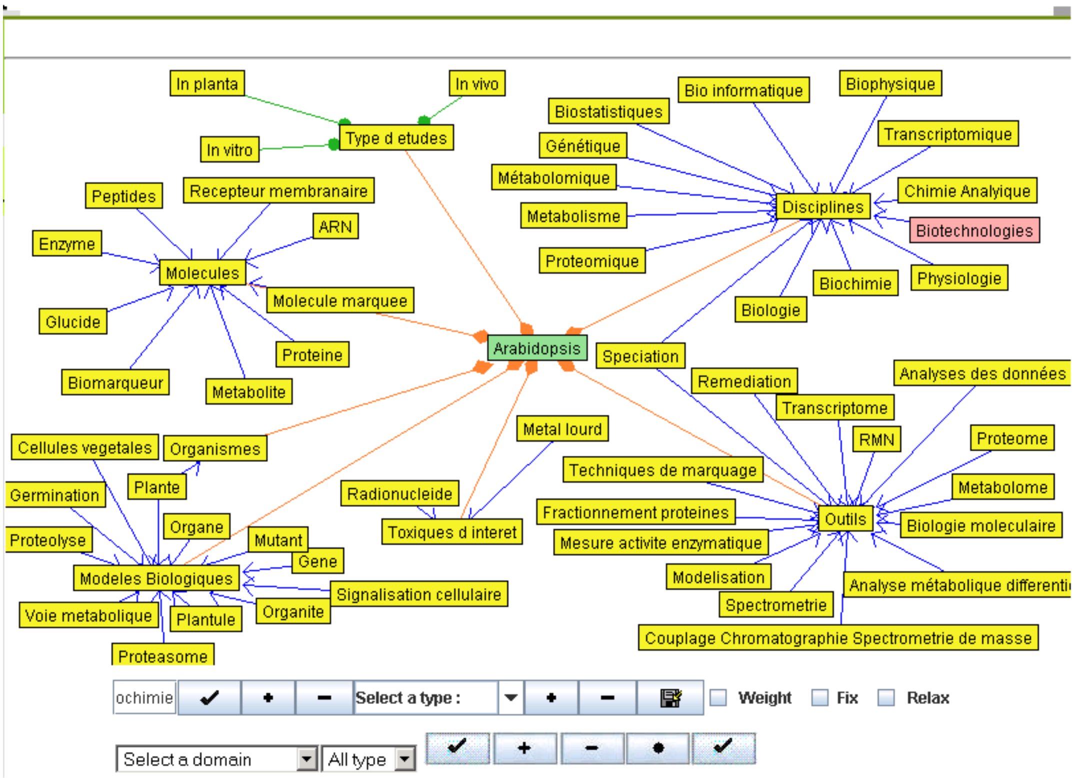


Figure 34: Semantic network on Arabidopsis visualized using graph editor

Each of these seven categories is in fact used as a connecting concept between the centre concept and the rest of the concepts present in our semantic network model. Using these seven concepts, will essentially help us categorize all the other concepts in the network very efficiently even with minimum knowledge on the domain. This as a matter of fact enables a person with very little domain knowledge to build a fairly knowledgeable semantic network using the prototype model we propose.

One can notice that the seven concepts are then connected to the rest of the concepts using the relational links provided in our model. It is important to note that a concept can be connected to one or more of these seven relational links as depicted in the figure. In the illustration we can see that the concept named “Plante” is connected to two of the seven categorizing links namely “Modeles Biologique” (biological models) as well as the concept named “Organisms” (organisms). It can also be noted that the relation it shares with both categories are similar.

The semantic model so developed is basically used in our research to form a core part of a wider network in information representation which will be detailed in later chapters.

5.5.4. Usage and Limitations:

Although our semantic network model is easy to develop and use for our users, one of the important limitation is the size of the network itself. We chose to keep our semantic network model limited to a small size with substantially small number of nodes basically for two reasons:

- Firstly we wanted to make the semantic network modeling easy and cost effective.
- The second reason being our desire to develop a model which requires minimum expert input thus reducing the construction time.

However, it is evident that the number of nodes we incorporate into our semantic model is directly related to the accuracy of our prototype results. This means that we can actually increase the precision of our model by increasing the concept limit that we have set for our model.

The second aspect is the part where we predefine seven category concepts in the network. This will actually force our semantic model users to follow the hierarchy structure from moving from an entry node towards the domain specific nodes connected to one another through our generic categorizing concept. Even with these limitations in our model, we have been able to demonstrate through our experimental prototypes that our semantic network model is a very close comparison to a classical semantic network both in structure and performance.

6. Extended Semantic Network: Hybrid model for knowledge representation

6.1. Introduction

One has witnessed an outburst in information availability ever since the arrival of the dotcom era. This has led to the ever growing concern over the problem of information flood due to the availability of increasing channels for information to flow across. The overabundance of information coupled with lack of information management techniques has created the information management and retrieval problem. The World Wide Web has been extremely successful in congregating data by providing simple tools to its users, thus encouraging more information exchange and diffusion.

Although this has been a great boon to the mankind the main downfall to this system is now the effort required in finding and identifying the required data. It has become extremely difficult for users to actually find relevant information and one tends to very easily get lost in the bundles of information provided by the internet. Hence it is of utmost importance to develop solutions which can enable machines to easily and efficiently categorize this information for its users. This will eventually enable machines to understand and categorize data leading to efficient information management and retrieval practice possible, with little difficulty.

Choosing appropriate knowledge representation formalism for building an information retrieval model can pose a challenge. The type of description can range from a highly precision based model to very detailed recall defined model. There are several knowledge representation models that make comparison of precision based and recall based models approaches. In this particular case comparisons can also be draw between models developed with complete human intervention with those been developed with minimum or no human intervention at all. The main challenge here is to identify and develop a model which can actually combine the advantages of high precision human developed model and the high recall semi-automated approaches.

We principally present a hybrid knowledge representation model called **Extended Semantic Network (ESN)** developed by us in response to the growing demand for efficient and productive automated knowledge representation techniques. These techniques are mainly required to resolve the current crisis of information management and overflow. Although there have been several knowledge representation techniques developed by various research groups and firms, finding the right information in easy and efficient way still remains a huge challenge that has been widely acknowledge.

This is mainly due to the reasons that majority of these knowledge representation techniques are firstly very expensive to build due to the high cost involved in developing them and also due to the fact that they are highly time consuming and difficult to develop.

Using the ESN knowledge representation technique we attempt to establish the idea of combining different methodologies used in developing knowledge representation models to build one hybrid model. In this hybrid model we try harmonizing different factors such as high precision, high recall, cost effectiveness, easy to build and most importantly minimizing human intervention in the process. The model mainly uses the recall factor from the machine developed model which is combined with the human developed model possessing the precision functionality.

The model chiefly targets in providing ontology like graphs with nodes and vertices representing information in a format that can be used by machines to understand these information. We also argue that it is not necessary to always have very high precision to actually obtain benefiting and satisfying end results. We also discuss about how, efficient knowledge representation techniques can be developed independent of the natural language processing techniques (NLP).

6.2. Extended semantic network prototype

There are numerous technologies existing in various domains ranging from information technology to medical science. These entire technologies target in fulfilling various tasks expected of them. They are able to achieve this by using a set of features possessed by them, which in fact represents their individual identity and usage. However, as one believes there always exists opportunity to improve the existing models by either adding new features or by combining the different already existing models / techniques with new ideas.

Consequently, hybrid models are created by the combination of one or more such technologies such that the existing technologies transform into a more sophisticated model. This is where the term hybrid appears in the discussion. In broad terms, hybrid refers to a product obtained by combining two or more different products.

There are several perspectives of the word hybrid based on the context it is used. In case of genetic studies, a hybrid is basically the result of combining elements from two or more different existing species. But when it comes to dealing with automobiles, a hybrid refers to a vehicle whose power train combines the aspects of different technologies (i.e. gasoline and electric) to improve efficiency and reduce emissions. Likewise, in case of information science a hybrid model represents techniques developed by combining and deriving functionalities from two or more different models to achieve a particular objective or goal.

Hybridization creates a marriage between different technologies leading to creation of an end product which inherit functionalities by combining available features from both the parent models. This will actually create a model possessing a combination of all the desirable qualities thus making it richer in quality and efficiency for the purpose it was developed initially. In majority of the cases hybrid models are created to mainly serve some of the purposes as stated below:

- To create new models using existing ones to actually resolve a specific issue which otherwise is not possible using either of the parent models. This will actually help gather the most desirable qualities of existing models packed into one single model helping obtain high quality end results.
- To maximize the benefits obtained by combining functionalities present in different models. This will mainly cut down on the number of different (hardware/ software) models used, thus saving on the time and cost involved in any operation.

- To chuck out all inutile contents and functionalities of a model thus retaining only the desirable functionalities of any model and thereby improvising it by adding other important functionalities.

Extended semantic network is one such hybrid model developed as a knowledge representation tool. The main idea here is to address and overcome the existing constraints in achieving efficient information classification and retrieval as discussed in the earlier sections. Our model mainly addresses the issues concerning the features like precision, recall, and human intervention level. In the proposed model we try to create a balanced match between these features and analyze the advantages and disadvantages while doing so.

Extended Semantic Network is a resulting model obtained from the collaboration between two conceptual word networks, one automatically constructed Proximal Network and the other manually constructed based on design models called the Semantic Network. Here, the primary idea is to develop an approach by formalizing a combination of features and functionalities from both man and machine theory of concept [Sowa, 1984], which can be of enormous importance in the latest information retrieval, classification, pattern matching and ontology development research.

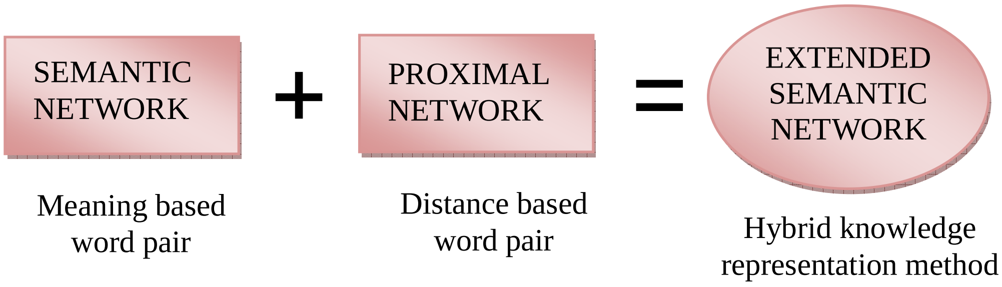


Figure 35: Schematic representation of ENS

We propose to envision and create a novel method where the data representation model is partly derived from mind modeling techniques and partly based on the mathematically operated machine model. This enables our model to inherit functionalities from both the underlying models. This is depicted by the figure**, shown below.

Our principle objective in building ESN is to combine the advantages of two different models of knowledge representation. We categorize our Semantic Network as a purely precision based model which requires considerable human intervention during its development stage. On the other hand we categorize our Proximal Network model as a recall model which is mainly developed using mathematical algorithms with minimum or no human intervention during its development process. The ultimate goal of our Extended Semantic Network model is to help information retrieval mechanisms recall information that is both accurate and relevant.

Firstly, it is very important to understand the line of balance between precision and recall factor in any efficient knowledge representation techniques. Both these factors are the most important parameters to consider while evaluating the measurement of efficacy in many existing techniques and models. However, some of the knowledge representation techniques rely mainly on the precision parameter while some other models consider recall as a more important parameter in many information search and retrieval models.

Secondly Extended Semantic Network focuses on a novel approach of using machine built network to replace actual human constructed networks that are currently used in many existing knowledge representation models. Here, the idea is to understand and analyze the extent of variations caused in the actual information classification carried out as an evaluation, using a machine constructed model with minimum human input instead of the standard human developed network.

Our basic idea is to understand the performance of a semi-automated model as against a completely human modeled knowledge representation technique. We believe that by using the combined advantages of both machine and human constructed model we achieve results similar if not better to those obtained using human constructed model. But in the process we are making our model semi-automated which acts as a huge advantage coupled with the factor that our knowledge representation model requires minimum human intervention.

Our proposal is to construct a network of concepts on similar lines of an ontology but using a method where minimal human intervention is required. We compare this to a semi-supervised ontology, representing certain qualities of ontology and this is later expatiated by adding the information obtained from the automatically developed proximal network. We propose that this method will produce similar output as any traditional ontology but will greatly decrease the construction time, attributed to its mathematically modeled extension method. Some of the major points we hope to achieve in ontology construction through our approach are

- To minimize time of construction using automated machine developed models without sacrificing on the quality of result by maintaining a good tradeoff between precision and recall.
- To make construction cost effective and productive by encouraging minimum human intervention.
- To avoid the difficulty involved in coordinating cooperation between experts and a way to avoid their disagreements.

6.3. Precision verses recall in knowledge representation models

The main idea of building a knowledge representation model is to enable machines to interpret information like humans. These knowledge representation models basically form the core of many information search and retrieval techniques. It is very important that these knowledge models are able to represent any given domain both in breadth and depth. A model should be so designed that, it is able to handle vastness of the available information as well as demonstrate accuracy while retrieving data. Developing such knowledge representation models will very much simplify information search and retrieval.

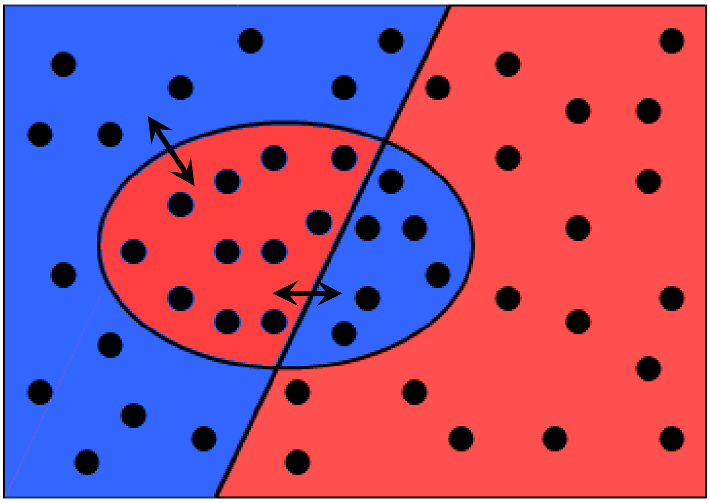


Figure 36: Precision versus Recall

In the figure2, the recall and precision depend on the outcome of a query represented by an oval and its relation to all relevant documents on the left hand side and the non-relevant documents to the right hand side. The more correct results (red), the better is the outcome.

But to build such efficient models it is very important to understand the underlying factors that actually help machines analyze and retrieve information. It is very important for any information retrieval model to firstly identify the entire available information source and then subsequently be able to extract the right data from the vast pool of information for any submitted query. Hence some of the most important evaluating factors for any knowledge representing model are to retain a high precision and recall ability. Precision and recall are two widely used measures for evaluating chiefly the quality of results in domains such as Information retrieval and Statistical classification.

Precision is a term that can have slightly different meanings, depending on the context in which it is used. It can be defined as a measure of the closeness of a series of measurements of the same material. In laboratories precision is expressed as a coefficient of variation, which is nothing more than the standard deviation divided by the mean and expressed as a percentage.

In engineering, science, industry, and statistics, precision characterizes the degree of mutual agreement among a series of individual measurements, values, or results. However, in computing, precision can be defined differently based on the context. It can be the precision of number of digits with which a value is expressed or the units of the least significant digit of a measurement; for example, if a measurement is 17.130 meters then its precision is millimeters. In evaluating the performance of information retrieval systems precision is the fraction of the information retrieved that are relevant to the user's information need.

Precision and accuracy are closely used terms thus raising confusions over their usage. While accuracy is the degree of veracity, precision can be stated as the degree of reproducibility. However, it is not possible to reliably achieve accuracy in individual measurements without precision.

$$\text{Precision} = \frac{|\ \{\text{relevant information}\} \cap \{\text{retrieved information}\}}{|\ \{\text{retrieved information}\}|}$$

Figure3: Equation of Precision

Most of the ontology and semantic network based knowledge representation models are usually high when it comes to retrieval effectiveness. This is mainly due to the quality demonstrated during the initially building process of such models where concepts are related on the lines of human reasoning. This ensures that with the guidance of such models machines are able to interpret queries similar to humans and thus provide relevant search results.

Another important parameter to be considered here is the breadth of such knowledge representation models. It is very important that these models are able to actually capture as much relevant information as possible. This ensures that for every query submitted every possible aspect is considered and thus there is no information loss due to limiting information sources. This is called the recall parameter of an information retrieval technique.

Recall is the fraction of the information that is relevant to the query that is successfully retrieved. Recall can also be defined as a measure of completeness. In binary classification, recall is called sensitivity. So it can be looked at as the probability that relevant information is retrieved by the query. Recall is the parameter that ensures that all possibly sources of information is considered and reflected as a result to a submitted query.

$$\text{Recall} = \frac{|\,\{\text{relevant information}\} \cap \{\text{retrieved information}\}|}{|\,\{\text{relevant information}\}|}$$

Figure 4: Equation of Recall

During this process it is very likely that some irrelevant information sources are also listed thus actually reducing the overall precision factor of the system. In order to achieve a good recall it is very important that the knowledge representation system actually support a huge knowledge base. This means that it should be capable of identifying all possible sources for a submitted query.

It is trivial to achieve recall of 100% by returning all available information in response to any query. Therefore recall alone is not enough but one needs to measure the number of non-relevant documents also, for example by computing the precision. Often, there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other. For example, an information retrieval system such as a search engine can often increase its recall by retrieving more documents, at the cost of increasing number of irrelevant documents retrieved thus decreasing precision.

Similarly, a classification system for deciding whether or not, say, a fruit is an orange, can achieve high precision by only classifying fruits with the exact right shape and color as oranges, but at the cost of low recall due to the number of false negatives from oranges that did not quite match the specification.

It is well accepted that a good Information retrieval(IR) system should retrieve as many relevant documents as possible i.e., have a high recall, and it should retrieve very few non-relevant documents i.e., have high precision. Unfortunately, as mentioned earlier these two goals have proven to be quite contradictory over the years. Techniques that tend to improve recall tend to hurt precision and vice-versa. Both recall and precision are set oriented measures and have no notion of ranked retrieval.

Researchers over the years have used several variants of recall and precision to evaluate ranked retrieval. For example, if system designers feel that precision is more important to their users, they can use precision in top ten or twenty documents as the evaluation metric. On the other hand if recall is more important to users, one could measure precision at 50% recall, which would indicate how many non-relevant documents a user would have to read in order to find half the relevant ones.

One such measure is average precision, a single valued measure most commonly used by the IR research community to evaluate ranked retrieval. Average precision is computed by measuring precision at different recall points (say 10%, 20%, and so on) and averaging [Salton and McGill, 1986]. Building models demonstrating a good balance between precision and recall is one of the most important criteria in the current scenario in information retrieval systems. Extended Semantic Network is one such knowledge representation model that uses the precision based Semantic Network and the recall based Proximal Network models to develop a hybrid model by combining the advantages of both the parent models.

6.4. Semi-automatic knowledge representation model

Knowledge representation models are often seen as basic building blocks for the semantic web, as they provide a shared and reusable piece of knowledge about a specific domain. With the rapid development of semantic web, the scale and complexity of knowledge representation models are growing fast. Majority of the existing models are constructed based of expert knowledge about a specific domain. Knowledge representation techniques like ontology are entirely based on expert knowledge about a domain. Here each and every concept present in a domain is carefully selected by experts and the relational links are drawn between them based on these expert opinions.

However, in most of the cases one can find that more than one expert input is required in constructing such models for broader purpose. Hence the construction of large-scale models will involve collaborative efforts of multiple developers. This means that there are several opinions to be considered while building such models. However, collaborative construction of knowledge representation models is a complicated task. The primary challenge ahead of constructing a large-scale knowledge representation model is how to harmonize different developers with different knowledge backgrounds to work together.

It is very important to understand that for efficient results it is very important that the experts involved in building such knowledge representation models agree on co-operation at different levels. It is very common to notice that there will be several disagreements of opinions among the different experts involved. This is mainly due to the fact that most of these developers posses different knowledge background and hence share different views about any domain. Hence it is a very complicated task of actually coordinating their diverse views. This becomes more complicated in case of large scale models where several different researchers from different knowledge areas are involved in building the same model.

Another important factor of concern is the cost involved in developing such large models using expert knowledge. It is a known fact that models developed using expert intervention is very expensive and not affordable to one and all. This is where the idea of creating semi-automated knowledge representation model arises.

If one is able to build these knowledge representation models using minimum expert input, where majority of the tasks are automated using different algorithms it becomes increasing affordable to build ontology at all levels. In this method by eliminating or minimizing the involvement of experts we are able to avoid the possible complications involved in handling these experts as well as bring down the cost involved to build such models. Extended semantic network explores this possibility of building knowledge representation techniques by automating their construction using mathematical models. At different levels of our research we illustrate our finding and experimental results to enable the evaluation our model against a benchmark provided by existing models.

6.5. Extended semantic network design and modeling

The idea in developing Extended Semantic Network is to identify an efficient knowledge representation method to overcome the existing constraints in information retrieval and classification. We employ a hybrid technique where mathematical models are combined with human developed concept networks. The combining process is done based on the frames model while the network is extended on the lines of graph theory.

To realize this we put our ideas into practice via a two phase approach. The first phase primarily consists in processing large amount of textual information using mathematical models to make our proposal of automatic knowledge representation model construction scalable. This is carried out by realizing a network of words mathematically computed using different statistical and clustering algorithms.

Thus creating a proximal network computationally developed, depending essentially on word proximity in documents as detailed in the proximal network chapter. This phase also involves the process of building small semantic networks developed using human intervention and expertise.

On the other hand, the second phase consists in examining carefully and efficiently the various possibilities of integrating information obtained from our mathematical model with that of the manually developed mind model. This is achieved by employing a heuristically developed method of network extension using the outputs from the mathematical approach. Here, we consider the manually developed semantic mind model as the entry point of our concept network.

This is achieved by carefully designing the combining process of the mathematical models with the human developed semantic models using the design model of Frames [Minsky, 1975] and [Brachman and Schmolze, 1985]. The second part of this process involves in actually extending the network while interconnecting the concepts from both models using graph theory [Biggs et al., 1986].

6.5.1. Design modeling:

The design process of extended semantic network primarily focuses on addressing some of the key problems faced in the existing knowledge representation models as stated below.

- The first and foremost issue in designing extended semantic network is to maintain good precision and recall level in the model. We aim to achieve this by building a very effective semantic network foundation using our semantic network model and as well build our proximal network from a set of high quality relevant documents using the mathematical models.
- The second issue is about how far our models can me automated. This entirely depends on how well we are able to integrate our mathematical model with the

mind model. Hence it is very important to understand the requirements to chose an appropriate technique that would help us achieve this. The automation will actually eliminate all the complexities involved in building such models.

- The third important factor is the time and cost involved in building knowledge representation models. We would like to provide a model which is fast, efficient, productive and also accessible by all.

To begin with our design process of extended semantic network we firstly consider the semantic network as the entry point of the entire structure, which is then extended using the proximal network. Based on our analysis of both our networks we clearly identify with the fact that our semantic network represents mostly the upper classes which have a significant position in any specific domain. Similarly our proximal network mainly represents the different instances of these classes being used in different context. We integrate our approach in two levels, as listed below:

- Frames and
- Graph theory

We use the frame system to actually draw the relation between the semantic network and the proximal network. Once this link is established we then continue to extend this model by simply employing the graph theory design.

As a first step in building the extended semantic network model, the entire semantic network developed on any specific domain is completely replicated including its central node as can be seen from the figure. This will form as the centre core of the model based on which rest of the network is built. This implies that our extended semantic network will also have a center concept with same name as the domain for which it is built. This will act as an entry point with a weight of 1 which is the highest weight given to a node in our design model.

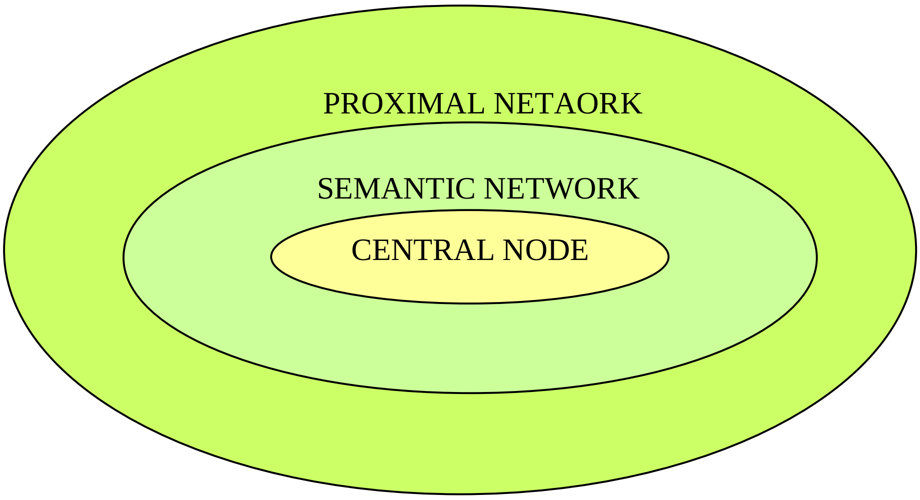


Figure 37: Extended semantic network design

The reason for choosing our semantic network model as the core part of our extended semantic network is basically to establish the following requirements:

- First and foremost reason is for the fact that our semantic network is uniquely built by experts based on their extensive knowledge about the domain. This means that each node and the relational link shared between nodes in this network are carefully chosen after proper expert analysis. This will automatically make it a very reliable network with a good precision factor.
- The second point to consider is that the semantic network is a small network with a maximum of 100 nodes. Each of these nodes carries the most representing concept of any specific domain in consideration. Thus they clearly represent the most representing concepts of that domain.
- The other important reason is the nodes in the semantic network. These nodes are so weighted that they all carry a value which is always above 0.5 (50). This will provide very reliable weights to later nodes when a calculation is involved while using this model in any search or retrieval tool.

All the concepts present in the semantic network are considered as super classes which will actually guide us in connecting the nodes from the proximal network. The theory used in connecting the nodes from proximal network on to the semantic network core; based on frames structure is detailed below.

Frames were initially proposed by Marvin Minsky in his 1974 article "A Framework for Representing Knowledge". He explains that a frame is an artificial intelligence data structure used to divide knowledge into substructures by representing stereotyped situations. Frames are connected together to form a complete idea. The frame contains information on how to use the frame, what to expect next, and what to do when these expectations are not met. Some information in the frame is generally unchanged while other information, stored in terminals, usually change. Different frames may share the same terminals.

A frame's terminals are already filled with default values, which are based on how the human mind works. For example, when a person is told "a boy kicks a ball," most people will be able to visualize a particular ball (such as a familiar soccer ball) rather than imagining some abstract ball with no attributes.

According to Minsky one can think of a frame as a network of nodes and relations. The "top levels" of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many terminals– which represent "slots" that are filled by specific instances or data. Here, slots are properties describing the Frames. Each terminal have the ability to specify conditions its assignments must meet. Collections of related frames are linked together into frame-systems.

For visual scene analysis, the different frames of a system describe the scene from different viewpoints, and the transformations between one frame and another represent the effects of moving from place to place. For non-visual kinds of frames, the differences between the frames of a system can represent actions, cause-effect relations, or changes in conceptual viewpoint. Different frames of a system share the same terminals; this is the critical point that makes it possible to coordinate information gathered from different viewpoints.

An example is KL-ONE [Brachman and Schmolze, 1985] a well known knowledge representation system in the tradition of semantic networks and frames; representing a frame language. The system is an attempt to overcome semantic indistinctness in semantic network representations and to explicitly represent conceptual information as a structured inheritance network.

Frames in KL-ONE are called concepts. These form hierarchies using subsume-relations; in the KL-ONE terminology a super class is said to subsume its subclasses. Multiple inheritances are allowed. Actually a concept is said to be well-formed only if it inherits from more than one other concept. All concepts, except the top concept, must have at least one super class.

In KL-ONE descriptions are separated into two basic classes of concepts: primitive and defined. Primitives are domain concepts that are not fully defined. This means that given all the properties of a concept, this is not sufficient to classify it. They may also be viewed as incomplete definitions. Using the same view, defined concepts are complete definitions. Given the properties of a concept, these are necessary and sufficient conditions to classify the concept.

The slot-concept is called roles and the values of the roles are role-fillers. There are several different types of roles to be used in different situations. The most common and important role type is the generic RoleSet that captures the fact that the role may be filled with more than one filler.

However in case of extended semantic network we partially follow the frame structure by representing the concepts derived from our semantic network model as a frame based model with nodes and relations structure. These nodes always remain fixed and hence represent the top level of the frames system. The semantic network model will constantly remain as the core of the entire model. However the later levels of our frame system are actually created using the nodes obtained from the proximal network model. These nodes actually represent terminals in our model which are connected to its preceding levels. The upper levels are represented by the nodes from our semantic network prototype.

The idea here is develop a design system based on which the two models can be easily converged to obtain our extended semantic network model. This will actually help us in automating a very large part of our knowledge representation model using machine built model which is incorporated with the semantic network model built using expert knowledge. This automated processing involved in our model will actually help us immensely in reducing the complications that are normally involved in building such knowledge representation models involving several domain experts.

However there are a set of predefined design procedures that is followed while building our extended semantic network model. These are predefined after extensively analyzing the requirements of our model. They actually help us decide which proximal node should be connected to which semantic node. These design procedures are detailed in the technical design section of this chapter.

The extended semantic network is basically a network obtained by extending our semantic network model by using the nodes from our proximal network model. However, once the initial combining is made using the frame structure we develop our model based on the graph theory design.

In mathematics and computer science, graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects from a certain collection. A graph in this context refers to a collection of vertices or nodes and a collection of edges that connect pairs of vertices. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another.

Alternative models of graph exist; for instance a graph may be thought of as a Boolean binary function over the set of vertices or as a square (0,1)-matrix. A vertex which is the basic element of a graph is simply drawn as a node or a dot. The vertex set of G is usually denoted by V(G), or V when there is no danger of confusion. The order of a graph is the number of its vertices, i.e. |V(G)|.

An edge which can be defined as a set of 2 elements is drawn as a line connecting two vertices, called end-vertices, or endpoints. An edge with end-vertices x and y is denoted by xy. The edge set of G is usually denoted by E(G) or simply E. The size of a graph is denoted by the number of its edges, that is |E(G)|.

Applications of graph theory are primarily, but not exclusively, concerned with labeled graphs and various specializations of these. Structures that can be represented as graphs are ubiquitous, and many problems of practical interest can be represented by graphs. The link structure of a website could be represented by a directed graph: the vertices are the web pages available at the website and a directed edge from page A to page B exists if and only if A contains a link to B. A similar approach can be taken to problems in travel, biology, computer chip design, and many other fields. The development of algorithms to handle graphs is therefore of major interest in computer science.

A graph structure can be extended by assigning a weight to each edge of the graph. Graphs with weights, or weighted graphs, are used to represent structures in which pairwise connections have some numerical values. For example if a graph represents a road network, the weights could represent the length of each road. A digraph with weighted edges in the context of graph theory is called a network. This aspect of representing networks using graphs is the functionality that interests us.

In our extended semantic network design we use the definition of vertices to represent the concepts nodes of our network. Similarly we use the definition of graph edges to actually represent the relational links connecting the nodes. Our design also uses the property of directed graph where the edges are directed by clearly stating its start and end point. In our model we use unidirectional property of the graph theory where all our edges are directed from one concept to the end concept node, thus making it's a directed knowledge representation network.

The other import feature of graph theory used in our model is the ability of these directed edges to actually carry weights representing the level of relation the connected nodes share. These weights help us represent the importance of each node in our knowledge representation model, especially when used in search and retrieval tools. Thus once we have been able to establish a connection between the semantic nodes with that of our proximal nodes using frame system we then enlarge our extended semantic network with the nodes obtained from the proximal network based on the graph theory system. Since the proximal network built using mathematical models is usually a very large network, it is very likely that our extended semantic network represents an infinite graph with a very large number of vertices and edges.

6.5.2. Technical design

This section entails the technical details involved in building our extended semantic network. We begin our design process using the 2 database tables containing the semantic network output and the proximal network output. Each of these tables has five columns each. In case of proximal network the first two columns represent the word pair, the third column represents the relational edge they share while the fourth and fifth column represent the proximity and distance the word pair share with one another. Similarly the semantic network has the first three columns similar to that of the proximal network table whereas the last 2 columns both represent the weights of the edges.

We firstly begin our design by simply copying the entire table of semantic network into the database table of extended semantic network which again has 4 columns representing the word pair, relational edge and the weight shared by the word pair. Once the semantic network is copied into the new table, we then start the process of connecting our proximal network nodes to the semantic network nodes. This is done by identifying the connecting words between the two tables.

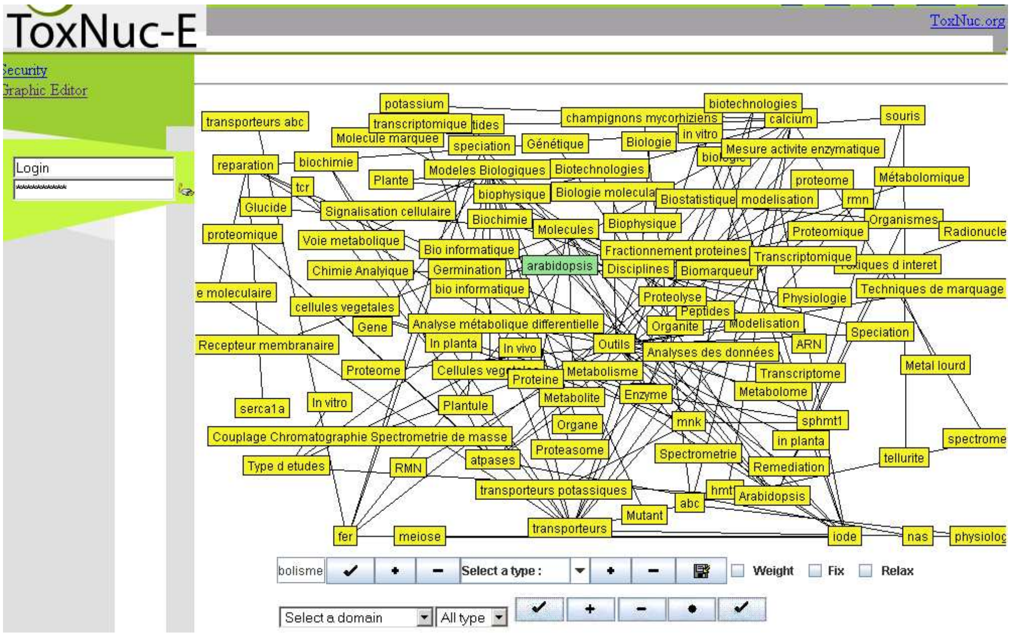


Figure 38: Extended semantic network visualized using graph editor

The above graph represents the graphical view of the data obtained by combining the results from Proximal and Semantic network.

We start with the word in the first column and first row of our ESN table and compare it with the proximal network table. If we find the word matching then we carry out a breadth and depth finding by collecting all the word nodes in the process as well as their relational values. All these nodes are then stored in the ESN table with the proximity value they share with one another.

However if in any place we find more than one value for any particular word pair we consider an average of this value and store this average as the end result. It is also important to note that while building this relation between nodes we make sure that every node is assigned a level number such that no relation can be draw from a lower level node to a upper level node while the vice versa is allowed as illustrated in the below figure number. Where S0, S1 etc represent the word level and the arrows show the possible paths allowed.

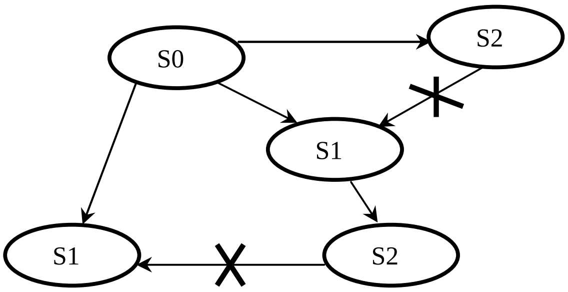


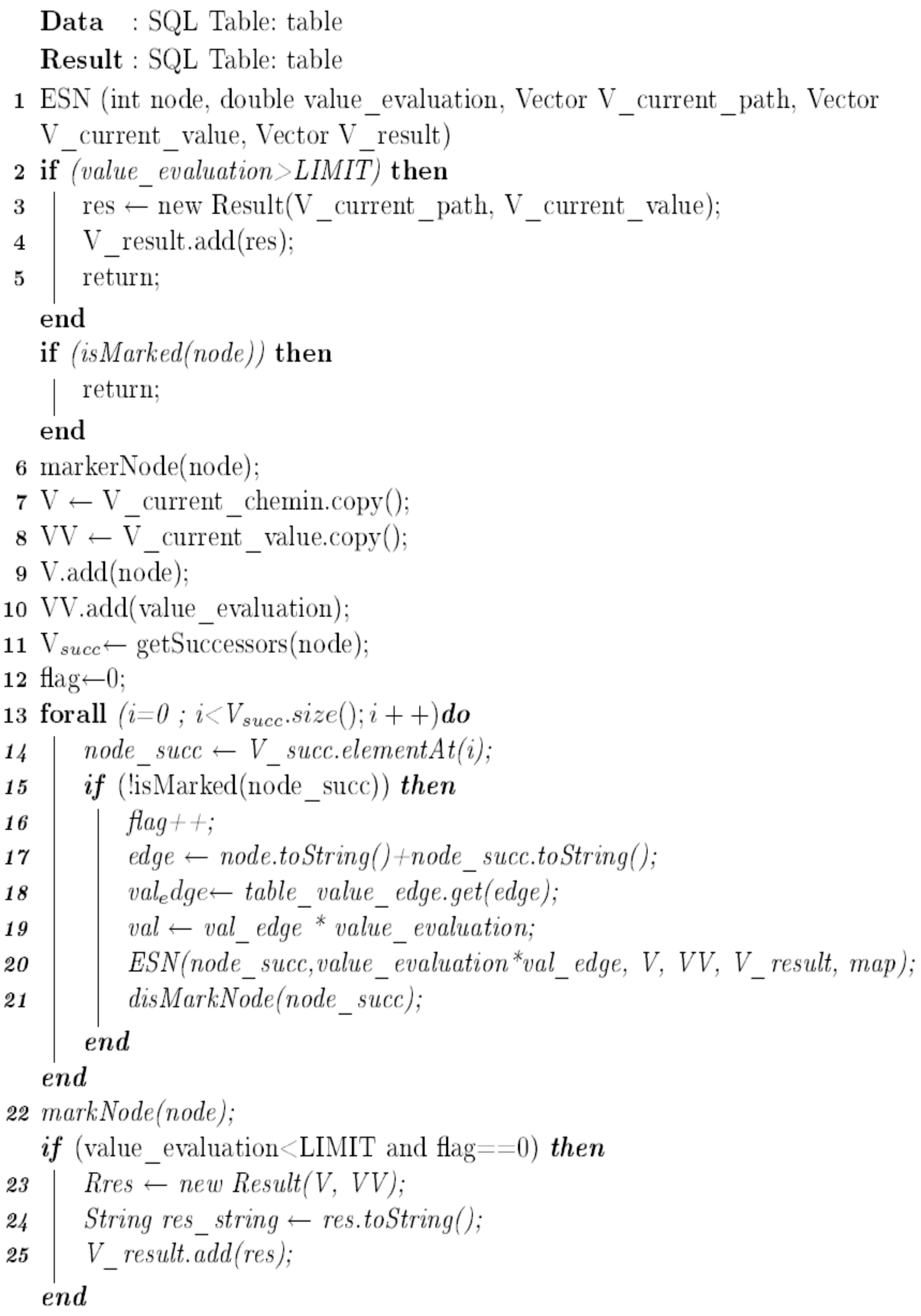
Figure 39: Relational flow illustration

For example, we start from the base word level which we tag as level 0. We then add on the next level word to the level 0. Hence there is possibility of level 0 connecting to a word of level 1 but not vice versa.

In the ESN algorithm shown below explains the central idea of how the ESN network is constructed using the data obtained from the semantic network and the proximal network. The algorithm mainly creates the extended semantic network by extending the semantic network with the data used from the proximal network. The main objective of this algorithm is to return all the possible paths that satisfies value(path) > LIMIT. By doing this we are actually increasing the depth of the network using the data obtained from the proximal network.

In line 2 we obtain the result path for a word. This is achieved by setting a LIMIT until which all the obtained paths are retained. This is added to the paths

ESN Algorithm-



```
   Data   : SQL Table: table
   Result : SQL Table: table
1  ESN (int node, double value_evaluation, Vector V_current_path, Vector
   V_current_value, Vector V_result)
2  if (value_evaluation>LIMIT) then
3      res ← new Result(V_current_path, V_current_value);
4      V_result.add(res);
5      return;
   end
   if (isMarked(node)) then
       return;
   end
6  markerNode(node);
7  V ← V_current_chemin.copy();
8  VV ← V_current_value.copy();
9  V.add(node);
10 VV.add(value_evaluation);
11 $V_{succ}$← getSuccessors(node);
12 flag←0;
13 forall (i=0 ; i<$V_{succ}$.size(); i++)do
14     node_succ ← V_succ.elementAt(i);
15     if (!isMarked(node_succ)) then
16         flag++;
17         edge ← node.toString()+node_succ.toString();
18         $val_edge$← table_value_edge.get(edge);
19         val ← val_edge * value_evaluation;
20         ESN(node_succ,value_evaluation*val_edge, V, VV, V_result, map);
21         disMarkNode(node_succ);
       end
   end
22 markNode(node);
   if (value_evaluation<LIMIT and flag==0) then
23     Rres ← new Result(V, VV);
24     String res_string ← res.toString();
25     V_result.add(res);
   end
```

Algorithm 6 : Extended semantic network

-obtained earlier to find the entire depth. The line function in line 5 backtracks all the nodes that are marked to show the path followed while finding the depth of the network.

But when doing so incase the program comes across a node which satisfies the LIMIT set but not yet marked then the line 6 in the algorithms enables us to identify such nodes. The line 11 in the algorithm helps us to identify the successive nodes and then the calculations are repeated of them. In case there are no more successive nodes to mark then the algorithm goes directly to line 22.

Thus the extended semantic network is formed using the results of SN and PN. This ESN network was mainly developed on the knowledge domains concerning the ToxNuc-E platform as the entire data sets used in our experimentation was provided by the ToxNuc-E project. Hence the main objective was to use our ESN model to develop several ESN networks for different knowledge domains on the ToxNuc-E platform.

6.6. ESN in ToxNuc-E

Once the designing of the extended semantic network was achieved the next stage was to test its effectiveness. This required that we develop models based on real time data using our knowledge representation technique. Since our research work initially started with the goal of helping the ToxNuc-E platform manage its high volume of information flow, we decide to build our prototypes on 2 of the 15 domains from the ToxNuc-E research platform namely,

- Arabidopsis Thaliana
- MSBE

The entire document set and information required for building these prototypes were provided by the ToxNuc-E platform. The main objective was to use our extended semantic models to build tools on the platform that would facilitate information management and retrieval. Some of the applications of our extended semantic network on the platform are,

- **Semi-Ontology like networks:** The extended semantic network model can be used to develop ontology like knowledge representation network on any specific knowledge domain, which can have a important role in the research activities carried out both within as well as outside the ToxNuc-E platform. It provides a cost effective semi-ontology like networks that can be developed automatically using a set of documents by any person with or without any knowledge on the domain. The so developed networks can be used by several research groups on various projects. They have several application like used as a knowledge network to understand the domain and represent a domain, to be used in tools such as search engines, classifiers etc to name a few.
- **Document Classifier:** These knowledge networks while used in combination with classification tools would help the platform in automatically managing the information by enabling automatic classification against the 15 listed projects of every new document entrant.
- **Virtual Library:** In this application for every document, ESN computes an n-dimensional vector, where n is the number of ToxNuc-E projects. A row of one vector is a value depicting the degree of interest of the document for the associated project. Now the aim is to visually display the similarities between documents with respect to their vectors. That means, we want an image where documents are represented by dots, and if two dots are close, it means that the two documents are similar. This application is called Virtual document library in Molage. It takes in entry the documents in the form of the set of n-dimensional vectors and computes a distance matrix between documents using Euclidian distance. Each document is assigned to a dot. Then it iteratively searches a configuration of the dots on the plane that respects distances between documents. So between two dots, there is one "ideal" distance, which is the Euclidian distance computed between the two documents, and the "observed" distance, which is the actual distance between the dots on the plane.

  The goal is to find a configuration of the dots that minimize the differences between "observed" and "actual" distances. In order to find this configuration, positions of dots are updated as if they were attached with each other by springs. This technique is called "Multidimensional scaling using Force directed placement". When the process is finished, the user can identify clusters of dots on the plane which represent clusters of similar documents.

Among the above listed applications we would like to present the results obtained from the document classifier. This results obtain through this experiment will serve as an illustration of the role of extended semantic networks as a knowledge representation technique.

6.6.1. Document classifier design and construction

The extended semantic network in itself is a knowledge representation technique which can help in information search and retrieval while used along with other search and classification tools. In order to apply our extended semantic network model to practical use we developed a document classifier which will use the information obtained from our extended semantic network in indexing any new document.

This document classifier uses the extended semantic network knowledge model to classify documents based on their inclination to any specific topic knowledge domain that are listed in the document classifier’s database. Some simple methods of occurrence are used to actually calculate the inclination of the document being processed. Every input document is firstly processed by analysing its contents and matching them with our extended semantic network sets.

This will enable our document classifier to actually identify the domain / domains that the document represents or belongs to. This process will also come with a list of inclination of the new document against the listed extended semantic networks in the document classifier database. This is because the document classifier mainly looks for inclination of any new document entry against a set of extended semantic networks stored in its data base. Each of these extended semantic networks actually represents the specific area or research domain it has been built on. The document classifier uses the results obtained from the domain inclination analysis to decide which domain / domains the new document most represents and should be listed under.

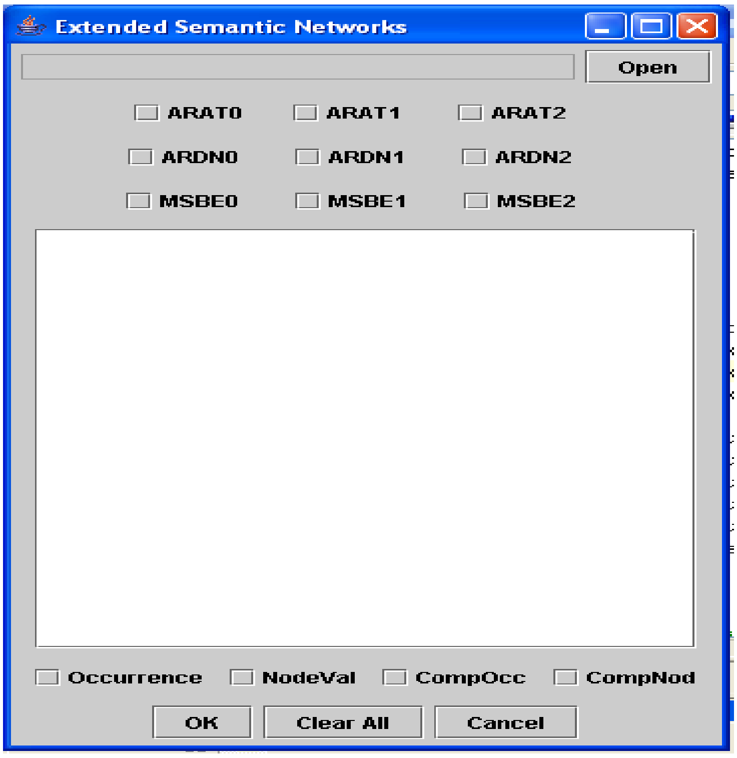


Figure 40: Document classifier

A step by step explanation detailing the functioning of our document classifier tool used to calculate the domain inclination of any new document is stated below:

- First step in the document classifier is to build the set of extended semantic networks each representing a specific domain of the ToxNuc-E platform. Once the extended semantic networks are ready, they are stored into the data base of the document classifier tool.

- The next step is to provide a new document to the document classifier, to be classified under one or more of the 15 projects carried out in the platform.

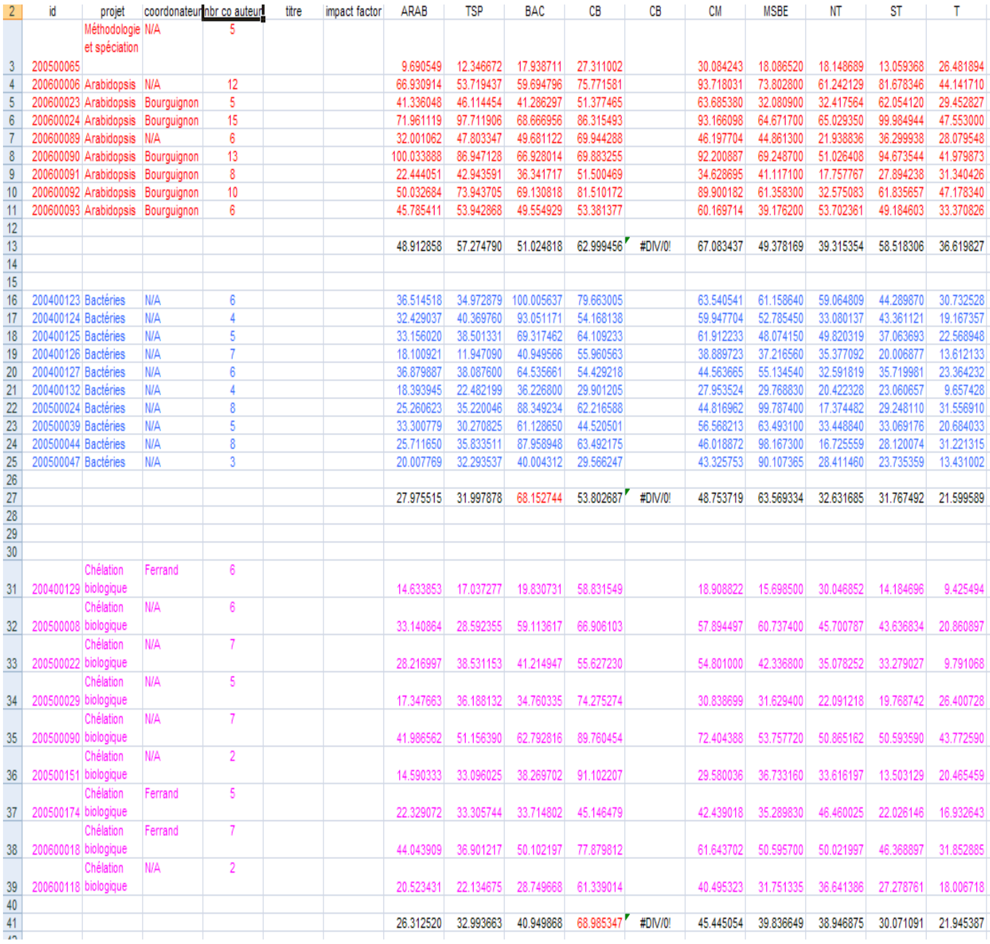
- The document is firstly analysed and compared with all the 15 ESN networks using the 4 functions namely Occurrence, NodeVal, CompOcc and Comp Node as show in the figure. These functions basically calculates the frequency of occurrence of each of the word concepts present in the network in the given document and later calculates the value of its matching using the value given to each of these word concepts in the ESN network.

- Finally the document classifier calculates the percentage of belonging for the given document against the 15 ESN networks and classifies the document accordingly.

- This classification of documents using our ESN networks actually enables us to show our users what the document is actually about and to what percentage is it addressing a field of interest without actually reading the entire document. This will significantly reduce time of processing new documents as well as provides an automated approach to easily handle large amounts of document that are loaded by the platform users.

- Although this application has been currently built mainly focusing on the ToxNuc-E platform users, however it can be easily extended to any other platform. Similarly the ESN network can be used on any topic and with any such application. The figure number shows a snapshot of the results displayed by our document classifier on the platform ToxNuc-E.

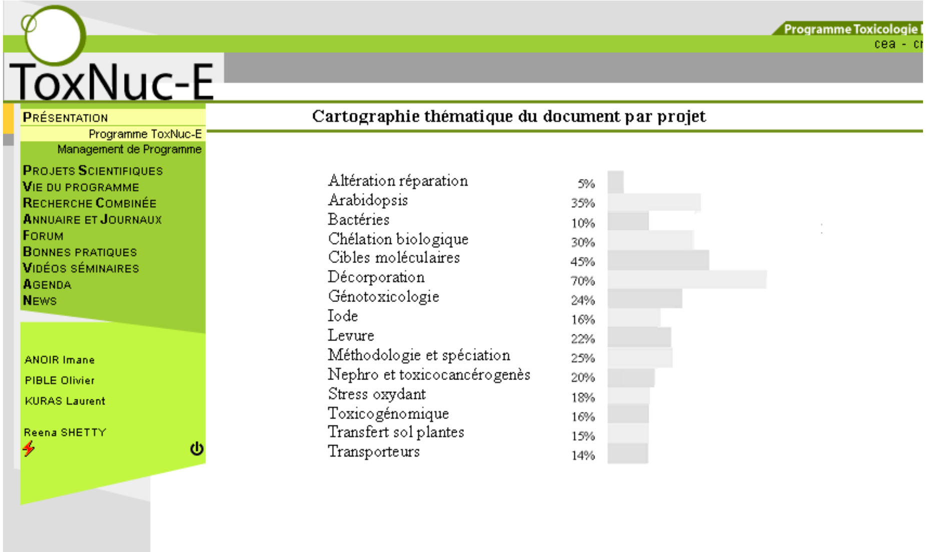
Table 3: Snapshots of the document domain inclination using document classifier



2	id	projet	coordonateur	nbr co auteur	titre	impact factor	ARAB	TSP	BAC	CB	CB	CM	MSBE	NT	ST	T
3	200500065	Méthodologie et spéciation	N/A	5			9.690549	12.346672	17.938711	27.311002		30.084243	18.086520	18.148689	13.059368	26.481894
4	200600006	Arabidopsis	N/A	12			66.930914	53.719437	59.694796	75.771581		93.718031	73.802800	61.242129	81.678346	44.141710
5	200600023	Arabidopsis	Bourguignon	5			41.336048	46.114454	41.286297	51.377465		63.685380	32.080900	32.417564	62.054120	29.452827
6	200600024	Arabidopsis	Bourguignon	15			71.961119	97.711906	68.666956	86.315493		93.166098	64.671700	65.029350	99.984944	47.553000
7	200600089	Arabidopsis	N/A	6			32.001062	47.803347	49.681122	69.944288		46.197704	44.861300	21.938836	36.299938	28.079548
8	200600090	Arabidopsis	Bourguignon	13			100.033888	86.947128	66.928014	69.883255		92.200887	69.248700	51.026408	94.673544	41.979873
9	200600091	Arabidopsis	Bourguignon	8			22.444051	42.943591	36.341717	51.500469		34.628695	41.117100	17.757767	27.894238	31.340426
10	200600092	Arabidopsis	Bourguignon	10			50.032684	73.943705	69.130818	81.510172		89.900182	61.358300	32.575083	61.835657	47.178340
11	200600093	Arabidopsis	Bourguignon	6			45.785411	53.942868	49.554929	53.381377		60.169714	39.176200	53.702361	49.184603	33.370826
12																
13							48.912858	57.274790	51.024818	62.999456	#DIV/0!	67.083437	49.378169	39.315354	58.518306	36.619827
14																
15																
16	200400123	Bactéries	N/A	6			36.514518	34.972879	100.005637	79.663005		63.540541	61.158640	59.064809	44.289870	30.732528
17	200400124	Bactéries	N/A	4			32.429037	40.369760	93.051171	54.168138		59.947704	52.785450	33.080137	43.361121	19.167357
18	200400125	Bactéries	N/A	5			33.156020	38.501331	69.317462	64.109233		61.912233	48.074150	49.820319	37.063693	22.568948
19	200400126	Bactéries	N/A	7			18.100921	11.947090	40.949566	55.960563		38.889723	37.216560	35.377092	20.006877	13.612133
20	200400127	Bactéries	N/A	6			36.879887	38.087600	64.535661	54.429218		44.563665	55.134540	32.591819	35.719981	23.364232
21	200400132	Bactéries	N/A	4			18.393945	22.482199	36.226800	29.901205		27.953524	29.768830	20.422328	23.060657	9.657428
22	200500024	Bactéries	N/A	8			25.260623	35.220046	88.349234	62.216588		44.816962	99.787400	17.374482	29.248110	31.556910
23	200500039	Bactéries	N/A	5			33.300779	30.270825	61.128650	44.520501		56.568213	63.493100	33.448840	33.069176	20.684033
24	200500044	Bactéries	N/A	8			25.711650	35.833511	87.958948	63.492175		46.018872	98.167300	16.725559	28.120074	31.221315
25	200500047	Bactéries	N/A	3			20.007769	32.293537	40.004312	29.566247		43.325753	90.107365	28.411460	23.735359	13.431002
26																
27							27.975515	31.997878	68.152744	53.802687	#DIV/0!	48.753719	63.569334	32.631685	31.767492	21.599589
28																
29																
30																
31	200400129	Chélation biologique	Ferrand	6			14.633853	17.037277	19.830731	58.831549		18.908822	15.698500	30.046852	14.184696	9.425494
32	200500008	Chélation biologique	N/A	6			33.140864	28.592355	59.113617	66.906103		57.894497	60.737400	45.700787	43.636834	20.860897
33	200500022	Chélation biologique	N/A	7			28.216997	38.531153	41.214947	55.627230		54.801000	42.336800	35.078252	33.279027	9.791068
34	200500029	Chélation biologique	N/A	5			17.347663	36.188132	34.760335	74.275274		30.838699	31.629400	22.091218	19.768742	26.400728
35	200500090	Chélation biologique	N/A	7			41.986562	51.156390	62.792816	89.760454		72.404388	53.757720	50.865162	50.593590	43.772590
36	200500151	Chélation biologique	N/A	2			14.590333	33.096025	38.269702	91.102207		29.580036	36.733160	33.616197	13.503129	20.465459
37	200500174	Chélation biologique	Ferrand	5			22.329072	33.305744	33.714802	45.146479		42.439018	35.289830	46.460025	22.026146	16.932643
38	200600018	Chélation biologique	Ferrand	7			44.043909	36.901217	50.102197	77.879812		61.643702	50.595700	50.021997	46.368897	31.852885
39	200600118	Chélation biologique	N/A	2			20.523431	22.134675	28.749668	61.339014		40.495323	31.751335	36.641386	27.278761	18.006718
40																
41							26.312520	32.993663	40.949868	68.985347	#DIV/0!	45.445054	39.836649	38.946875	30.071091	21.945387

For instance, let us consider a database containing several documents on a particular topic. A user needs the best 20 documents related to this topic from the database containing several hundred documents. If the user has to go through every document to find the best 20 results it will take several weeks of work and something which is highly impractical. But by using our approach the same can be achieved in few hours time without requiring for the user to actually posses specific knowledge on the domain.

When the same documents classified by our document classifier were manually classified it not only took considerably more effort and time but the manual classification could not provide the information of how many different knowledge domains the document might be addressing. We noticed that the results by our classifier highlighted information about certain documents belonging to the original domain Arabidopsis showed inclination to other domains like MSBE a detail not specified until and unless the document is completely read by the user. This information was seen missed by the manually classified result. The correctness of our classifier result was demonstrated to the domain experts over several meetings and was validated and approved by them for its effectiveness.



Cartographie thématique du document par projet

Altération réparation	5%
Arabidopsis	35%
Bactéries	10%
Chélation biologique	30%
Cibles moléculaires	45%
Décorporation	70%
Génotoxicologie	24%
Iode	16%
Levure	22%
Méthodologie et spéciation	25%
Nephro et toxicocancérogenès	20%
Stress oxydant	18%
Toxicogénomique	16%
Transfert sol plantes	15%
Transporteurs	14%

Figure 41: An example of document indexation as represented on the ToxNuc-E platform

6.7. Experimentation and Validation

We conducted an experimental analysis on the results given by the document classifier to validate its performance. Firstly, a set 5 scientific document series were chosen from the ToxNuc-E document database. These documents were later classified against 9 domains from the ToxNuc-E project using the document classifier. The percentage inclination results obtained is as shown in the figure below.

The figure shows 5 document series classified against the domains Arabidopsis (ARAB), Transfert sol plantes (TSP), Bactéries (BAC), Chélation biologique (CB),Cibles moléculaires(CM), Méthodologie et spéciation (MSBE), Nephro et toxicocancérogenèse

(NT), Stress oxydant (ST) and Transporteurs (T). The document classifier is able to highlight the inclination of each of these documents against the domains specified. Here one not only identifies a single domain that the document represents but can also identify other domains the document might be related to as can been seen.

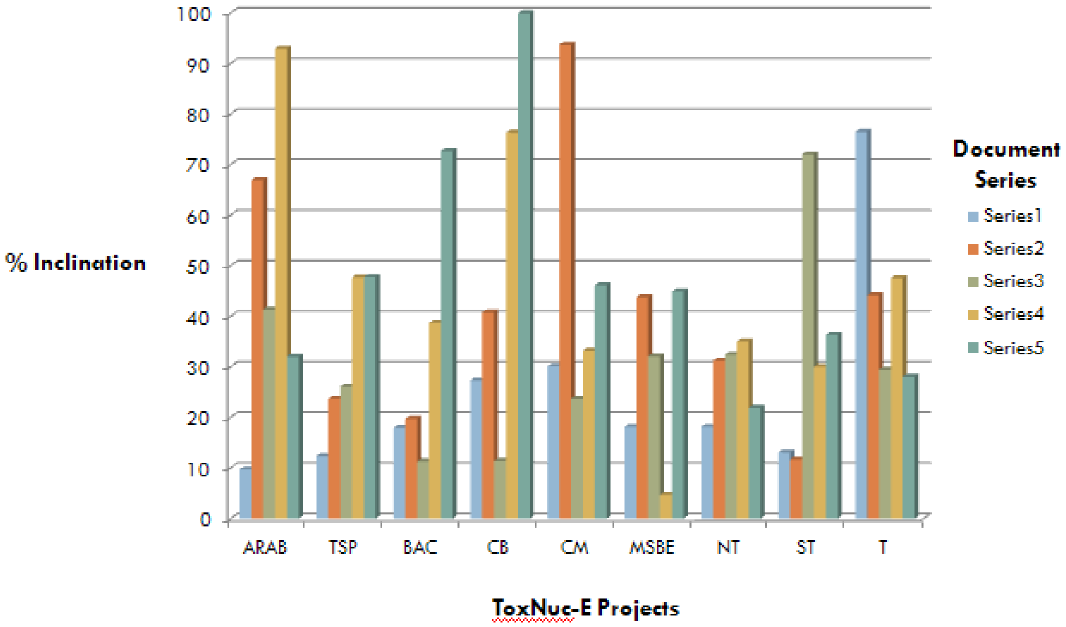


Figure 42: Domain inclination % of new documents on ToxNuc-E calculated using the document classifier

We later classified the same set of documents with the help of domain experts. During this classification the experts were able to identify the domain the document most represented but however over looked the information each document held about other domains. The domain the experts identified as most matching for each of these documents when compared with our document classifier results proved to be the same but the results from the document classifier also identified the information the documents contained related to the other domains.

In the above illustrated graph representing the output obtained using a document classifier we see how each document is classified based on the percentage inclination it has against each domain name. In the example let us consider the document represented by series 4. One can clearly see that this document has the highest domain inclination for Arabidopsis and least inclination for the domain MSBE. Hence it can be clearly inferred that the document is mainly related to Arabidopsis domain and is of interest to researchers working in this domain. However our document classifier results also highlights the information that the document has on other domain. In this example we see that the series 4 has Information related to CB, TSP and T domains. This helps highlight the information related to other domains that an expert from a different domain can easily overlook.

Another important aspect of the document classifier is the fact that it not only is accurate as an human expert but also faster than an expert. We conducted an experiment where we recorded the time taken by human as compared to our document classifier to analyze a given scientific document and the results were extremely positive.

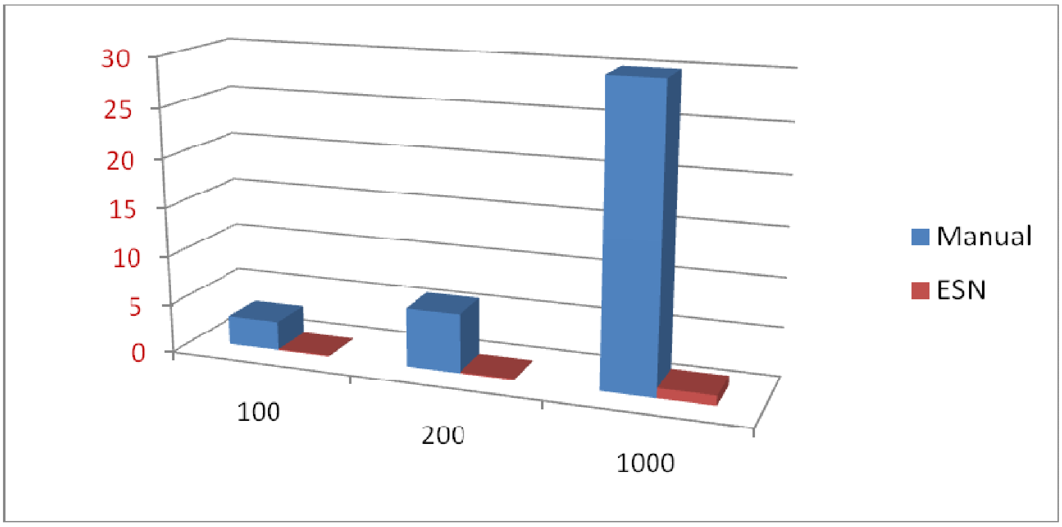


Figure 43: Prototype time comparison in classifying new documents

The graph shows the time taken by the document classifier as compared to a human in analyzing and classifying a new document. The experiments were carried out for sets of 100, 200 and 1000 pages. One can clear see that the time taken by the document classifier is considerably lower as compared to the time taken by an human expert to classify the same.

6.8. Conclusion

Through our extended semantic network model we try to explore the possible ways of combing human intelligence with machine computation in order to improve overall efficiency and productivity of knowledge representation models. Through our experiment results we attempt to illustrate the fact that using hybrid models like extended semantic network one might obtain results far more satisfying not only in terms of efficiency alone but also in the overall productivity of the task. We thereby encourage more and more researchers to explore this approach in different fields and harvest the advantages of this approach.

7. Conclusion and perspectives

In this chapter we conclude this thesis report by presenting a summary of the various contributions our research model is able to offer and also present the possible future prospects of our research work. This chapter has been broadly divided into two parts. The first part mainly illustrates the principle contribution we have been able to make and also presents the different topics we have researched on. The second part of this chapter largely concentrates on the future perspectives that our research work promises and the possible developments of the work.

7.1. Contribution

Throughout this research report we have made an attempt in highlighting the various problems and shortcomings in the current existing models and the methods that are being employed in knowledge representation and retrieval processes. One of our primary goals right through our research work is to propose models and solutions in making knowledge representation a reliable automated process. We have explored the various possibilities of replacing expert intervention by employing data processing methods that are entirely supported by machines using mathematical models.

The question on knowledge representation, management, sharing and retrieval are both fascinating and complex, essentially with the co-emergence between man and machine. This research report presents one such novel collaborative working method, specifically in the context of knowledge representation and retrieval. The proposal here is to make ontology construction cost effective, faster and easy to design. In this division we envisage and highlight the advantages of adopting our methodology as compared to the existing methods and models. We explore the prospect of introducing an innovative approach of integrating machine calculations with human reasoning abilities.

After having carried out a thorough research and study on the different existing approaches and models in knowledge representation techniques, we were able to more specifically identify the problem existing in making knowledge representation an automated process.

More particularly we concentrated on exploring ways of increasing the involvement of machines and subsequently minimizing human intervention in the process.

However as seen in chapter 2 we carried out a thorough research on the various knowledge representation models that currently exist and propose to develop a model that is automated requiring minimal human intervention. We mainly concentrate our research on exploring possible techniques that would help machines to analyze and interpret data more efficiently and in a cost effective manner. We then put forth our approach that addresses the issues of how the current knowledge representation models can harvest the advantages of the mathematical model that helps in analyzing large amounts of data in considerably small time and using this approach we build our first model called the proximal network. We mainly position this model as the part of our research work that largely contributes in enabling us to automate our data analyzing. The proximal network using various statistical models creates a network of words based on proximity of words in any given document. This model basically represents our machine model with no human intervention required in actual positioning of the word concept and deciding on its distance from one another. We build our proximal network model using the documents provide by the ToxNuc-E platform. Although our model can be generalized and can be used on any set of documents, in our current prototype model we have mainly used documents from the platform in all our experimental results.

In the Semantic network chapter we primarily study the existing semantic network models and propose a customized model sufficing our goal of making it requiring minimal expert intervention. This model although developed by experts has been customized such that it will require very little input from humans for the model to work. Our main goal here was to customize the existing techniques into models that can be easily used by our users with very little or no domain knowledge to develop Semantic networks. We use our prototype to illustrate examples of different topics derived from the ToxNuc-E platform. This model basically being considered as human developed model in our approach forms the heart of our model with acts as a building base of the knowledge representation model we propose.

In chapter 5 we argue the importance of automating the design process of knowledge representation models. We show how essential it is to generate automated / semi-automated models providing satisfactory results and thus be used in helping machines analyze and understand information to helps us better manage information. This chapter basically introduces our idea of finding a way between approaches which are either completely automated but might not efficient enough or completely human developed but not economic enough. We try to identify a balance between these 2 approaches by accommodating and combining both models. This approach as we named is the extended semantic network which actually forms and grows from a small semantic network with limited nodes into a vast word network with on an average of about 90,000 interconnected word concepts. We basically use simple techniques to enlarge small semantic networks into bigger word networks based a few defined conditions to enable automatizing of knowledge representation models.

We use the precise, non estimated results provided by human expertise in case of semantic network and then merge it with the machine calculated knowledge from proximal results. The fact that we try to combine results from two different aspects forms one of the most interesting features of our current research. We view our result as structured by mind and calculated by machines. The main objectives that we intend to address through this approach are:

- Exploring the possibilities of designing an automated approach for knowledge representation with minimum human intervention.
- Presenting models that would enable semi-automated or automated networks.
- Developing models that would make knowledge representation efficient, easy, cost effective and fast.

We were able to illustrate the applications of our proposed model through the tools such as document classifier and virtual library as detailed in the earlier chapters. We basically show the possible ways of how existing knowledge representation methods which are completely dependent on expert knowledge can be eventually replaced with automated systems that would be able to bail out experts intervention to a large extent. We make an effort to draw attention to the fact that it is not necessary that all knowledge representation models be developed by experts themselves. Infact a right combination of human and mathematical models would actually provide a better result while considered in a wider scenario.

We open doors for future research in this regard by providing few results through our tools such as document classifiers that have been used on real time data and whose results have been actually used in classifying documents on the ToxNuc-E platform and have proven to show considerable consistency and accuracy. It has also helped us open doors n facts where our tool is able to analyse a document against a set of domains as specified by the user. This will infact allow considering research into a new area where a document originally belonging to one particular domain might contain information that might be useful to a researcher belonging to another research domain or field of study. This possible overlap of knowledge we are able to identify and illustrate through our various experiments form a very promising area of research to explore further.

The ToxNuc-E presently with around 660 researchers registered with their profile, background and area of research interest are physically distanced in different geographical locations. Our research is applied in this platform to provide these researchers knowledge representation tool like ESN which can be utilized in information retrieval specifically in limiting the information they desire to obtain and also in identifying information specific to their interests. As explained earlier we have basically carried out our experiment on 15 different domains of this platform and developed a prototype for each one.

The results of our algorithm have been used to illustrate our finding rather than actually evaluate it against other models. However we have been able to demonstrate that our approach is able to produced very large and vast word networks in very small time and the user can develop such networks on any domain he desires with actually requiring very little knowledge on the domain itself. Our experiments also display the fact that the vastness of the extended semantic network is such that it displays a very high recall factor on any particular domain of interest.

Another important factor of our network is the simple computations that we in point of fact employ and this makes our model not only faster but very simple to use and easily adaptable into different tools. We demonstrate through our experimental prototypes that the extended semantic network is able to provide results on similar lines to that of NLP based models for indexing but without actually involving the heavy computations that normally NLP-based models are based on. In our approach if a user needs specific information on any specific subject it is enough to change the input documents for the proximal network. Based on these documents the entire network is reconstructed in a time span of 30 minutes.

7.2. Perspectives and future work

The question on knowledge representation, management, sharing and retrieval are both fascinating and complex, essentially with the co-emergence between man and machine. This research presents a novel collaborative working method, specifically in the context of knowledge representation and retrieval. The proposal is to attempt at making ontology construction faster and easier. The advantages of our methodology with respect to the previous work, is our innovative approach of integrating machine calculations with human reasoning abilities.

7.2.1. Hybrid: combining machine results with human expertise

One of the important aspects in our research with potential for future work is finding a more reliable approach and an effective method for combining the two different results one derived from the machine model and the other from the semantic model. As presented previously we currently extended our semantic network using the results from our proximal network model based on the graph theory approach. However it will be a very interesting aspect to research into and find possible algorithms that can be used in mapping these two results in a more effective manner. Some of the areas we would suggest and would like to research on are mapping techniques specially used in neural networks, hybrid models and other similar models.

An additional area with scope for future research is the way we combine different results from different statistical models in our proximal network model. Due to the several constraints that guided our research work we were unable to actually explore the various possibilities that one would experiment in combining these results. We have chosen to ignore the fact that each of these statistical model with its unique approach of analyzing the data from the documents that are input, independently calculate results that noticeable differ from the one another. We currently have chosen to use a simple mean calculation technique to find a combination of all the different statistical models in our proximal network. However, it will be of great interest to explore the various possible combinations that could be applied on these results to find the most effective way of finding the combination between them.

7.2.2. User specific modeling- Personalising search and classification

Multi-user environments provide the necessary tools to allow individuals to communicate and share information. Examples of such environments can be found in computer supported collaborative work, learning management systems, communities of common interests, and peer-to-peer systems. Due to the great number and diversity of users and types of information, the system should facilitate users' interaction. Supporting users in multi-user adaptive environments requires an understanding of the interaction that takes place, which is shaped not only by the individuals' characteristics, but also the group members' individual behaviours, their relationships, and the dynamics of their interaction.

The new information to be represented includes information about the users and groups, and the collaboration and relationships between users. These models could then be used for different purposes (e.g., supporting collaboration, supporting group awareness in multi-user environments and sustainability of groups) and in different areas (e.g. collaborative environments, communities of common interest or multi-agent environments) [Sheth and Maes 1993].

User modeling is one way of obtaining predictive evaluation of real-world tasks by trying to represent some aspect of the user's understanding, knowledge, intentions or processing. And there are many different techniques that are used to build user models. User models can be divided into the following three categories:

- Hierarchical representation of the user's tasks and goal structure.
- Linguistic and grammatical models.
- Physical and device level models as this appear in the field - Adaptive Document.

The first category deals with the issue of formulation of goals and tasks. The second category deals with the grammar of the articulation translation and how it is understood by the user. The third category deals with articulation not at the high level of human understanding but at the human motor level.

We see a great opportunity in exploring the various methods to construct a user model prototype based on the data we obtain from the ToxNuc-E website. We intend to monitor the behaviour; interests and research works carried out by the members of ToxNuc-E Platform and then build a model unique to each user. This model in fact builds a profile for each user and in turn stores the details obtained onto a database. These details are utilized to better understand the user requirements thus helping the user in efficient data management, sharing and retrieval.

We envisage an enormous prospect in combining the user modeling technique into our extended semantic network model to actually customize our knowledge representation model to each user on the platform based on the user's behavioural pattern. This will enable us to narrow down and actually identify the research interests of each and every user on the platform and thus provide them with better service by making information of their interest readily available to them through their profiling. We find this a very challenging and interesting future prospective for our research model.

7.2.3. Semi-automated Ontology network

We have been able to demonstrate the ability of our models to develop large word networks using statistical models. We have also been able to use these networks as a knowledge representation model in tools used for classifying documents and information. However it will be very interesting if we are able to validate our approach of developing large semi-automated networks that can actually replace ontology use in tools and techniques that require knowledge representation design models displaying a combination of recall and precision. This fact is what distinguishes us from the original ontology. By using our model the users not only are able to find a good recall and precision combination but are also able to use our knowledge representation model to develop such networks without actually possessing any knowledge about the domain. Our model promises to provide a good alternative to any classical ontology in terms of not only efficiency and recall factor but also in making it simple to design and build and making it highly cost effective due its automated feature.

We would consider in actually continuing our research in the future in this area to carry out more experiments to validate our approach and evaluate it in comparison to classical ontology models.

7.2.4. Finding inter–relation and over lapping between domain subjects

Another important aspect that our research highlights is the possibility of using knowledge representation models like extended semantic network to not only categorize information but also highlight the similarities between information that are otherwise considered unconnected.

We have been able to show initially illustrations of this fact through the experiments that we carried out on ToxNuc-E platform using our document classifier. As detailed previously we were able to highlight the fact that certain documents although classified as belonging to a particular field of study might always contain information about other research fields that might be of interest to different researchers. Using our extended semantic model we were able to easily (in terms of effort and cost) develop several ex ended semantic networks and use them to actually not only classify documents based on their contents but also demonstrate a percentage of their inclination towards each subject. This helped us in highlighting the inter-relations between documents and research fields which otherwise in majority of the cases are over looked or completely ignored. We believe that there is considerable scope in pursuing a more detailed research in this particular aspect to validate these findings in a more concrete manner by carrying out more experiments and research.

7.2.5. Collaboration and Sharing specific to ToxNuc-E

Our research has been able to demonstrate based on our knowledge representation model on ToxNuc-E platform that users when provided with convenient tools are more willing to share their research work and findings with their fellow researchers. This is for the simple fact that it feels more convincing and convenient when they are provided with authentic ways for them to evaluate their work as well as be able to stay connected with the rest of the research community. They see this as an opportunity to enhance their knowledge rather seeing it as a barrier where they will have to face difficulties in managing safe information exchange. We identify this as a very interesting research area for future research activities that research groups mainly involved in collaboration and sharing should explore.

It will be very encouraging for users when provided with easy and efficient tools that will help persuade the different research groups to share information with the rest of the research community. Encouraging people to collaborate on topics of their research interest on platforms such as ToxNuc-E which helps bring in research groups geographically distanced seems a very important factor in collaborative research. This approach makes collaboration stronger and more convincing when supported with tools and models facilitating information diffusion and sharing.

On an overall context we have mainly aimed through our research to highlight the importance of designing automated models in the current scenario of information overflow. We have also attempted simultaneously to touch base on the various current short comings that we have been able to identify during the course of our research work. In this section we have basically summarise the facts that one could follow up on from where we have left for further deliberation.

Bibliography

[Aberg, 2001] Aberg J. and Shahmehri N. “User Modeling an Aid for Human Web Assistants”, User Modeling 2001: 8th International Conference, UM 2001, Southaven, Germany, 2001.

[Albertazzi, 1996] Albertazzi L., “Formal and material ontology”, in: Roberto Poli & Peter Simons (ed.) – “Formal Ontology” – Kluwer, p. 199, 1996.

[Amaravadi, C. S, 2005] Amaravadi, C. S., “Knowledge Management for Administrative Knowledge,” Expert Systems, 25(2), pp 53-61, USA, May 2005.

[Arpirez et al., 2003] Arpirez J. C., Corcho O., Fernandez-Lopez M. and Gomez-Perez A., “WebODE in a nutshell”, Published in AI Magazine 24(3): pp 37-47, USA, 2003.

[Bachimont, 2001] Bruno Bachimont “Modélisation linguistique et modélisation logique des ontologies : l’apport de l’ontologie formelle”, In Proceedings of IC 2001, pp 349-368 Plate-forme AFIA, Grenobl, France 25-28 June 2001.

[Ballard, 2004] Ballard R. L., “Fundamental Definitions in Knowledge Science & Engineering”, Course book for 10-week Knowledge Engineering course UC, Irvine, pages: 400, 2004.

[Bates, 1995] Bates, M. (1995) “Models of natural language understanding”, Proceedings of the National Academy of Sciences of the United States of America, Vol. 92, No. 22, pp. 9977-9982, USA, 1995.

[Belkin and Croft, 1992] Belkin N.J. and Croft W.B. “Information Filtering and Information Retrieval: Two Sides of the Same Coin?”, Published in Communications of the ACM Vol. 35 n°12, 1992.

[Berners-Lee, 2001] Berners-Lee, T., Hendler J. and Lassila O., “The Semantic Web”, Published in Scientific American Magazine, USA, 2001.

[Biggs et al., 1986] Biggs N., Lloyd E., and Wilson R., “Graph Theory”, 1736-1936. Oxford University Press, Oxford, UK, 1986.

[Boag, 2003] Boag S., Chamberlin D., Fernandez M., Florescu D., Robie J. and Simeon J., “XQuery 1.0: An XML query language”, http://www.w3.org/TR/xquery, November 2003.

[Brachman and Schmolze, 1985] Brachman R. and Schmolze J., “An overview of the KL-ONE Knowledge Representation System”, Cognitive science 9, pp171-216, USA, 1985.

[Brachman et al., 1991] Brachman R. J., McGuinness D. L., Patel-Schneider P. F., Resnick L. A., and Borgida A., “Living with Classic: When and How to Use a KL-ONE-like Language”, In J. Sowa, editor, Principles of Semantic Networks: Explorations in the representation of knowledge, pp 401-456, Morgan Kaufmann, San Mateo, CA, USA, 1991.

[Brachman, 1983] Brachman, R.J., “What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks”, In Computer published by IEEE Computer Society, Volume 16, pp 30-36, USA, 1983.

[Bratko, 2000] Bratko I.,, “Programming for artificial intelligence”, Prolog (3rd ed.), Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2000.

[Brickley and Guha, 1999] Brickley, D. and Guha, R.V., “Resource Description Framework (RDF) Schema Specification”, published in Proposed Recommendation: World Wide Web Consortium, 1999.

[Buzan, 2000] Buzan, T., “The Mind Map Book”, Published byPenguin Books, ISBN 978-0452273221, 1996.

[Christopher et al., 1999] Manning C. D. and Schutze H.,“Foundations of Statistical Natural Language Processing”, Published by MIT Press, ISBN 978-0262133609, USA, 1999.

[Clark, 1999] Clark J. and DeRose S., “XML Path Language: Version 1.0”, W3C Recommendation, 1999.

[Collins and Quillian, 1969] Allan M. Collins and M.R. Quillian (1969). "Retrieval time from semantic memory". Journal of verbal learning and verbal behavior 8 (2): 240–248. doi:10.1016/S0022-5371(69)80069-1.

[Collins and Quillian, 1970] Allan M. Collins and M. Ross Quillian, "Does category size affect categorization time?", Published in Journal of verbal learning and verbal behavior 9 (4): 432–438. doi:10.1016/S0022-5371(70)80084-6.

[Collins and Quillian, 1972] Collins A., Quillian M. R. "Experiments on Semantic Memory and Language Comprehension" in Cognition in Learning and Memory. Wiley, New York, USA, 1972.

[Cuarino et al., 1999] Cuarino N., Masolo C., and Vetere G., "Ontoseek: Content-based Access to the Web," IEEE Intelligent Systems, Volume 14, no. 3, pp. 70-80, 1999.

[DAML,2000] DAML. Drapa Agent Markup Language. http://www.daml.org/about.html, 2000.

[Davis and Buchanann, 1984] Davis R. and Buchanann B.G. "Meta-Level knowledge: Overview and applications", Published in IJCAI, ACM SIGIR, n° 5, Cambridge, United Kingdom, 1984.

[Dean et al, 2003] Dean, Mike, Schreiber, Guus, Harmelen V., Frank. Hendler, Jim, Horrocks, Ian, McGuinness, Deborah L., Patel-Schneider, Peter F., and Stein, Andrea L., "OWL Web Ontology Language Reference", http://www.w3.org/TR/owl-ref/, 2003.

[Eco, 1999] Eco U., "Kant And the Platypus", published by Secker and Warburg, United ingdom, 1999.

[Farquhar et al., 2000] Fraquhar A., Fikes R. and Rice J., "Ontolingua server: a tool for collaborative ontology construction", In International Journal for Human Computer Studies (46), pp 707-727, 2000.

[Fass and Wilks, 1983] Fass D. and Wilks Y., "Preference Semantics, III-Formedness, and Metaphor", Published in American Journal of Computational Linguistics 9(3-4): 178-187, USA, 1983.

[Fensel et al., 2000] Fensel D., Harmelen F. V., Horrocks I., McGuinness D. L., and Patel-Schneider P. F., "OIL: An Ontology Infrastructure for the Semantic Web", IEEE Intelligent Systems 16(2), pp 38-45, New Jersey, USA, 2001.

[Ferber, 1995] Ferber J., " les systems Multi-Agents: vers une intelligent collective", InterEditions, Paris, France, 1995.

[Fernandez et al., 1997] Fernandez M., Gomez-Perez A., and Juristo N., "METHODONTOLOGY: From Ontological Art to Ontological Engineering", In Workshop on Knowledge Engineering: Spring Symposium Series (AAAI'97), pp 33-40, AAAI Press, Menlow Park, CA, USA, 1997.

[Forrest, 1974] Forrest, D.W., "Francis Galton: The Life and Work of a Victorian Genius", Taplinger, ISBN 0-8008-2682-5, 1974.

[Golub and Kahan, 1965] Golub G. H. and Kahan W., “Calculating the singular values and pseudo-inverse of a matrix”, Published in Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis 2(2): 205–224, 1965.

[Gomez-Perez, 1994] Gomez-Perez A., “Some Ideas and Examples to Evaluate Ontologies”, Published In: Technical Report Technical Report KSL-94-65, Knowledge Systems Laboratory, Stanford, USA, 1994.

[Gruber, 1993] Gruber T.R., “Towards principles for the design for ontology’s used for knowledge sharing, Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, Padova, Italy, 1993.

[Gruber, 1993] Gruber T. R., "A translation approach to portable ontology specifications". In: Knowledge Acquisition 5, pp. 199-220, 1993.

[Gruninger and Fox, 1995] Gruninger M. and Fox M.S. “Methodology for the Design and Evaluation of Ontologies”, In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI, Montreal, Canada, 1995.

[Guarino et Giaretta, 1995] Guarino N. and Giaretta P., “Ontology and Knowledge Bases Towards a Terminological Clarification”, in N. Mars (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, IOS Press, pp 25-32, Amsterdam, 1995.

[Guarino et Welty, 2000] Guarino N, and Welty C. “A Formal Ontology of Properties, in Dieng R. and Corby o., eds., Knowledge Engineering and Knowledge Management: Methods , Models and Tools, International Conference EKAW’2000, Springer-Verlag, pp 97-112, 2000.

[Guarino, 1997] Guarino N. “Understanding, Building, and Using Ontologies” In: International Journal of Human-Computer Studies, SpecialIssue on Putting Ontologies to Use, Vol. 46, Issue 2-3, pp 239-310, Duluth, MN, USA, 1997.

[Guarino, 1998] Guarino N., “Formal ontology and information systems", In: N. Guarino (ed.), Formal Ontology in Information Systems”, Proceedings of the First International Conference, IOS Press, p. 4, Trento, Italy, 6-8 June 1998.

[Helder and McGuinness, 2000] Helder, J. and McGuinness, D.L., “The DARPA AgentMarkup Language”, IEEE Intelligent Systems, 2000.

[Herman, 2007] Herman I., “Semantic Web Activity Statement”, W3C, 2007.

[Horrocks et al, 2001] Horrocks I. and Patel-Schneider P. F., "Reducing OWL Entailment to Description Logic Satisfiability”, Murray Hill, New Hersey, USA, 2001.

[Jasper and Uschold, 1999] Jasper R. and Uschold M., “A Framework for Understanding and Classifying Ontology Applications”, In Twelfth Workshop on Knowledge Acquisition Modeling and Management KAW'99, Alberta, Canada, 1999.

[Jean et al, 2006] Jean S., Pierra G. and Ait-Ameur Y., "Domain ontologies : a database-oriented analysis”. In Web Information Systems and Technologies,WEBIST’2006. http://www.lisi.ensma.fr/ftp/pub/documents/papers/2006/2006-WEBIST-Jean.pdf.

[Jevons, 1870] Jevons, “Elementary Lessons in Logic”, London, 1870.

[Jolliffe, 2002] Jolliffe I.T., “Principal Component Analysis”, Published in Springer Series in Statistics, 2nd ed., Springer, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4, New York, USA, 2002.

[Jung and Jaffé, 1965] Jung C.G. and Jaffé A., “Memories, Dreams, Reflections”, Published in New York: Random House, p. 8, .New York, USA, 1965.

[Keene, 1989] Keene S. E. “Object-Oriented Programming in Common-Lisp”, Published in Addison Wesley, 1989.

[Landauer et al., 1998] Landauer T., Foltz P. W. and Laham D.,“Introduction to Latent Semantic Analysis”, 1998.

[Lassila and Swick, 1999] Lassila O. and Ralph S., “Resource Description Framework (RDF) Model and Syntax Specification”, World Wide Web Consortium, http://www.w3.org/TR/REC-rdf-syntax/, 1999.

[Mach et al., 1999] Mach M., Dzbor, M., Furdik, K., Paralic, J., “Organisational Memory - A Knowledge Modeling Approach”, Published in IIS 99, Varazdin, Croatia, 1999.

[MacQueen, 1967] MacQueen J. B., “Some Methods for classification and Analysis of Multivariate Observations”, Published in Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:pp 281-297, USA, 1967.

[Maedche and Staab, 2001] Maedche A. and Staab S. “Ontology Learning for the Semantic Web”, Volume 16 IEEE Intelligent Systems, 2001.

[Maedche and Staab, 2004] Maedche A. and Staab S., “Handbook on Ontologies, chapter Ontology Learning”, p. 173–190. Handbook in Information Systems. Springer, 2004

[Mahe and Riccio, 2001] Mahé S.A., Riccio P.M. et Vailliès S., “des elements pour un modèle: la lutte des classes!”, Revue Génie Logiciel, n°58, Paris, septembre 2001.

[Makhfi, 2003] Makhfi P., “Introduction to knowledge modeling”, http://www.makhfi.com/KCM_intro.htm, 2003.

[McCarthy, 1959] McCarthy J., “Programs with common sense.”, In Proceedings of the Teedington Conference on the Mechanization of Thought Processes, 756-91. London: Her Majesty's Stationery Office, United Kingdom, 1959.

[McCarthy, 1963] McCarthy J., “A basis for a mathematical theory of computation”, Published In Computer Programming and formal systems, North-Holland, 1963.

[McGuinness and Wright 1998] McGuinness, D.L. and Wright, J., “Conceptual Modeling for Configuration: A Description Logic-based Approach”, published in Artificial Intelligence for Engineering Design, Analysis, and Manufacturing - special issue on Configuration, pp 333-344, New York, USA, 1998.

[McGuinness et al. 2000] McGuinness D.L., Fikes,R., Rice J. and Wilder S., “An Environment for Merging and Testing Large Ontologies”, published in Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000), Morgan Kaufmann Publishers, A. G. Cohn, F. Giunchiglia and B. Selman, editors. San Francisco, CA, USA, 2000.

[Ménager, 2004] Ménager M. “Programme Toxicologie Nucléaire Environnementale : Comment fédérer et créer une communauté scientifique autour d’un enjue de société”, Intelligence Collective Partage et Redistribution des Savoirs, Nimes, France, 2004.

[Meyer, 1988] Meyer B., “Object-oriented Software Construction”, Published in Prentice-Hall, 1988.

[Michael et al., 1996] Covington M. A., Nute D. and Vellino A., “Prolog Programming in Depth”, published by Prentice Hall,ISBN 0-13-138645-X, New Jersey, USA, 1996.

[Minsky, 1975] Minsky M., “A Framework for Representing Knowledge, in Patrick Henry Winston (ed.), The Psychology of Computer Vision. McGraw-Hill, New York,U.S.A., 1975.

[Moore, 2003] Moore A., “K-means and Hierarchical Clustering - Tutorial Slides”, 2003.

[Musen M. A. 1992] Musen M. A., “Dimensions of knowledge sharing and reuse”, Comput Biomed Res., 25(5), pp 435–467, Oct 1992.

[Noy et al, 2001] Noy N. F. and McGuinness D. L., “Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory. Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, USA, March 2001.

[Noy et al, 2001] Noy N. F., Sintek M., Decker S., Crubezy M., Fergerson R. W., and Musen M. A., “Creating Semantic Web Contents with Protégé 2000”, IEEE Intelligent Systems, 16(2), pp 60-71, 2001.

[Packard, 1961] Packard V., “The Hidden Persuaders”, Published in Penguin, paperback edition, p. 129, 1961.

[Pearson, 1901] Pearson, K., “On Lines and Planes of Closest Fit to Systems of Points in Space”, Published in Philosophical Magazine 2 (6): 559–572, 1901.

[Penalva and Commandre, 2006] Penalva J. M. and Commandre M., “Typologie Du Travail Collaboratif Variations Autour Des Collectifs en Action”, RIC, Nimes, France, 2006.

[Quillian, 1968] Quillian M.R. “Semantic Memory”, Published in M Minsky, Ed, Semantic Information Processing,pp.216-270. Cambridge, Massachusetts: MIT Press, USA, 1968.

[Rao, 2008] Rao D., “ WordNet - An Introduction”, John Hopkins University, http://knol.google.com/k/delip-rao/wordnet/1tjpfaxusbh7/3#, Baltimore, USA, 2008.

[Rosch, 1978] Rosch E.,“Cognitive Representation of Semantic Categories”, Published by University of California, Berkeley, USA, 1978.

[Rothenfluh et al. 1996] Rothenfluh,T.R., Gennari J.H., Eriksson H., Puerta A.R., Tu S.W. and Musen M.A., “Reusable ontologies, knowledge-acquisition tools, and performance systems: PROTÉGÉ-II solutions to Sisyphus-2”, published in International Journal of Human-Computer Studies 44:, pp 303-332, USA, 1996.

[Rumelhart and Ortony, 1977] Rumelhart D. E. and Ortony A., “The representation of knowledge in memory”, Published in Anderson R. C., Spiro R. J. and Montague W. E. (eds.), Schooling and the Acquisition of Knowledge, Hillsdale, New Jersey, Lawrence Erlbaum Associates, USA, 1977.

[Salton and McGill, 1986] Salton G. and McGill M. J., “Introduction to Modern Information Retrieval”, McGraw Hill Book Co., pp 400, Newyork, USA, 1986.

[Shapiro and Rapaport, 1992] Shapiro S. C., Rapaport W. J., “The SNePS Family”, Published in computers and Mathematics with applications, 1992.

[Sheth and Maes, 1993] Sheth B. and Maks P., “Evolving agents for personalized information filtering”, Published In Proceedings of the IEEE CAIA-93. IEEE, New York, pp: 345-352, USA, 1993.

[Shetty et al., 2006] Shetty R. T. N., Riccio P. M. and Quinqueton J.,“Hybrid method for knowledge processing, integration and representation”, IEEE-IRI ‘06 proceedings, Hawaii, USA, 2006.

[Sowa, 1984] Sowa J. F., “Conceptual structures: information processing in mind and machine”, Proceedings of Addison-Wesley Longman Publishing Co., pp 481, Boston, MA, USA, 1984.

[Sowa, 1987] Jonh F.Sowa, "Semantic Networks", Published in Encyclopedia of Artificial Intelligence, Ed. Stuart C Shapiro.

[Sowa, 2000] Sowa J. F.,“Knowledge Representation: Logical, Philosophical, and Computational Foundations”, Brooks Cole Publishing Co., Pacific Grove, CA, USA, 2000.

[Sure et al., 1999] Sure Y., Erdmann, M., Angele J., Staab S., Studer R. and Wenke D., “OntoEdit: Collaborative ontology development for the Semantic Web”, Proceedings of the International Semantic Web Conference (ISWC), Sardinia, Italy, 2002.

[Swartout, 1996] Swartout W. R., “Future Directions in Knowledge-Based Systems”, ACM Comput. Surv, Vol. 28(4es): 13, New York, USA, 1996.

[Tulving and Donaldson, 1972] Tulving E., Donaldson W., “Episodic and semantic memory”, Published New York: Academic Press, pp 381-403, USA, 1972.

[UML, 2000] Rational Corporation: UML Notation Guide 2, 2000.

[Uschold and Gruninger, 1996] Uschold M. and Gruninger M., “Ontologies: Principles, Methods and Applications”, Knowledge Engineering Review, 11(2): pp 93-113, June 1996.

[Uschold and King, 1995] Uschold M., King M., “Towards a methodology for building ontologies”, in Prceedings of IJCAI’95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 1995.

[Uschold et al., 1996] Uschold M., King M., Moralee S., and Zorgios Y., “The Enterprise Ontology”, In: The Knowledge Engineering Review, 13(1): pp31-89, 1998.

[Van Heijst et al., 1996] Van Heijst G., Schreiber A., and Wielinga B. “Using explicit ontologies in KBS development”, published in International Journal of Human-Computer Studies, 45:xxx-yyy (to appear). 1996.

[Van Renssen, 2005] Van Renssen, A., “Gellish: A Generic Extensible Ontological Language”, Delft University Press, ISBN 90-407-2597-7, Netherlands, 2005.

[W3C, 2008] W3C Semantic Web Frequently Asked Questions. W3C, 2008.

[Winston et al., 1987] Winston M.E., Chaffin R. and Hernnann D., “A taxonomy of part – Whole Relations”, Cognitive Science 11, 1987.

[Woods and Schmolze, 1992] Woods W. A. and Schmolze J. G., “The KL-ONE Family”, Published in Computers and Mathematics with Applications, 23(5):133--177, 1992.

[WordNet, 2002] “WordNet An Electronic Lexical Databse”, MIT Press, 2.002

Publications and Awards

Keywords: Ontology, Hybrid methods, Semantic network, Knowledge representation and reasoning, Learning, Data mining, Information retrieval, Classification, Classical search problems, KL-One based knowledge modeling.

Papers

- “Collaborative Platform Using Knowledge Cartography - ToxNuc-E”, Reena T. N. Shetty, Joël Quinqueton, Pierre-Michel Riccio, Jean-Michel Penalva, Jean Villerd , (**Nominated for best paper award**) International Symposium on Collaborative Technologies and Systems (CTS), May 2007, Orlando, USA.. The paper was chosen by the jury committee for the best paper award.

- “A Dynamic Knowledge Representation Model based on Hybrid Approach”, Reena T. N. Shetty, Pierre-Michel Riccio, Joël Quinqueton, “Review” International Journal for Digital Content Technology and its Application (JDCTA), June 2007, South Korea.

- “Hybrid Knowledge Model for Relevant Information Retrieval”, Reena T. N. Shetty, Pierre-Michel Riccio, Joël Quinqueton, Workshop « Knowledge and Reasoning for Answering Questions», International Joint Conference on Artificial Intelligence (IJCAI), January 2007, Hyderabad, India.

- “Hybrid Model for Knowledge Representation”, Reena T. N. Shetty, Pierre-Michel Riccio, Joël Quinqueton, IEEE - International Conference on Hybrid Information Technology, November 2006, Jeju, South Korea.

- “Extended Semantic Network for Knowledge Representation – An Hybrid Approach”, Reena T. N. Shetty, Pierre-Michel Riccio, Joël Quinqueton, International Conference on Intelligent Information Processing, proceedings publication by Springer, September 2006, Adelaide, Australia.

- “Hybrid Method for Knowledge Processing, Integration and Representation”, Reena T. N. Shetty, Pierre-Michel Riccio, Joël Quinqueton, IEEE-International conference on Intelligent Reuse of Information, September 2006, Hawaii, USA.

- “Extended Semantic Network for Knowledge Sharing”, Reena T. N. Shetty, Pierre-Michel Riccio, Joël Quinqueton, National conference on Emerging Trends In Computer Science and Engineering, January 2006, Tiruchengode, India.

Posters

- “ Classification et recherche d’information pour la Toxicologie Nucléaire Environnementale”, Reena T. N. Shetty, Pierre-Michel Riccio, Joël Quinqueton, Seminar ToxNuc-E -- CEA, December 2005, Paris, France. Won the best poster award presented by CEA (Commissariat à l’Energie Atomique).

- “Réseaux sémantiques étendus pour la gestion d’informations et de connaissances dans le domaine de la Toxicologie Nucléaire Environnementale”, Reena T. N. Shetty, Pierre-Michel Riccio, Joël Quinqueton, Rencontre Intelligence Collective, May 2006, Nîmes, France. Won the best poster award presented by AFIA (Association Française d’Intelligence Artificielle).

- “Classification de documents scientifiques basée sur les réseaux sémantiques étendus”, Reena T. N. Shetty, Pierre-Michel Riccio, Joël Quinqueton, Seminar ToxNuc-E -- CEA, December 2006, Paris, France.