PhD Dissertation

International Doctorate School in Information and Communication Technologies

DIT - University of Trento

Aligning Controlled vocabularies for enabling semantic matching in a distributed knowledge management system

12th April 2010

Advisor:

Prof.Fausto Giunchiglia

Università degli Studi di Trento

April 2010

Abstract

The underlying idea of the Semantic Web is that web content should be expressed not only in natural language but also in a language that can be unambiguously understood, interpreted and used by software agents, thus permitting them to find, share and integrate information more easily. The central notion of the Semantic Web's syntax are ontologies, shared vocabularies providing taxonomies of concepts, objects and relationships between them, which describe particular domains of knowledge. A vocabulary stores words, synonyms, word sense definitions (i.e. glosses), relations between word senses and concepts; such a vocabulary is generally referred to as the Controlled Vocabulary (CV) if choice or selection of terms are done by domain specialists. A facet is a distinct and dimensional feature of a concept or a term that allows a taxonomy, ontology or CV to be viewed or ordered in multiple ways, rather than in a single way. The facet is clearly defined, mutually exclusive, and composed of collectively exhaustive properties or characteristics of a domain. For example, a collection of rice might be represented using a name facet, place facet etc. This thesis presents a methodology for producing mappings between Controlled Vocabularies, based on a technique called "Hidden Semantic Matching". The "Hidden" word stands for it not relying on any sort of externally provided background knowledge. The sole exploited knowledge comes from the "semantic context" of the same CVs which are being matched. We build a facet for each concept of these CVs, considering more general concepts (broader terms), *less general concepts (narrow terms) or related concepts (related terms). Together these form a concept facet (CF) which is then used to boost the matching process*

Keywords
[Controlled Vocabulary, Thesauri, Ontology, Facet, AGROVOC, CABI]

Declaration

I hereby declare that I have carried out the entire work in this thesis on my own and that, to the best of my knowledge and belief, it contains no material previously published by another person or institution. I also declare that I have not submitted this thesis to any other university.

Ahsan-ul Morshed
April, 2010

Acknowledgements

I would like to sincerely thank my PhD advisor Professor Fausto Giunchiglia for his continuous guidance and support during my PhD. Through his meticulous training, I learned what research is all about and how to carry it out to perfection. With him, I also learned how to manage my research activities and to carry them out responsibly and seriously. I acknowledge that working with him has provided me with the ideal research training, which will assist me immensely in my future research endeavors.

I would also like to thank Johannes Keizer, Gudrun Johanssen and Margherita Sini for their support and encouragement they provided during the mapping project at FAO.

I would also like to thanks Shaun Hobbs from CABI for providing access to CABI materials.

I would also like to thank Adriano Venturini of eCTRL Solutions for believing in me and giving me an opportunity to work with his company during my PhD Studies.

A strong round of applause goes to my family for all the support and encouragement they provided during my PhD. My wife, Nargish Sultana provided the essential framework of moral support and encouragement that considerably facilitated my PhD work and my thesis-writing task. My sister, Sabiha Afroz rimi always provided the required motivating words to propel me onwards, and to eventually complete my thesis. My father, Azizur Rahman, and my mother, Anjuara Begum, deserve a very special acknowledgement. They have always believed in me and have done everything possible to ensure that I succeed. It would have been really difficult for me to have gotten to where I am now without them. I dedicate this thesis to them.

List of Publications

- A. Morshed. Controlled Vocabulary Matching in Distributed Systems, at BNCOD 2009 Conference,UK.
- A. Morshed and M. Sini. Aligning Controlled vocabularies: Algorithm and Architecture at Workshop on Advance Technologies for Digital Libraries 2009, AT4DL, Trento, Italy.
- A. Morshed, M. Sini and J. Keizer. Aligning Controlled Vocabularies using a Facet based approach (Technical report)
- M. Sini, J. Keizer, G. Johannsen, A. Morshed, S. Rajbhandari and M. Amirhosseini. The AGROVOC Concept Server Workbench System: Empowering management of agricultural vocabularies with semantics at International Association of Agricultural Information Specialists (IAALD), France, 2010.
- M. Zeng, A. Morshed, G. Johanssen, and J. Keizer. Bridging End Users Terms and AGROVOC Concept Server Vocabularies. International Conference on Dublin Core and Metadata Applications (DC-2010), Pittsburgh, USA, 2010 (to be submitted).
- A. Morshed. Towards the Automatic Classification of Documents in User-generated Classifications (Technical paper).
- A. Morshed and R. Singh. Evaluation and Ranking of Ontology Construction Tools (Technical Paper).

Contents

List of Tables

List of Figures

0.1 The Context

The underlying idea of the Semantic Web is that web content should be expressed not only in natural language but also in a language that can be unambiguously understood, interpreted and used by software agents, thus permitting them to find, share and integrate information more easily. The central notion of the Semantic Web's syntax [93, 27, 123], are ontologies, shared vocabularies providing taxonomies of concepts, objects and relationships between them, which describe particular domains of knowledge. These taxonomies contain the idea of a particular domain of knowledge; the domain contains a set of vocabularies that consist of a set of words and phrases used to describe concepts. A vocabulary stores words, synonyms, word sense definitions (i.e. glosses), relations between word senses and concepts [63]; such a vocabulary is generally referred to as the Controlled Vocabulary (CV) if choice or selection of terms are done by domain specialists. They help to organize the knowledge for subsequent retrieval. The importance of the controlled vocabulary can hardly be underestimated; generally, each company or research group has its own information source, such as databases, schemas, and structures. Each of these sources has its respective set of individual CVs, creating a high level of heterogeneity.

This thesis is devoted to CV matching as a solution to handle this huge heterogeneity problem faced by computer systems. The objective of CV matching is finding correspondences between semantically related concepts of different vocabularies. These correspondences may include not only equivalence but also subsumption or disjointness relations between concepts.

However, different matching solutions have been introduced [93, 71, 114, 128] with different results, including databases, information systems, and artificial intelligence. They take advantage of various properties of thesauri and ontologies, e.g., label, structure, or data instances and use techniques from different fields, e.g., linguistic, automated reasoning, statistic and data analysis and machine learning. These solutions share some techniques and tackle similar problems, but differ in the ways they combine and exploit their outcomes.

0.2 The Problem

To demonstrate the problem, let us consider a small example of the United Nations Food and Agriculture Organization AGROVOC [117] and Commonwealth Agricultural Bureaux International (CABI) thesauri [118] which are our respective CVs that are shown in Figure 1 and represent a possible real-world situation.

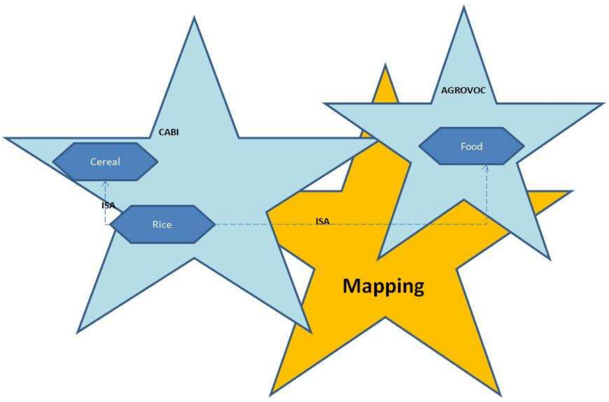


Figure 1: CABI AND AGROVOC Thesaurus

Imagine that one thesaurus organization wants to browse the information in another thesaurus. To execute the operation, we have to integrate two databases from two thesauri. Both of them model their information through RDF schema. Now, we have to identify concepts from the source vocabulary and match them to concepts in the target vocabulary. In this example, CABI is our source vocabulary and AGROVOC is our target vocabulary. In CABI, “rice” is a kind of “cereal” but in AGROVOC, “rice” is a kind of “food”. To correctly integrate the two thesauri, cereal should be subsumed under food.

0.3 The Solution

Different methodologies for CV matching have been introduced so far. This work concentrates not only on string matching but also on matching between the position of trees which is called structure matching. It proposes the so-called concept facet approach. This approach is based on a concept and its surrounding relationships. The principal idea is to take a concept and its narrower and broader terms. This kind of structure of a concept is a so-called concept facet. Next, the correspondence between them is calculated by computing the logical relations (e.g., equivalence, subsumption, disjointness). Since background knowledge will not be used, the matching task must be done exclusively with concept labels.

0.4 Innovative Aspects

This thesis makes contributions in the following areas:

- Its results will help to find the right information within the Agriculture domain in a large.
- It works as an online resource so that anyone can download and use it as a research tool.
- It presents a development prototype of the CABI Abstract search System.

0.5 Structure of the Thesis

The thesis is organized in the following manner: In Chapter 1 present background information. In Chapter 1 we discuss the state-of-the-art, and in sections, we discuss semantic matching in lightweight ontologies, the role of controlled vocabularies in semantic matching, different matching techniques and matching systems. In Chapter 2, we discuss different Knowledge Organization Systems (KOS's), factors for the problems and the main problem with which we are going to deal. In Chapter 3, we discuss facet and a facet based controlled vocabulary matching. In Chapter 4, we illustrate a system architecture for our mapping system. In Chapter 5, we discuss evaluation methodologies, evaluation and results. Finally, we discuss future directions.

Chapter 1

State of the Art

1.1 Matching Techniques

In order to solve the matching problem, there are several matching techniques available. These techniques can be classified into two categories. The first is automatic matching. The second is manual matching. The previous and recent techniques are described in [114, 93]. Most of the techniques are defined at [35, 123]. We also discuss automatic and manual techniques below:

1.1.1 Automatic Matching techniques

Element Matching techniques

To start the process of element level matching techniques [49, 18], we need to clean noisy content from the strings. In order to do this, we need to go through the following steps.

- *Normalization.* It is necessary to reduce the strings to be compared with a common format.

- *Diacritical suppression.* Replacing characters with diacritical signs with their most frequent replacement.

- *Link Stripping.* Normalizing some links between words, such as replacing apostrophes and underscores into dashes or blanks.
- *Digit suppression.* Check if there are any digits in the given strings and remove these from the strings.
- *Punctuation elimination.* Eliminating punctuation signs.
- *Transliteration.* Replacing missing or wrong alphabets with right alphabets.

After normalizing a string, we can compare it using the following techniques:

String comparison:

- *String equality.* If two strings are identically equal, we returns 1, otherwise we return 0. For instance, "Agriculture" and "Agriculture" return 1.
- *Substring test.* We consider two strings as similar when one is a substring of another. For instance, "net" is a substring of "network".
- *Levenshtein distance.* We take two strings and compute the distance between them with minimal cost of operation (i.e. insertion, deletion, and substitution of characters) to transform one string into another, normalized by the length of longest string [74]. For instance "Agriculture Industry" and "Agriculture manufacture" return 0.347

Corpus-based Techniques

Corpus-based techniques exploit the information contained in large collection of documents called corpora. They are mainly used as an alternative to string-based techniques [101, 102, 115]. The key advantage of corpus-based representation is that it avoids the need for careful logical design of a single comprehensive ontology. Corpus-based techniques are mainly token-based or extension-based.

Token-based distances:

We consider strings as a set of words in which a particular item can appear several times. It may be adapted to ontology entities by splitting strings into independent tokens.

- *Cluster Code Difference (CCDiff).* A score function can be defined as the difference of the summed length of the coded string tokens that are members of the cluster, and the length of the cluster when it is updated with the candidate string.

- *Latent Semantic Indexing (LSI).* Some techniques are used to reduce spaces, like those obtained by correspondence analysis, in order to deal with a dimension as well as to automatically map words of similar meanings to the same dimension. An example of this technique uses singular value decomposition and is known as latent semantic indexing.

- *Term frequency-Inverse document frequency.* It is used for scoring the relevance of a document, i.e., a bag of words, to a term by taking into account the frequency of appearance of the term in the corpus. It is usually not a measure of similarity, but rather of the relevance of a term to a document.

Extension-based techniques:

We consider classes and their instances for comparing the ontologies. The following techniques talk about the classes and instances comparisons.

- *Common extension comparison.* The simplest way to compare ontology classes when they share instances is to test the intersection of their instance sets A and B and to consider these classes very similar when $A \cap B$=A=B, or more generally when $A \cap B$=B or $A \cap B$=A.

- *Formal Concept Analysis.* One of the tools of formal concept analysis (FCA) [127] is the computation of a concept lattice. The idea behind FCA is the duality of a set of objects and their properties: the more their properties are constrained, the fewer the objects that satisfy the constraints. So a set of objects with properties can be organized in a lattice of concepts covering these objects. Each object can be identified by its properties (the intent) and covers the individual satisfying these properties (the extent).

- *Instance identification techniques.* If a common set of instances does not exist, it is possible to try and identify which instance from one set corresponds to which other instance from the other set. This method is usable when one knows that the instances are the same. A first natural technique for identifying instances is to take advantage of keys in databases. Keys can be either internal or external . When keys are not available, or they are different, other approaches to determine property correspondence use instance data to compare property values. In databases, this technique is known as record linkage.

- *Disjoint extension comparison.* When it is not possible to directly infer a data set common to both ontologies, it is easier to use approximate techniques for comparing class extensions. These methods can be based on statistical measurement of the features of class members, on the similarities computed between instances of classes or on a matching between entity sets.

- *Statistical approach.* Instance data can be used to compute some statistics about the property values found in instances, such as maximum, minimum, mean, variance, existence of null values, existence of decimals, scale, precision, grouping, and number of segments. This allows the characterizing of the domains of class properties from the data.

- *Similarity-based extension comparison.* Similarity-based techniques do not require the classes to share the same set of instances. The methods based on common extensions always return 0 when the two classes do not share any instances, disregarding the distance between the elements of the sets. In some cases, it is preferable to compare the sets of instances. This requires a (dis) similarity measure between the instances that can be obtained with the other basic methods.

- *Matching-based comparison.* Matching-based comparisons consider that the elements to be compared are those which correspond to each other, i.e. the most similar ones. To that extent, the distance between two sets is considered a value to be minimized and its computation is an optimization problem: that of finding the elements of both sets which correspond to each other. In particular, it corresponds to solving a bipartite graph matching problem.

Meaning-based techniques

Path comparison. We can compare not only the labels of objects but the sequence of labels of entities to which those bearing the label are related.

Knowledge-based techniques

These techniques are mainly used as background knowledge [36, 104]. The following sources are involved in these techniques.

External sources:

- *Mono Lexicons.* A set of words together with a natural language definition of these words in a single language. For instance, dictionary, WordNet [89], etc.

- *Multi-lingual lexicons.* The lexicons in which the definition is replaced by the equivalent terms in another language. For instance, EuroWordNet [122].

- *Semantico-syntactic lexicons.* A set of documents or resources is used in natural language analyzers. They very often not only record names but also their categories. They are difficult to create and are not greatly used in ontology matching [102].

- *Thesauri.* A kind of lexicon to which some relational information has been added. The following relationship exist in thesauri: Broader terms (BT), Narrow Terms (NT), Related Term (RT), and Used For (UF) [89].

- *Terminologies.* Thesauri for terms, which very often contain phrases rather than single words. Usually domain specific and tend to be less equivocal than dictionaries.

- *Alignment reuse.* These techniques represent an alternative way of exploiting external sources, which record alignments of previously matched ontologies. Alignment reuse is motivated by the intuition that many ontologies to be matched are similar to already matched ontologies, especially if they are describing the same application domain.

- *Upper level and domain specific formal ontologies.* For instance, SUMO [91] ontology. The key characteristic of these ontologies is that they are logic-based systems [16], and therefore matching techniques exploiting them are based on semantics. For the moment, we are not aware of any matching system which uses this kind of technique.

Structure-level techniques:

- *Iterative structure matching method.* This method uses the proposed matching pairs from all the previous methods in order to compute mappings based on a concept's enhanced vicinity, which includes all the concepts related to it.

- *Constraint-based techniques / Internal Structure Methods.* These methods are based on the internal structure of entities and use such criteria as the set of their properties, the range of their properties (attributes and relations), their cardinality or multiplicity, and the transitivity or symmetry of their properties to calculate the similarity between them. This can be applied to a set of classes and a set of relations. It means that if we have a set of relations r1. .rn in the first ontology which are similar to another set of relations r'1...r'n in the second ontology, it is possible that two classes, which are domains of relations in those two sets, are similar too.

- *Property comparison and keys.* Property comparisons involve comparing the property datatype. Keys are mostly used as a means to identify individuals and they apply the methods on common set of instances. Keys can also be used for identifying classes: two classes identified in the same way are likely to represent the same set of objects.

- *Datatype comparison.* It is possible to determine how close a datatype is to another (ideally this is based on the interpretation of datatypes as sets of values and the set-theoretic comparison of these datatypes).

- *Domain comparison.* Depending on the entities being considered, what is inferred from a property may be different: in classes these are domains while in individuals these are values. Moreover, they can be structured as sets or sequences. It is thus important to consider this fact in the comparison.

- *Graph-based techniques.* An ontology can be considered a graph whose edges are labeled by relation names. Finding the correspondences between elements of such graphs corresponds to solving a form of the graph homomorphism problem: namely, it can be related to finding a maximum common directed subgraph.

- *Taxonomy-based techniques.* There have been several measures proposed for comparing classes based on taxonomic structure [5]. The most common ones are based on counting the number of edges in the taxonomy between two classes. The structural topological dissimilarity of a hierarchy follows the graph distance, i.e., the shortest path distance in a graph, taken here as the transitive reduction of the hierarchy.

- *Super-or-subclass rules.* These matchers are based on rules that classes are similar if their super-or subclasses are similar.

- *Bounded path matching.* Two paths with links between the classes are defined by hierarchical relations, comparing terms and their positions along these paths, and identifying similar terms.

- *Mereological structure.* This structure corresponds to the part-of relationship. If it is possible to detect the relations that support the part-of structure, this can be used for computing similarity between classes: they will be more similar if they share similar parts.

- *Data analysis and statistical techniques.* These techniques take advantage of a representative sample of a population in order to find regularities and discrepancies. This helps in grouping together items or computing distances between them.

- *Repository of structures.* Repositories of structures store ontologies and their fragments together with pair-wise similarity measures, e.g. coefficients in the range [0..1]. Unlike alignment reuse, repositories of structures store only similarities between ontologies, not alignments.

Semantic-based techniques:

- *Techniques based on external ontologies.* When two ontologies are matched, they often fail to manage common ground for comparisons. Formal ontologies define the common context or background knowledge. Most of time, lexicons are used for background knowledge [36, 88, 101].
- *Anchoring / contextualizing* . We can consider ontologies O1 and O2 to the background ontology O.

1.1.2 Manual mapping techniques

In this technique a domain expert acts as a mapper to map the ontologies or vocabularies by hand or with the help of a computer. The expert performs analysis and translation, but this often takes too much time and can be very costly. To minimize the time, the expert constructs rules [20] and, by using these rules two vocabularies are mapped.

1.2 Matching Systems

The following are the most popular systems for matching of ontologies.

1.2.1 ONION

ONION (Ontology compositION system) provides an approach [103, 114] for resolving heterogeneity between different ontologies. Its basic assumption is that merging whole ontologies is too costly and inefficient. Therefore, it focuses on creating so called articulation rules, which link corresponding concepts. As manual creation of these rules is not very efficient either, it uses a semi-automatic approach, which takes into account heuristics on several simple relations, such as labels, subsumption hierarchies and attribute values. Dictionary information is also used for the alignment process. From these relations a match is presented to the user who then has to decide whether the alignment is valid or not. In the articulation rules, linking can be applied when an application requires information from two ontologies.

1.2.2 RiMOM

RiMOM [66] is a multiple strategy ontology alignment framework based on risk minimization of Bayesian decision theory. The RiMOM automatically determines which ontology alignment methods to use, what kind of information to use in the similarity calculation and how to combine multiple methods as necessary. This tool includes edit distance and vector distance strategies for ontology matching purposes; it also takes OWL or Resource Description framework (RDF) as input files.

1.2.3 Smart, Prompt, Anchor-PROMPT, PromptDiff, and Chimera

Anchor-Prompt [92, 114] is an ontology merging and alignment tool with a sophisticated prompt mechanism for possible matching terms. It is an extension of Prompt (formerly known as SMART). It handles ontologies expressed in such knowledge representation formats as OWL and RDF

Schema. Anchor-Prompt is a sequential matching algorithm that takes as input two ontologies, internally represented as graphs and a set of anchors-pairs of related terms, which are identified with the help of string-based techniques, such as edit-distance or defined by a user or another matcher computing linguistic similarity. The algorithm then refines them by analyzing the paths of the input ontologies limited by the anchors, in order to determine terms frequently appearing in similar positions on similar paths. Finally, based on the frequencies and user feedback, the algorithm determines matching candidates. Chimera is an interactive tool for ontology merging. Its basic ontology content can be accessed through OKBC is meta language but it can be ontology protocol [119]. After executing a linguistic matcher, Chimera uses the results to perform the merging operation. During this process, a human user must to decide whether to merge or not. Chimera also provides proposals for reorganizing the taxonomy when a merge has been processed. Overall, Chimera allows diagnosing and manual editing for ontology merging. The actual alignment of entities, however, is based on simple measures.

1.2.4 Cupid

Cupid [60, 114, 35] is an automatic ontology matching system based on element and structure level matching. In terms of inputting data, it is very generic and has been applied to XML and different relational data models. The algorithm comprises three steps. In the first step, elements (nodes of the schema) are compared by linguistic means, including external information about synonyms. In the second step, for the structural matching, the data model is transformed into a tree. Pairs are then compared by examining their leaf sets. A similarity is calculated through a weighted mean of linguistic and structural similarity. In the third step, a threshold is applied to finally decide on an alignment or not.

1.2.5 Similarity Flooding, Rondo

The Similarity Flooding (SF) is an algorithm [114, 113, 111] for automatic ontology matching based on the idea of similarity propagation in Schemas. Schemas are presented as directed labeled graphs, grounding on the OIM specification. The algorithm manipulates them in an iterative fixed-point computation to produce an alignment between the nodes of the input graphs. The technique starts from string-based comparison, such as common prefix suffix tests, of the vertices labels, to obtain an initial alignment which is refined within the fixed-point computation. The basic concept behind the similarity flooding algorithm is the similarity spreading from similar nodes to adjacent neighbors through propagation coefficients. From iteration to iteration the spreading depth and a similarity measure increase till the fixed-point is reached. The result of this step is a refined alignment which is further filtered to finalize the matching process.

1.2.6 COMA (COmbination of Matching algorithms)

COMA is an automatic ontology [22, 35] matching tool based on the composition of several matchers. It provides a nice users interface that user can use easily to upload their ontologies and obtain results. These results can be evaluated with human edited matching results (also called golden standard). COMA contains six elementary matchers. Most of them implement string-based techniques, such as affix, n-gram, edit distance; others share techniques with Cupid (thesauri look-up, etc.). Schemas are internally encoded as directed acyclic graphs, where elements are the paths. This aims at capturing contexts in which the elements occur. Distinct features of the COMA tool compared to Cupid are a more flexible architecture and the possibility of performing iterations in the matching process. It presumes interaction with users who approve obtained matches and mismatches to gradually refine and improve the accuracy of a match. COMA++ is built on top of COMA by elaborating in more detail the alignment reuses operations and provides a more efficient implementation of the COMA algorithms and a graphical user interface.

1.2.7 CTXMatch, S-match

Context Match (CTXmatch) and Semantic Matcher (S-match) [114, 35, 13, 99] is developed by the University of Trento. CtxMatch presents an approach to derive semantic relations between classes of two classification schemas, which are extracted from databases or ontologies. Based on the labels the system identifies equivalent entities. For this, it also makes use of synonyms defined in WordNet. Other element level matchers are also included. Through an SAT-solver the system identifies additional relations between the two schemas. The SAT-solver takes the structure of the schemas into account, especially the taxonomy and its inferred implications, e.g., the fact that any object in a class is also an element of all the superclasses there of. As a result, the system returns equivalence, subsumption, or mismatch between two classes. A recent version S-Match also provides explanations of the alignments.

1.2.8 SemInt

SEMantic INTegrator (SemInt) is an automatic ontology matching tool [114, 76, 75] based on mapping between individual attributes of two schemas. Unlike most other approaches, it does not provide name-based or graph-based matching. It bases its analysis on the information available from the schema of a relational database management system and the instance data. Value distributions and averages are consequently converted into signatures. For these signatures, SemInt applies two similarity operators.

It uses either Euclidian distance or a trained neural network to determine the match candidates. The authors express that both approaches have advantages and disadvantages, which differ according to the application. The neural network further faces some efficiency problems. However, SemInt was one of the first approaches not opting for a hard-wired combination of individual rule-based similarities, but using a machine-learning based approach.

1.2.9 DIKE

DIKE is a platform [80, 114] to automatically determine synonym and inclusion (is-a, hypernym) relationships. The DIKE takes an entry-relationship schema as input. This platform calculates different similarity values between two objects based on their related objects such as attributes. These may also only be related indirectly through relation paths. The more distant related object are, the less important they are for determining the similarity. The goal is to find similar, but not necessarily identical objects. It also identifies other kinds of relations. A relation holds if the similarity value is above a fixed threshold.

1.2.10 ARTEMIS

ARTEMIS (Analysis of Requirements:Tool Environment for Multiple Information Systems) is a platform [67], of the MIMOS [10] heterogenous database mediator. The ARTEMIS is based on different similarities (which the authors refer to as affinity), such as name similarity (Using WordNet [89]), datatype similarity, and structure similarity, of the involved entities. These similarities are then summed with appropriate weights. Based on the overall similarity and a hierarchical clustering technique ARTEMIS categorizes classes into groups where each group presents a more general class with a set of global attributes. Through a mapping table, the original source schemas are linked to the virtual global schema.

1.2.11 KAON

Kaon [98] is an open-source ontology management infrastructure tailored for business applications. A modeling language based on RDFS has been developed to provide a unified environment for ontology creation, evolution and reuse. Kaon is not specifically designed for a peer-to-peer environment. The authors show how different nodes can interact to search and reuse different ontologies.

1.2.12 FALCON-AO

Falcon [58, 125], is a platform for Semantic Web applications that provides fundamental technology for finding, aligning and learning ontologies. Falcon-AO, is an automatic ontology matching system that aids interoperability between ontologies. The Falcon-AO tool takes RDF /OWL as input and produces RDF as output. Furthermore, this tool includes LMO (linguistic matching for ontologies), GMO (graph matching for ontologies) and PBM (a partition-based matcher for large ontologies).

1.3 Projects about Matching Initiatives

There are some existing matching projects in the vocabularies and ontologies matching fields. In below, we describe some of them.

1.3.1 HILT (High Level Thesaurus Project)

HILT [109, 90, 83] is a JISC-funded (Joint Information Systems Committee), UK-based, collaborative project with the overall aim of creating a

JISC shared service to facilitate the cross-searching of distributed information services by subject in a multi-scheme environment, ideally by identifying a generic approach that allows a service to be built up through distributed collaborative action. Primarily focused on an inter-scheme mapping based approach to provide a subject interoperability service, the project has recently adopted a new distributed model that will allow it to encompass other approaches to provide interoperability services. The project has recently begun its fourth phase. HILT Phase III built an M2M pilot interoperability service that:

- Offers web services access via the (SOAP-based[1]) SRW (Search and Retrieve Web Service) protocol, but designed so that an extension to other protocols (Z39.50 or SRU (Search-Retrieve by URL), for example) is an option at a later date.

- Uses SKOS Core as the "mark-up" for sending out terminology sets and classification data but allows other formats such as MARC[2] and Zthes[3] to be added later as alternatives.

- Provides the pilot datasets (DDC (Dewey Decimal Classification)[4], LCSH (Library of Congress Subject Headings)[5], IPSV (Integrated Public Sector Vocabulary)[6], AAT (Art and Architecture Thesaurus)[7], etc.), mappings (between DDC spine and other schemes), and functionality capable of servicing the five use cases agreed on in the HILT M2M Feasibility Study.

- Bases the pilot on the centralized approach to the provision of mapping

[1] Simple Object Access Protocol

[2] http://www.loc.gov/marc/

[3] http://zthes.z3950.org/

[4] http://www.oclc.org/dewey/

[5] http://www.loc.gov/

[6] http://www.esd.org.uk/standards/ipsv/

[7] http://www.getty.edu/

services piloted in HILT Phase II, but leaves open the possibility of moving towards a more distributed model.

HILT bases its matching techniques on the reuse of external ontologies. Its reports do not give statistics for the number of terms matched. The document does not indicate how many terms were selected from each ontology for the experiment.

1.3.2 CAT to AGROVOC

A mapping project [77, 85] from CAT (Chinese Agriculture Thesaurus) to AGROVOC [117] was carried out by the food and agriculture organization of UN (FAO). As part of the process, the team (Dr.Chang Chun and his students) created a new OWL document, imported the whole CAT and AGROVOC ontologies and then saved the document. Afterwards, they inserted the whole middle part of the mapping project into the upper document. The results were a whole mapping OWL document which works with entire CAT and AGROVOC thesauri A presentation was given by the team from the Agricultural Information Institute, Chinese Academy of Agricultural Sciences (AII/CAAS). The CAT source ontology contains 64,638 Chinese terms and 51,614 descriptors; 13,024 non-descriptors; 2,332 top terms organized into 40 categories (e.g. crops, etc.). In AGROVOC, the number of descriptors was not specified. They considered equivalent relationships, broader term (BT) relationships and narrow term (NT) relationships. They used the protege tool for matching purpose. They found 13,105 exact matches, 11,408 broader term matches, 173 narrow term matches and 1,747 othermatches. They used taxonomy-based matching techniques. Most of the work was done manually. For example, “cereal crops” exactMatches “cereal crops”, “universal education” broadMatches “education”, “island” narrowMatches “atolls”.

1.3.3 OAEI 2007 (Ontology Alignment Evaluation Initiative) - Food Track

The idea of this evaluation initiative was to find matches between the AGROVOC and the NALT [78] thesauri of Agriculture domain. OAEI [125, 85, 121] used taxonomy-based and linguistic-based matching techniques. These matching techniques are used in FALCON-AO, RiMOM, DSSim tools. The results are presented in Table 1.1.

System	Alignments	Alignment Type
Falcon-AO	15,300	exactMatch
RiMOM	18,420	exactMatch
X-SOM	6,583	exactMatch
DSSim	14,962	exactMatch

Table 1.1: Evaluation Results

1.4 Matching in distributed System

A peer-to-peer (P2P) distributed network [1, 68, 69, 131] is a wider communication model in which participants make a portion of their resources (such as disk storage, files and network bandwidth) available directly to their peers without intermediary network hosts or servers. P2P networks were popularized by file sharing , e.g., of pictures, music, videos, books. Former file sharing systems include Napster[8], Kazaa[9],and BitTorrent[10]. These applications describe file contents by a simple schema (set of attributes, such as title of song, author, etc.) to which all the peers in the network have to subscribe.Therefore, in the above mentioned systems the semantic heterogeneity problem (at the schema level) does not exist.

[8] www.napster.com

[9] www.kazaa.com

[10] http://www.bittorrent.com/

The use of a single system schema violates the total autonomy of peers. Although industry-strength P2P system allows peers to connect to and disconnect from the network at any time, thereby respecting some forms of peer autonomy, such as participating autonomy, they still restrict the design autonomy of peer, in matters such as how to describe the data and what constraints to use on the data.

If peers are meant to be totally autonomous, they may use different terminologies and metadata models in order to represent their data, even if they refer to the same domain of interest. Thus, in order to establish (meaningful) information exchange between peers, one of the steps is to identify and characterize relationships between their ontologies. This is a matching operation [87]. Having identified the relationships between their ontologies, these can be used for the purpose of query answering, e.g., using techniques applied in data integration systems [126].

There are some projects which use lightweight ontologies for matching purposes [14, 55, 2].

- Edutella. Edutella[11] is an open source project that creates an infrastructure for sharing metadata in RDF format. It applies the peer-to-peer model using the JXTA protocol [1]. The network is segmented into thematic clusters. In each cluster, a mediator semantically integrates source metadata. Edutella is an example of a hybrid peer-to-peer architecture, in that each source sends queries to the mediator of its own cluster, and the mediator returns a list of nodes eligible to offer semantically related information. The mediator handles a request either directly or indirectly: directly, by answering queries using its own integrated schema; indirectly, by querying other cluster mediators.

- Swap. The Swap[12] project aims [131] at overcoming the lack of se-

[11] http://www.edutella.org/

[12] http://swap.semanticweb.org/

mantics in current Peer-to-Peer systems. To this purpose, an RDF (S) metadata model for encoding semantic information is introduced, allowing peers to handle heterogeneous and even contradictory views on the domain of interest. Each peer implements an ontology extraction method to extract from its different information sources an RDF (S) description (ontology) compatible with the SWAP metadata model. Such ontologies are used by the SeRQL (Second Generation RDF Query Language) Query Language to perform query processing: peers storing knowledge semantically related to a target concept are localized through SeRQL views defined on specific similarity measures. Views from external peers are integrated through an ontology merging method to extend the knowledge of the receiving peer according to a rating model.

1.5 Semantic Matching in Lightweight ontologies

Information decoration or classified information is mostly used on the web. For instance, the most popular web portals like Google [29], Yahoo [33], DMoz (open directory project) [24], etc. classify information for user consumption. Behind this information decoration are ontologies. The use of ontologies started in ancient age time and has continued to the present. It has spread from the field of philosophy to computer science, medical science, and biology field. There are varieties of different ontologies that range from glossaries to taxonomies or database schema or a full-fledge logic theory that consists of concepts, relationships, constraints, axioms and inference machineries [57, 54]. [57] illustrates a variety of ontologies forming a continuum from lightweight, rather informal knowledge structures, to heavyweight, and formal ontologies.

Formal ontologies and lightweight ontologies are often used differently and have different strengths and weaknesses. Lightweight ontologies are directed graphs with "is-a" type relationship among the concepts hierarchy [51]. Lightweight ontologies are relatively easy to construct but are difficult to use due to their natural language labels that have ambiguous meaning. To use a lightweight ontology for matching purposes, all entities need to agree on the exact meanings of the concepts. Reaching such agreements can be difficult. By contrast, formal ontologies are very difficult to create but easy to use. In [52, 38], Fausto et. al, introduced automatically created full-fledged lightweight ontologies that are used for matching purpose. This chapter focuses on the lightweight ontologies. Match acts as an operator that takes two graph-like structures, e.g. user-classification or business catalogs that refer to lightweight ontologies and produces mappings between the nodes that correspond semantically to each other [47]. A great amount of work has been done on matching systems and techniques so far [123]. We focus on the schema matching approach proposed in [47]. This approach mainly focuses on two key themes. The first theme is that mapping ontological entities is better done by computing logical relation (e.g.equivalence, subsumption), instead of string matching. The second theme is that relations are determined by analyzing the meaning which is codified in the entities and the structure of ontologies. In particular, node labels written in natural language are translated into propositional unsatisfiability problem, which can then be efficiently solved using state-of-the art positional satisfiability. In our case, we follow the pioneer work of the semantic matching [47, 46]. Furthermore, we have adopted a semantic matching algorithm that was introduced in [47] and we analyzed this algorithm with two lightweight ontologies.

1.5.1 Ontology

As a consequence, the definition of ontology has changed considerably. As ontology is a key term in this chapter, we will further refine it here.

Ontology Definition

The word 'ontology" originates in philosophy where it is defined as the theory of " *the nature of being or the kinds of existences*". The notion of ontology was first introduced by the Greek philosophers Socrates and Aristotle. Socrates proposed abstract ideas, a hierarchy and class-instance relations. Aristotle subsequently added logical formulas. As a result a well-structured model emerged which is capable of describing the real world. If we look at an ontology as mathematicians, we perceive it as a directed graph that expresses knowledge about the world [43].

Currently the most widely-accepted definition of an ontology is: *"an explicit specification of a conceptualization"* [53]. A conceptualization refers to an abstract model of some phenomenon in the world and identifies the relevant concepts of that phenomenon. "Explicit" means that the types of concepts used and the constraints on their use are explicitly defined. It also expresses a shared conceptualization of a domain of interest. Shared does not necessarily mean globally shared, but only accepted by a sub group. The matching problem addressed in this chapter therefore stays unsolved by this definition. Most ontologies are full-fledged. However, their simple and easy version can be thought as lightweight ontologies consisting of human crafted classifications or taxonomies [52]. In taxonomies, the "is-a" relation expressing concept subsumption still matches the basic properties of backbone taxonomies:namely a lightweight ontology, the extension of a concept is a subset of the intersection of the extensions of its parent concepts. A Formal definition of lighweight ontologies is first introduced in [52]: "*A lightweight ontology is a triple $O = \langle N, E, C \rangle$, where N is a finite set of nodes, E is a set of Edges on N, such that $\langle N, E \rangle$ is a rooted tree, and C is a finite set of concepts expressed in a formal language F, such that for any node $n_i \in N$, there is one and only one concept $c_i \in C$, and, if n_i is the parent node for n_j, then $c_i \subseteq c_j$. The formal language F, used to encode concepts in C, belongs to the family of description logic languages and it may differ in its expressive power and reasoning capability. However, the least expressive one with still useful reasoning capabilities has been shown to be the propositional DL language, i.e., a DL language without a role for examples of practical applications of formal lightweight ontologies based on the propositional language*" [50].

The different types of lightweight ontologies are taxonomies, thesauri, business catalogs, faceted classifications, and user classifications. They are easier to be understood and built for an ordinary user. Automatic creation of a lightweight ontology by a normal user is shown in [47].

Different Kinds of Lightweight Ontologies

Based on their usage, two kinds of lightweight ontologies can be identified: [51, 54]

- Descriptive lightweight ontologies
- Classification lightweight ontologies (document classification lightweight ontologies)

The first is used for defining the meaning of terms as well the nature and structure of a domain. The second is used for describing, classifying, and accessing collections of documents, or more generally, data items. Due to this difference, formal classification lightweight ontologies have a different domain of interpretation for their concepts. Namely, the extension of a

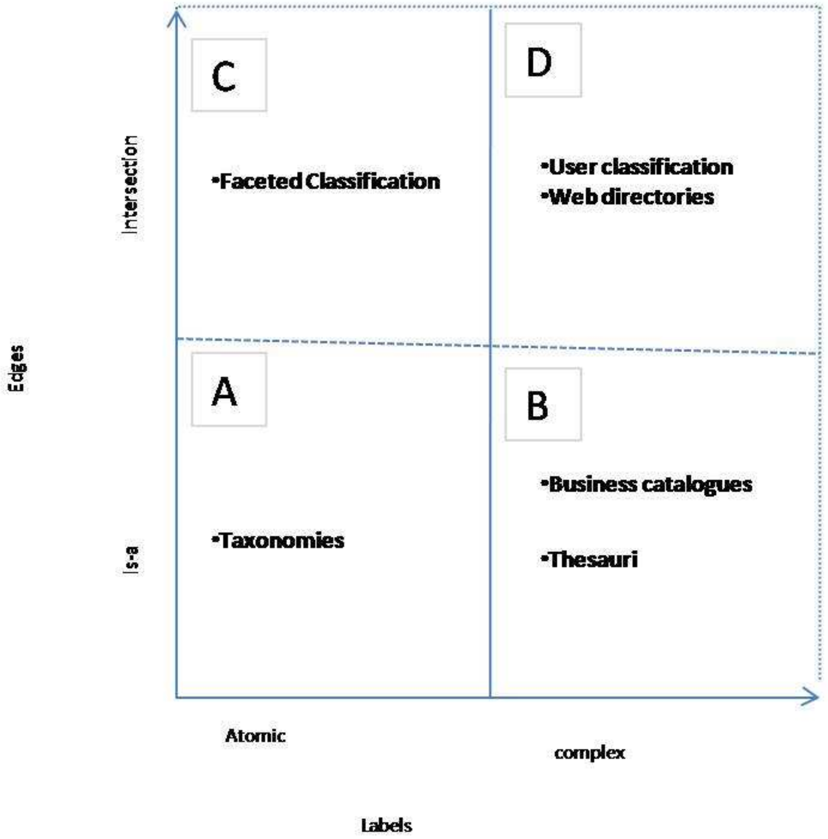


Figure 1.1: Lightweight ontology. Adapted from [52]

concept in a formal classification lightweight ontology [52] is the set of documents about the objects or individuals referred to by the (lexically defined) concept. For example, the extension of the concept "*India is the set of documents about India*".

In addition, any descriptive lightweight ontology can be used as a classification lightweight ontology, but not vice versa. Figure 1.1 shows the differences in properties of two kinds of ontologies. Classification lightweight ontologies are usually more complex than descriptive ones and the complexity is defined along two dimensions: Label complexity (atomic vs.complex labels) and edge complexity ("is-a" vs. "intersection" edge). Below, we list them from the simplest to the most complex classes.

Class A: atomic labels and "is-a" edges. This class usually has atomic concept labels [46](e.g., "bank","river") and "is-a" relations (e.g. "India is a child of Asia"). Typical examples of this category are (biological) taxonomies such as NCBI [97]. Further, ontologies in this category are descriptive.

Class B: complex labels and "is-a" edges. This class of ontologies are mostly descriptive but a few can be used for classification lightweight ontologies as well. Here, ontologies labels can be compound nouns which represent complex concepts and the relation between labels is usually the "is-a" relationship. Typical examples of this category are thesauri such as GLIN [96] and business catalogs such as UNSPSC [105]. A higher complexity of labels (with respect to category A) in these domain is required by the need of richer descriptions of indexing terms in thesauri and of e-commerce items in business catalogs. However, the business catalogs UNSPSC can be used as a descriptive ontology or as a classification ontology in which e-commerce items are classified. Note that even if the labels are complex, they are still mapped to atomic concepts in formal descriptive lightweight ontologies. In classification lightweight ontologies, complex labels represent a dimension of power of classification as one label can describe one complex concept that identifies a (very) specific set of documents. Moreover, complex labels can be mapped to complex concepts in formal classification ontologies, which allows for higher modularity in concept definitions. For instance, the concept "rice and fish" can be defined as the intersection of two concepts, "rice" and "fish", whereas the interpretation of formal concepts is the set of documents about rice (including pictures of rice as a kind of documents) and the interpretation of the latter concept is the set of documents about fish. Note that in formal descriptive lightweight ontologies, the extension of the concept "rice and fish" cannot be expressed as a function of the extension concept "baby" and the extension of concept "picture".

Class C: atomic labels and "intersection" edges. This class of ontologies usually represent single atomic concepts and intersection relations between labels which mean that the labels of a parent node specify the meaning of the label of its child node. For example, the parent node "Italy" specifies the meaning of its child node "picture", namely a "picture of Italy". A typical example of this category is a faceted classification such as Flamenco [23], in which child nodes represent aspects or facets of their parent nodes along atomic orthogonal dimensions (e.g.,time, space, function, material, etc). All ontologies in this category are classified as lightweight ontologies, for which the "intersection" relation creates an additional dimension of power of classification by allowing it to describe a specific set of documents through levels of categories in the ontology. Note that the interpretation domain of formal classification ontologies allows it to treat edges as the intersection of parent and child concepts and, therefore, compute concepts of nodes given their position in the ontology tree. For example, the intersection of the root concept "Italy" with its child concept "vacation" results in a concept whose extension is the set of documents about vacation in “Italy”, which is the actual meaning of the child node, given its position.

Class D: complex labels and “intersection” edges. In this class, ontology labels are represented as complex concepts and relationships between labels are usually “intersection” relationships. All ontologies in this category are classified as lightweight ontologies for which the combination of complex labels and “intersection” edges creates maximum classification power. Labels in this category can represent the names of individuals. These labels are mapped to concepts whose extension is the set of documents about the individuals (e.g., the extension of the concept “Asia” is the set of documents about the “Asia”). A Typical example of this category is web directories like DMoz [24] (in which web pages are classified) and user classification (in which email messages, favorites, and files are classified). Note that user classifications may have more complex labels and more “intersection” relations than web directories due to the fact that there are basically no rules and restrictions for user classifications which are commonly followed in web directories.

Key Applications

The key applications of light weight ontologies are document classification, semantic search, semantic matching, data integration.

- **Semantic Searching** Semantic searching seeks to improve traditional searches by leveraging XML [72] and RDF data from semantic networks to disambiguate semantic search queries and web text in order to increase the relevancy of results. In [110, 38] the author provides a list of semantic search systems. There are two major forms of search:navigation and search. In navigation search, the search engine is used as a navigation tool to navigate to a particular document. Semantic search is not applicable to navigational searches. In traditional searching, the user provides the search engine with a phrase which is intended to denote an object which the user is trying to gather/research information about. There is no particular document which the user knows about that he is trying to get. Rather, the user is trying to locate a number of documents which together will give him the information he is trying to find. Semantic Search lends itself well here. In general, semantic search is the problem of finding categories and documents (when applicable) classified into categories of (informal) lightweight ontologies, such that the found objects semantically correspond to a provided natural language query. Loosely speaking, semantic correspondence of an object to a query means that the meaning associated with the object is more specific or equivalent to the meaning given to the query means under common sense interpretation. For instance, a document about “Elephant” semantically corresponds to a query about “Asian elephants”. The approach reported in formalizing the above informal description and introducing a semantic search algorithm for lightweight classification ontologies populated with documents. The underlying idea is that the user query is converted to a concept in the manner presented earlier in this document and that the answer to the query is computed as the set of documents whose concepts are more specific or equivalent to the concept of the query. In order to reduce the computational complexity, the query is first run on the structure of the corresponding formal lightweight ontology in order to identify the scope of relevant nodes and then it is run on the documents populated in some of the nodes from the scope.

- **Data integration** Data integration is the process [17, 16] of combining data residing in different sources and providing the query with a unified view of these data. This process includes both commercial (when two similar companies need to merge their databases) and scientific (combining research results from different medical repositories) applications. General, data source applies to both commerce and science represented as a directed graph where is-a relation existing between nodes. Data integration can be seen as a semantic relation. A semantic relation between two nodes can be more/less general, equivalent, or disjoint. In the domain of lightweight ontologies [38] , semantic relations can be found between elements of controlled vocabularies, taxonomies, thesauri, business catalogs, faceted classifications, web directories, and user classifications. Found relations can then be used for enabling integration or inter-operation of web directories, for merging business catalogs, and so on.

- **Document classification** A set of documents put into a hierarchical classification is called a document classification. However, in lightweight ontologies a document is placed according to controlled vocabulary terms. It is placed in taxonomies, business catalogs, faceted classifications, web directories, or user classifications according to categories. In [37] the authors describe fully automatic classification of document into web directories based on the get-specific document classification algorithm. The underlying idea is that a web directory is converted into a formal lightweight ontology, that a document is assigned a concept, and that the document classification problem is then reduced to reasoning about subsumption on the formal lightweight ontology.

- **Background knowledge** Lightweight ontologies perform a crucial role for enriching the ontologies if there is any missing knowledge [3]. For example, taken an ontology consisting of "Net" and another ontology consisting of "network". In this case lightweight ontologies act as a mediator in order to get the appropriate meaning of two terms.

- **Indexing** Lightweight ontologies are used to index web material. Indexing languages are used in the Semantic web for classifying documents inside the browser [43].

1.5.2 Semantic Matching

Semantic matching [114, 128] is currently a topic of great interest among the semantic web community. Match acts as an operator which takes two graph-like structures, e.g., lightweight ontologies [38] such as LookSmarth [30], Yahoo [33], Google [29], or business catalogs, such as UNSPSC [105] and eCl@ss[13] or user classifications and produces mappings among the nodes of two graphs that correspond semantically to each other.

In general, we can divide matching approaches into

- Syntactic matching
- Semantic Matching

Syntactic matching is the task of comparing co-efficient [0,1] range [115]. This matching is a rather time consuming task. We concentrate our goal only on semantic matching as introduced in [46, 12]. The key intuition behind semantic matching is that we should calculate mappings by computing the semantic relation holding between the concepts assigned to nodes. Thus, for instance, two concepts can be equivalent, one can be more general than the other, and so on.

Basically, the semantic matching [46] approach is based on two key notions, namely:

- the concept of a label, which denotes the set of documents (data instances) that one would classify under a label the set encodes.

[13] http://www.eclass-serviceportal.com/

- the concept of a node, which denotes the set of documents (data instances) that one would classify under a node, given that it has a certain label and that it is in a certain position in a tree.

In semantic matching, existing relationship are denoted using set-theoretic semantics:equivalence ($\equiv$); more general ($\sqsupseteq$); less general ($\sqsubseteq$); disjointness($\perp$). These relationships hold between labels of nodes. If there is no relationship exists then a special "no" relation is returned. The relations are arranged according to their binding strength, i.e., from the strongest($\equiv$) to the weakest(no), with "more general" and "less general" relations having equal binding power. Here, the strongest semantic relationship always exists when two nodes have an equivalence relationship together. More general and Less general relationship are less stronger than equivalence relationship.

We define a matching as 4-tuple of the from: $\langle ID_{i,j}, c_i, d_j, R \rangle$, $i = 1, ..., N_C$; $j = 1, ..., N_D$ where $ID_{i,j}$, is a unique identifier of the given mapping element;c_i is the i-th node of the O1, N_C is the number of nodes in the O1, d_j is the j-th node of the O2, N_D is number of nodes in the O2; and R specificies a semantic relation which may hold between the concepts at nodes c_i and d_j. So, in light of the above discussion, semantic matching defines the following problem: given two lightweight ontologies T_C and T_D compute the $N_C \times N_D$ mapping element $\langle ID_{i,j}, c_i, d_j, R \rangle$ with $c_i \in T_C$, $i = 1, ..., N_C$, $d_j \in T_D$, $= 1, ..., N_D$ and R is the strongest semantic relation holding between *concepts at nodes* c_i, d_j. Since we are looking for the $N_C \times N_D$ correspondence, the cardinality of mapping between elements we are able to determine is $1 : N$. Also, these, if necessary, can be decomposed straightforwardly into mapping elements with the 1:1 cardinality.

We can describe semantic matching algorithm [47] via a running example. We consider respectively O1 and O2 shown in Figure 1, which are user defined classifications. The algorithm takes as inputs two ontologies and

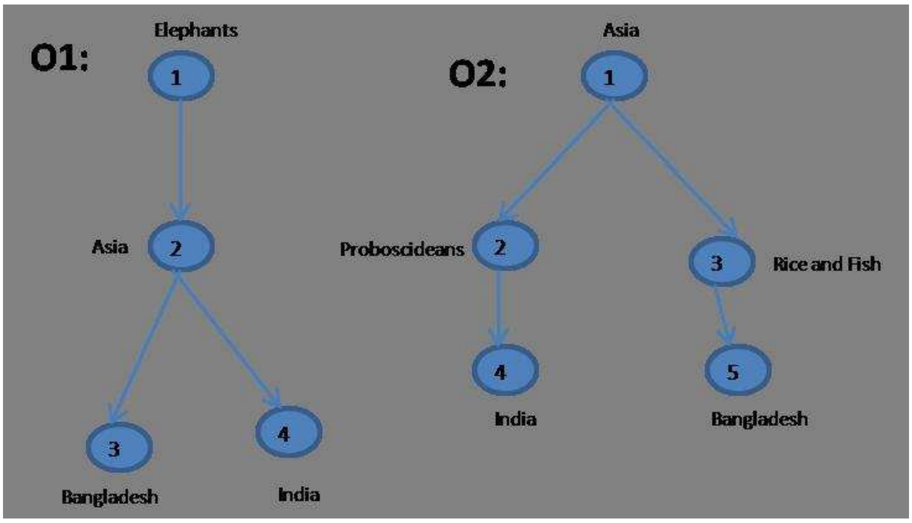


Figure 1.2: Lightweight ontology

outputs a set of mapping elements in four steps, as follows.

1.5.3 Example

- Step 1: for all labels L in two trees, compute concepts of labels C_L
- Step 2: for all nodes N in two trees,compute concepts at nodes, C_N
- Step 3: for all pairs of labels in two trees, compute relations among C_L's
- Step 4: for all pairs of nodes in two trees, compute relations among C_N's

In this algorithm, the first two steps represent the preprocessing phase, while the third and fourth steps are the element level and structure level matching respectively.

Step 1. For all labels L in two trees, compute concepts of labels. Labels represent concepts themselves. For example, the label "elephant" can be characterized as a set of documents which describe the elephant.

However, these labels are mostly written in natural language and natural language presents many ambiguities. For instance, there are several possible way to represent the same concept: “elephant” means “five-toed pachyderm ” or “the symbol of the Republican Party; introduced in cartoons by Thomas Nast in 1874 ”. In order to remove the ambiguities, natural language labels are translated into internal language such as a propositional descriptive language. Specifically, atomic formulas are atomic concepts, written as single words or multi-words. Complex formulas are obtained by combining atomic concepts using logical operators such as conjunction ($\cap$), disjunction ($\cup$), and negation ($\neg$). Note that negation can only be applied to atomic concepts. There are also comparison operators such as less general ($\sqsubseteq$), more general ($\sqsupseteq$), and equivalence ($\equiv$). The interpretation of these operators is the standard set-theoretic interpretation. The reasons for choosing a simple propositional description logics language are as follows. First, given its set-theoretic interpretation, it “maps” naturally to the real world semantics. Second, natural language labels used in classifications and XML schemas are usually short expressions or phrases are having simple structure. These phrases can be converted into a formula in our knowledge representation formalism with no or little loss in the meaning . Finally, these formulas can be converted into equivalent formulas in a propositional logic language with boolean semantics. Thus, technically, the concept of a label is the propositional formula which stands for the set of data instances (documents) that one would classify under a label it encodes.

Computing atomic concepts, as they are denoted by atomic labels (namely, labels of single words or multi-words), as the senses provided by WordNet. In the simplest case, an atomic label generates an atomic concept.

However, atomic labels with multiple senses or labels with multiple words generate complex concepts. The translation process from labels to concepts is accomplished as follows (note that the first two steps are common to many matching approaches):

- Tokenization: labels of nodes are parsed by a tokenizer which recognizes punctuation, cases, digits, stop characters, etc. Thus for instance, “Rice and Fish” becomes $\langle Rice, andFish \rangle$.
- Lemmatization: tokens at labels are lemmatized, namely they are morphologically analyzed in order to find all their possible basic forms. Thus, for instance, “Elephants” is associated with its singular form, “elephant”.
- Building atomic concepts: WordNet is queried to extract the senses of lemma at tokens identified during step 2. For example, the label “elephants” has the only one token “elephants” and one lemma elephant, and from WordNet we find out that image has eight senses, seven as a noun and one as a verb.
- Building complex concepts: all existing tokens that are propositions, punctuation marks, conjunctions (or strings with similar roles) are translated into logical connectives and used to build complex concepts out of the atomic concepts built in step 3 above. Thus, commas and conjunctions are translated into disjunctions, prepositions like “of” and “in” are translated into conjunctions, and so on. For instance, the concept of the label “Rice and Fish”,$C_{RiceandFish}$ are computed as $C_{RiceandFish} = \langle$ Rice,senseswn#1$\sqcup$ Fish,senseswn#2$\rangle$

After the first phase, all labels have been translated into sentences in the internal concept language.

Step 2 For all nodes N in two trees, compute concepts at nodes We analyze the meanings of the positions of labels at nodes in the trees. By doing this, concepts of labels are extended to concepts at nodes, C_N. This is required to capture the knowledge residing in the structure of a tree, namely the context in which the given concept at label occurs.Technically, concepts at nodes are written in the same propositional logic language as concepts of labels. Thus, for example C_4 in O1 in Figure 1 (node label "India") is computed by taking the intersection of the concepts of the labels "elephants", "Asia", "India", namely C_4= elephants$\sqcap$ Asia $\sqcap$ India stands for the concepts describing all the documents about the Indian "elephant".

Step 3. For all pairs of labels in two trees, compute relations among concepts of labels. Relations between concepts of labels are computed by using a library of element level matchers: see table 1.2.

The first column contains the name of the matchers. The second column lists the order in which they are executed. The third column introduces the matchers' approximation level. The relations produced by a matcher with the first approximation level are always correct. For example, according to WordNet [89], the concept denoted by the label "Asia" has a first sense which is a homonym to the first sense of the concept denoted by the label "India". Therefore, India is less general than "Asia". Notice that, with WordNet, we cannot compute overlap, and the fact that WordNet does not provide us with any information is taken to mean that two concepts have no relation.

Matcher name	Execution order	Approximation level	Matcher type	Schema Info
WordNet	1	1	Sense-based	WordNet senses
Prefix	2	2	String-based	Labels
Suffix	3	2	String-based	Labels
Edit distance	4	2	String-based	Labels
Ngram	5	2	String-based	Labels

Table 1.2: Element level Semantic Matcher

The relations produced by a matcher with the send approximate levels are likely to be correct (e.g. net=network, but hot=hotel by prefix). The WordNet matcher has two WordNet senses in input and computes equivalence, generality, and disjointness relations. String based matchers have two labels as input. These compute only equivalence relations (e.g. equivalence holds if the weighted distance between the input strings is lower than a threshold), see. String based matchers are used off where WordNet fails to find a relation. The result of step 3 is a matrix of relations holding between atomic concepts of labels. A part of it, for the example of Figure 1.1, is shown in Table 1.2.

	C_{Asia}	$C_{Proboscideans}$	C_{rice}	C_{Fish}	C_{India}	$C_{Bangladesh}$
$C_{Elephant}$		$=$				
C_{Asia}	$=$				$\sqsupseteq$	$\sqsupseteq$
$C_{Bangladesh}$	$\sqsubseteq$				$\perp$	$=$
C_{India}	$\sqsubseteq$				$=$	$\perp$

Table 1.3: cLabsMatrix relation holding among atomic concepts of labels

Step 4. For all pairs of nodes in two trees, compute relations among concepts at nodes. This mapping problem is reformulated into a set of node matching problems. It cannot be solved by asking an oracle or a knowledge base. The key idea of this approach is to translate the node matching problem into a propositional validity problem. It tries to prove that axioms→ rel(context1, context2) is valid. Axioms, context1 and context2 are defined in the tree matching algorithm. nodeMatch checks for sentence validity by proving that its negation is unsatisfiable. The algorithm uses, e.g., a DPLL-based SAT solver. From the example, the concept at node “elephants (c1)“ in A1, is more general than the concept at node “Proboscidean (C2)”in A2. Notice that this table, contrary to Table 1, is complete in the sense that we have a semantic relation between any pair of concepts of nodes. However, the situation is not as nice as it looks as, in most matching applications, intersection gives us no useful information (it is not easy to know which document should be discarded or kept) it suggests that, when we have intersection, we iterate and refine the matching results; however, thus, we have not been able to pursue this line of research.

1.5.4 Summary

The aim of semantic matching is to find semantic correspondences between classification, taxonomies, web directories, and business catalogs that refer to lightweight ontologies. To date a lot of work has been done in the field of matching, but there is still the issue of missing background knowledge. This issue can be solved in theory by using central/universal knowledge. Currently, there is no classification or ontology which can act as a universal classification. Some research work has been done in the library sciences field, but there has been no concrete works in other fields such as Computer science, biological sciences, etc.

1.6 Role of Controlled vocabulary in Semantic Matching

In the spite of explosive growth of the Internet, information relevant to user is often unavailable even when using the latest browsers. At the same time, there is an ever increasing number of documents that vary widely in content, format and quality. The documents often change in content and location because they do not belong to any kind of centralized control. On the other hand, there is a huge number of unknown users with extremely diverse needs, skills, education, and cultural and language backgrounds. One of the solutions to these problems might be to use standard terms with meaning, this can be termed as controlled vocabulary

(CV) [39]. Though there is no specific notion of CV, we can define it as a set of concepts or preferred terms and existing relations among them. For example, thesauri, WordNet [89], MeSH [116], LCSH [94], all kinds of ontologies, etc. are sorts of CVs. These CVs are used to matching purpose that makes more flexible for information extraction. In a semantic or controlled vocabulary [46], a matching operator takes two-graph like structures, for instance ontologies or classifications and produces matching relationship among them. This semantic matching system is based on two-key notions. One of them is the concept of nodes and other is the concept of labels. In semantic matching, labels are written in natural language. These labels are disambiguated using a lexicon [89]. In this case, they are working as a background knowledge. In this chapter, we will see the contribution of CV for information retrieval purpose and review the main applications of controlled vocabularies.

1.6.1 Different Kind of Controlled Vocabulary

In our case, we can classify our controlled vocabularies based on nature, construction perspective and usage. These constructions are based on regions, countries, products, services, vertical markets, clients, customer alliances, structure subsidiaries histories and cultures etc. For instance, two words “Center” and “Centre” both are having same meaning but different spelling in different regions and cultures.

We can classify controlled vocabularies in the following way:

1. **General controlled vocabulary** This class of controlled vocabulary is mainly included in usage and existing relationships among the concepts and entities. For example, the most prominent representation of these vocabularies are Thesaurus, WordNet, Classification, Directories, Lightweight Ontologies [57], etc.

- Thesaurus: A thesaurus can be defined as a "controlled vocabulary that includes synonyms, hierarchies and associative relationships among terms to help users to find the information they need" [39]. For example, two users are looking for information "Automobile". One may use the term "Car" while the other may use "Auto". Each of them queries the same information with different terms, but these terms belong to same concept. So, the success of finding relevant documents varies based on demand and context. To address the problem, thesauri map variations in terms (synonyms, abbreviations, acronyms and altered spelling) of a single preferred term for each concept. For document indexer, the thesauri provide the index term to be used to describe each concept. This enforces consistency of document indexing. For users of a Web site, the thesauri work in the background, mapping their keywords onto single preferred terms, so they can be presented with the complete set of relevant documents.
- WordNet: a human compiled electronic dictionary which is one kind of ontology that expresses meanings of bounded terms. It was developed by Prof. George Miller at Princeton University. It mainly builds up on a lexical knowledge base born out from psycholinguistic research into the human lexicon. It has applications in different fields of research, sense disambiguation, semantic tagging and information retrieval [89].
- EuroWordNet: a European project for WordNet. The aim of this project is to develop multilingual dictionaries with WordNet for several European languages. In this project based on WordNet, each individual net is linked to a central system which is called Inter-Lingual-Index. Each net is composed of about 30,000 synsets and 50,000 entries [34].

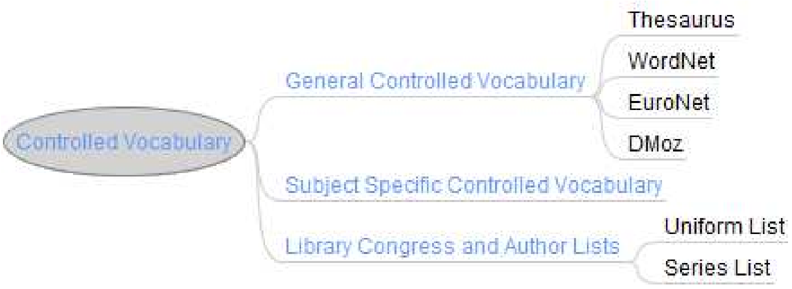
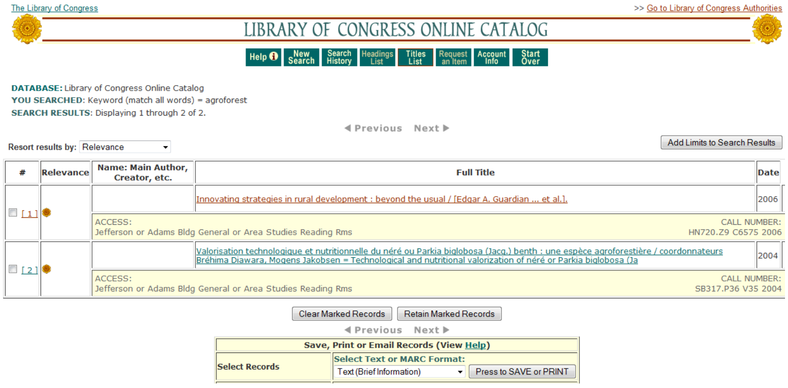

Controlled Vocabulary
General Controlled Vocabulary
Thesaurus
WordNet
EuroNet
DMoz
Subject Specific Controlled Vocabulary
Library Congress and Author Lists
Uniform List
Series List

Figure 1.3: Different Kind of Controlled Vocabulary

- DMoz: an open directory project which is most panoptic human-edited directory of the Web. It is constructed and maintained by a vast, global community of volunteer editors. Web content is growing at staggering rates. Search engines are increasingly unable to provide useful results to search queries. The open directory provides a way to keep the Internet classified itself. It uses standard terms to tag the directories so that anyone can browse it [24].

2. **Subject specific controlled vocabulary (SSCV)** Construction of sentences, words and data are most of the time used in subject specific controlled vocabularies, for example languages to express chronology, hypothesis, comparison, etc. Typically an SSCV is expressed as keywords, key phrases or classification codes that describe the theme of the resource. In the library sciences, due to the ever-increasing number of records, bibliographic systems are facing difficulties. Documents in library system are heterogeneous: some of them provide few hints, some are disparate while in others structural tags are sometimes not used properly, which results in inefficiency in extracting documents. However, controlled vocabularies which have traditionally been used in libraries, could serve as good-quality structures for subject browsing among entire documents. Subject heading systems and thesauri



DATABASE: Library of Congress Online Catalog
YOU SEARCHED: Keyword (match all words) = agroforest
SEARCH RESULTS: Displaying 1 through 2 of 2.

◀ Previous Next ▶

Resort results by: Relevance

Add Limits to Search Results

#	Relevance	Name: Main Author, Creator, etc.	Full Title	Date
□ [1]	●		Innovating strategies in rural development : beyond the usual / [Edgar A. Guardian ... et al.].	2006
		ACCESS: Jefferson or Adams Bldg General or Area Studies Reading Rms	CALL NUMBER: HN720.Z9 C6575 2006	
□ [2]	●		Valorisation technologique et nutritionnelle du néré ou Parkia biglobosa (Jacq.) benth : une espèce agroforestière / coordonnateurs Bréhima Diawara, Mogens Jakobsen = Technological and nutritional valorization of néré or Parkia biglobosa (Ja	2004
		ACCESS: Jefferson or Adams Bldg General or Area Studies Reading Rms	CALL NUMBER: SB317.P36 V35 2004	

Clear Marked Records Retain Marked Records

◀ Previous Next ▶

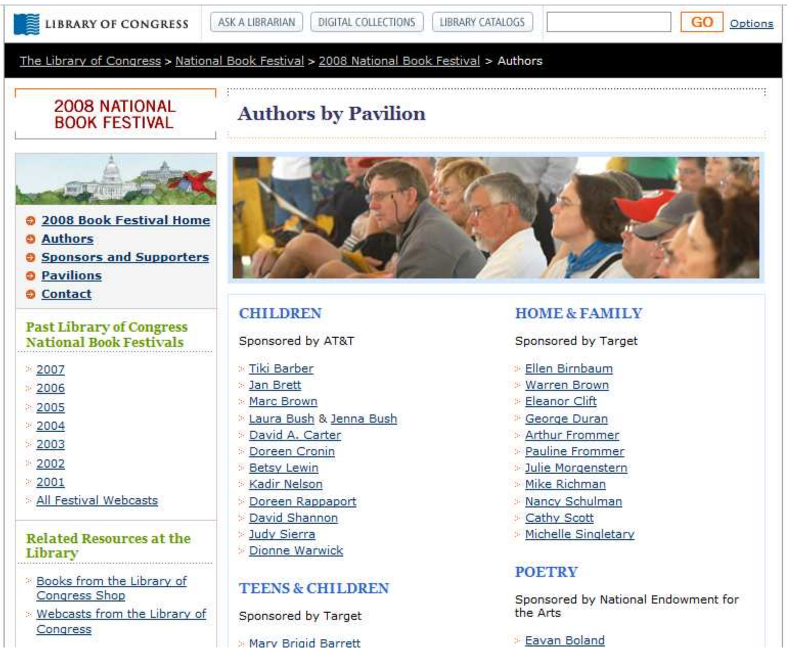
Save, Print or Email Records (View Help)	
Select Records	Select Text or MARC Format: Text (Brief Information) Press to SAVE or PRINT

Figure 1.4: Library of Congress Online Catalog

that have traditionally been developed for subject indexing that would describe topics of the document more specifically [95].

3. **Library of Congress and Authors List** The Semantic Web and library communities have both been working toward the same set of goals: naming concepts, naming entities and bringing different forms of those names together. The Semantic Web's efforts toward this end are relatively new, whereas libraries have been doing work in this area for hundreds of years. Vocabularies developed in libraries, particularly at the Library of Congress, are sophisticated and advanced in searching and representation. Libraries have a long-standing history of developing, implementing and providing tools and services that encourage the use of numerous controlled vocabularies. When the naming conventions are translated into Semantic Web technologies, they will help realize Berners-Lee's dream [56]. Furthermore, the roles of libraries in the Semantic Web are as follows:

   - Exposing collections-using Semantic Web technologies to make content available
   - Web'ifying,



LIBRARY OF CONGRESS | ASK A LIBRARIAN | DIGITAL COLLECTIONS | LIBRARY CATALOGS | GO | Options

The Library of Congress > National Book Festival > 2008 National Book Festival > Authors

2008 NATIONAL BOOK FESTIVAL

Authors by Pavilion

- 2008 Book Festival Home
- Authors
- Sponsors and Supporters
- Pavilions
- Contact

Past Library of Congress National Book Festivals

- 2007
- 2006
- 2005
- 2004
- 2003
- 2002
- 2001
- All Festival Webcasts

Related Resources at the Library

- Books from the Library of Congress Shop
- Webcasts from the Library of Congress

CHILDREN

Sponsored by AT&T

- Tiki Barber
- Jan Brett
- Marc Brown
- Laura Bush & Jenna Bush
- David A. Carter
- Doreen Cronin
- Betsy Lewin
- Kadir Nelson
- Doreen Rappaport
- David Shannon
- Judy Sierra
- Dionne Warwick

TEENS & CHILDREN

Sponsored by Target

- Mary Brigid Barrett

HOME & FAMILY

Sponsored by Target

- Ellen Birnbaum
- Warren Brown
- Eleanor Clift
- George Duran
- Arthur Frommer
- Pauline Frommer
- Julie Morgenstern
- Mike Richman
- Nancy Schulman
- Cathy Scott
- Michelle Singletary

POETRY

Sponsored by National Endowment for the Arts

- Eavan Boland

Figure 1.5: Library Congress Author List

- Thesaurus/Mappings/Services
- Shared Learning.
- Persistence.

As all of the above roles are equally important, the intuition to move controlled vocabularies into a standard to which web services can gain easy access to information management.

Conforming all these vocabularies to Semantic Web standards such as controlled vocabularies will provide limitless opportunities to use them in different ways. This can make possible searching and browsing diverse records, verifying and identifying particular authors and browsing sets of topics related to a particular concept [79]. Authors List can be categorized into two ways:

- Uniform List: This category includes all universal names. For example, the “Bible”, the “Gita”, the “Quran”, the “Tripod and the "Lake of Garda" etc. This kind of series list of controlled vocabularies are included in different consecutive names. From a unique list it is easier to match the concepts they represent.

- Series List: This category includes the series of same name with the different themes such as "Terminator-1", "Terminator-2", and "Terminator-3".

1.7 Applications

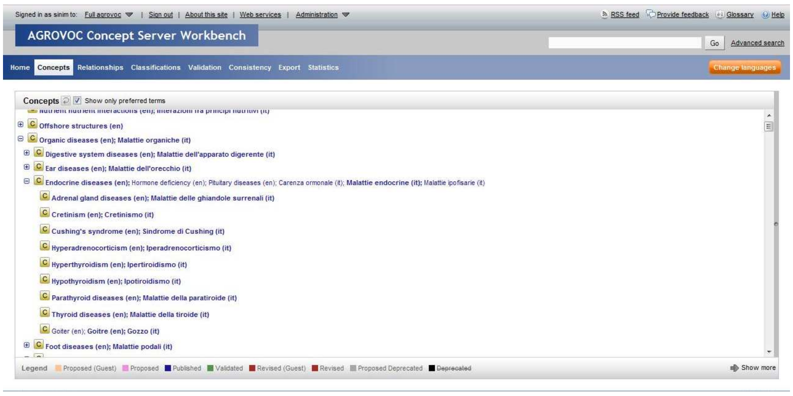
1.7.1 Applications for managing controlled vocabularies

- Traditional Controlled Vocabulary tools.

  The vocabulary which is used in legacy systems is called the traditional vocabulary. For example, the AGROVOC [117] thesaurus is mainly in relational database format and is published on the web site for browsing and navigating concepts and their relations. It was previously available only in four languages. Now it is available in 19 languages. Major drawbacks of traditional controlled vocabularies are that they were not well structured, they were only text format or SQL format, their relationships were not well defined, there was no semantics between the concepts and there was no Unified Resource Identifier (URI) for locating the concepts.

- A Modern Controlled vocabulary collaborative management system: Example of AGROVOC Concept Server Workbench (ACSW).

  Modern controlled vocabularies [21] are one kind of lightweight ontologies with well defined multiple formats (SKOS, RDF, and OWL etc). In this vocabulary, each concept is assigned a URL. Using this URI, one can populate concept information and use this information



Concepts ☑ Show only preferred terms

Nutrient nutrient interactions (en); Interazioni fra principi nutritivi (it)
⊞ C Offshore structures (en)
⊟ C Organic diseases (en); Malattie organiche (it)
  ⊞ C Digestive system diseases (en); Malattie dell'apparato digerente (it)
  ⊞ C Ear diseases (en); Malattie dell'orecchio (it)
  ⊟ C Endocrine diseases (en); Hormone deficiency (en); Pituitary diseases (en); Carenza ormonale (it); Malattie endocrine (it); Malattie ipofisarie (it)
    C Adrenal gland diseases (en); Malattie delle ghiandole surrenali (it)
    C Cretinism (en); Cretinismo (it)
    C Cushing's syndrome (en); Sindrome di Cushing (it)
    C Hyperadrenocorticism (en); Iperadrenocorticismo (it)
    C Hyperthyroidism (en); Ipertiroidismo (it)
    C Hypothyroidism (en); Ipotiroidismo (it)
    C Parathyroid diseases (en); Malattie della paratiroide (it)
    C Thyroid diseases (en); Malattie della tiroide (it)
    C Goiter (en); Goitre (en); Gozzo (it)
  ⊞ C Foot diseases (en); Malattie podali (it)

Figure 1.6: AGROVOC Workbench

for further research. One example of modern controlled vocabulary is AGROVOC Concept Server (ACS) [19].

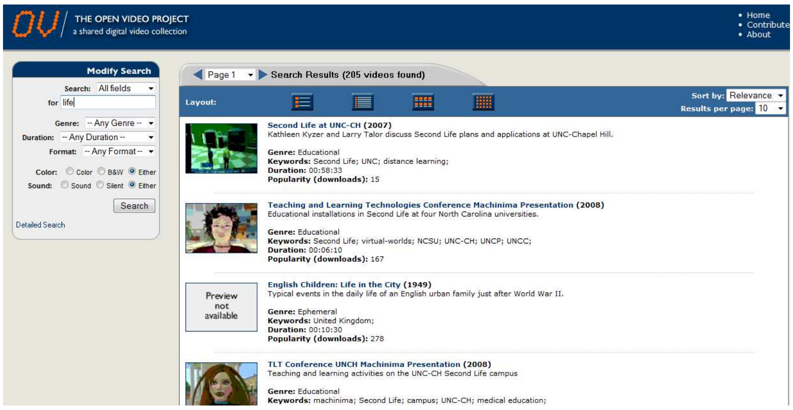
1.7.2 Applications for exploiting controlled vocabularies

- Background Knowledge: Controlled vocabularies are used in subject indexing schemes, subject headings, thesauri and taxonomies to provide a way to organize knowledge for subsequent retrieval [3, 36]. The Controlled vocabulary strategy assigns the use of predefined, authorized terms that have been preselected by the designer of the vocabulary. For easy accessing to the digital information and library catalogs, tags are carefully selected from the words and phrases in a controlled vocabulary. CV controls the use of synonyms (and near-synonyms) by establishing a single form of the term. This ensures that indexers apply the same terms to describe the same or similar concepts, thus reducing the probability that relevant resources will be missed during a user search. The biggest advantage to controlled vocabularies is that once you find the correct term, most of the information you need is grouped together in one place, saving you the time of having to search under all of the other synonyms for that term. In large organizations, controlled vocabularies may be introduced to improve inter-departmental communication. The use of controlled vocabularies ensures that everyone is using the same word to mean the same thing. This consistency of terms is one of the most important concepts in technical writing and knowledge management, where effort is expended to use the same word throughout a document organization instead of slightly different ones refers to the same thing.

- Document annotation: The objective of document annotation is to use appropriate terms so that machines can easily understand and correctly classify the documents, allowing the user easily access while searching or browsing. For example, Clusty [28], Vivisimo [32], Swoogle [31], etc. are classified documents under pre-defined keywords or terms so that one can go to specific locations to find the needed information [70]. Furthermore, document annotation is needed for building knowledge bases that will be used in the future Web and existing large sets of corporas. However, existing information retrieval systems use string matching techniques for full-text search or key phrase search. Thus, a major problem with these systems is overlapping the matching terms or matching results. To overcome these difficulties, more semantic information should be added to matching techniques. The present NLP (natural language processing) techniques cannot provide the complete solution. There is more work to be done. In additional, document annotation can help to improve the performance of information extraction.

- Information retrieval and extraction: WordNet has been used as a comprehensive semantic lexicon in a module for full text message retrieval as a communication aid, in which queries are expanded through keyword design. In [61, 27], automatic construction of thesauri, based on the occurrences determined by the automatic statistical identification of semantic relations is used for text categorization. English words can have different meanings or the same meaning with different structures or descriptions. For example, “center” and “centre” have the same meaning but different spelling for American and British English. Conversely, the same words can have different meanings, for example “bank” means “river side” or “financial institution”. It is hard to classify documents or satisfy user queries according to the meaning of words. Text categorization is the process of categorizing the document under a specific class. WordNet lexical information builds a relation between sentences and coherent categories. [112] describes an algorithm for text categorization using WordNet.

- Audio and Video retrieval: In the digital age, the most challenge is to handle the huge amount of hyper-media or non-textual information on the Web. For example, an YouTube [129], over 150,000 videos are uploaded and 100,000,000 queries are performed every/day. In order to control these high volumes of hyper-media information, information must be used and used in the right way. For instance, the multimedia miner [106] is a prototype to extract multimedia information and knowledge from the web to generate conceptual hierarchies for interactive information retrieval and build multi-dimensional cubes for multimedia data. Finally, WordNet or Thesaurus are used in query expansion for TV or radio programs to index the news automatically. It has some drawbacks; for instance, it is not domain specific and it is not possible to find relationships between terms with different partsofspeech.



Search: All fields
for life
Genre: -- Any Genre --
Duration: -- Any Duration --
Format: -- Any Format --
Color: ○ Color ○ B&W ● Either
Sound: ○ Sound ○ Silent ● Either

Page 1 Search Results (205 videos found)

Sort by: Relevance
Results per page: 10

Second Life at UNC-CH (2007)
Kathleen Kyzer and Larry Talor discuss Second Life plans and applications at UNC-Chapel Hill.

Genre: Educational
Keywords: Second Life; UNC; distance learning;
Duration: 00:58:33
Popularity (downloads): 15

Teaching and Learning Technologies Conference Machinima Presentation (2008)
Educational installations in Second Life at four North Carolina universities.

Genre: Educational
Keywords: Second Life; virtual-worlds; NCSU; UNC-CH; UNCP; UNCC;
Duration: 00:06:10
Popularity (downloads): 167

English Children: Life in the City (1949)
Typical events in the daily life of an English urban family just after World War II.

Genre: Ephemeral
Keywords: United Kingdom;
Duration: 00:10:30
Popularity (downloads): 278

TLT Conference UNCH Machinima Presentation (2008)
Teaching and learning activities on the UNC-CH Second Life campus

Genre: Educational
Keywords: machinima; Second Life; campus; UNC-CH; medical education;

Figure 1.7: Video Indexing

1.7.3 Why Controlled vocabulary on the web

The endlessly growth of information resources on the web demands better classification. This classification is needed to browse web pages more smoothly. Previous orthodox information resources were not consistent because of changing static to dynamic pages on the Web. After changing those information resources to modern information resources, a more consistent to categorization is needed. However, the problem was not only browsing the pages but also consisting of qualities of Web sites content. To overcome this problem, a change to apply online vocabulary resources is needed to help end users to find what they are looking for. Furthermore, social networking, linking data, Flickr [40], Google Maps [86] and inter-company collaboration, etc. brings have a common ground which further necessitates a controlled vocabulary.

1.7.4 Controlled vocabularies in Semantic matching

As information and communication technology is expanding day by day, it is essential to access that information easier and efficiently. Semantic heterogeneity is the main obstacle to ease of information extraction, which

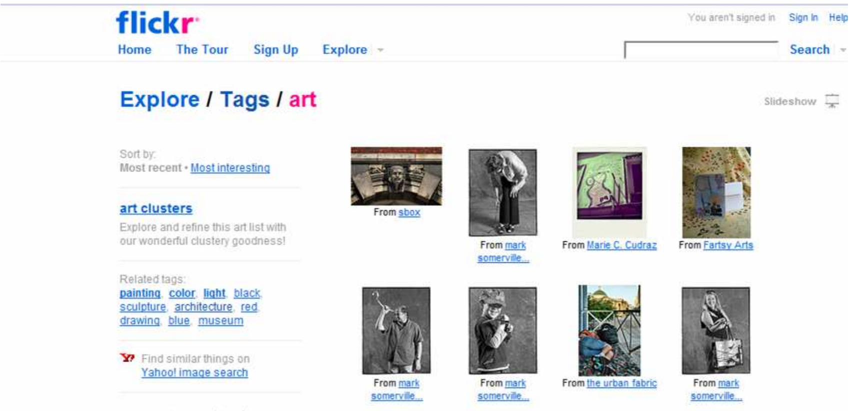

flickr
You aren't signed in Sign In Help
Home The Tour Sign Up Explore
Search
Explore / Tags / art
Slideshow
Sort by:
Most recent • Most interesting
art clusters
Explore and refine this art list with our wonderful clustery goodness!
Related tags:
painting, color, light, black, sculpture, architecture, red, drawing, blue, museum
Find similar things on Yahoo! image search
From sbox
From mark somerville...
From Marie C. Cudraz
From Fartsy Arts
From mark somerville...
From mark somerville...
From the urban fabric
From mark somerville...

Figure 1.8: Controlled Vocabulary used as Tagging in Flickr

is growing critical day by day as the database of vocabularies is getting larger. One possible solution might be that of matching [17, 115, 120, 20, 101, 66, 99, 48, 71]. A matcher acts as an operator and takes two classification, or ontologies, and discovers the similarities and dissimilarities among entities that exist in classification hierarchies. These matchers are used in string matching techniques or Scoring techniques to find the match results. However, Semantic Matching is introduced in [88, 47] and does not consider straight string matching techniques for matching purposes. It takes two classifications and produces matches. This matching system is based on two key notation; one is the concept of a node, the other the concept of a label. However, background knowledge is a major factor for its functionality. WordNet plays a vital role in this took. More precisely, [7, 8], introduced concept facets for matching two controlled vocabularies for accessing information more easily.

1.7.5 Summary

Controlled vocabularies plays a vital role in information integration and information retrieval. They can be useful in linking information, discovering knowledge, knowledge residing on the Web. However, a complete universal controlled vocabulary has yet to be done assembled. It is extremely important in the fields of information science, earth science, biological science, cyber science and medical science to establish common vocabularies so that anyone can access information even if he does not understand full the language. We discuss pros and cons, different kind of controlled vocabularies, and mention some on going work in this domain.

Chapter 2

Problem

A knowledge organization system (KOS) consists of vocabularies [114, 35]: ontologies, thesauri, classification, terminologies, etc. The structure of these vocabularies changes from time to time. Thus, it creates vocabularies heterogeneity problems.

Some work in this chapter has been supported by the FAO KOS mapping project [85].

In this chapter, we illustrate vocabularies presentation, factors of heterogeneities, different kind of heterogeneities problems [59] and our research problem that we are going to solve.

2.1 Vocabulary Presentation

2.1.1 Thesauri

Thesauri are a kind of KOS, where specialists group terms together by judging their similar meaning. The most well-known thesaurus in the world is the historical thesaurus of the "Oxford English Dictionary", which contains more than 920,000 words and meanings.

Terms are the basic unit for building thesauri [61, 25]. They are categorized into descriptor (also called preferred term) and non-descriptor (non-preferred term). A descriptor term is a term that is used for controlling the indexing in the thesauri and the rest of terms are considered non-descriptor terms.

In thesauri, terms are associated with each other by relationships. These relationships can be divided into three types:

- Hierarchical relationships include broader terms (BTs) and narrower terms (NTs). BTs or hyperonym are more general terms, e.g. "Agriculture" is a broader term then "Agriculture Industry". Similarly, a narrow Term (NT) or hyponym is a more specific term, e.g. "Agriculture Industry" is narrow term then "Agriculture". Both of them are associated with class type relationships, as well as "IS-A" relationships.

- Equivalency relationships are used primarily to connect synonyms and near-synonyms.

- Associative relationships are used to connect two related terms whose relationship is neither hierarchical nor equivalent. This relationship is described by the indicator "Related Term" (RT). This relationship should be applied with caution, since excessive use of RT will reduce specificity in searches.

The main usage of thesauri is for information retrieval. They are kinds of controlled vocabularies so they are used in indexing, tagging, subject cataloging, etc. We found these thesauri in TEXT, XML, RDF and OWL format. For example, the AGROVOC thesaurus from FAO is represented in OWL, TXT, SKOS, RDF, and SQL format.

2.1.2 Ontology

Ontology [53] can be defined as the formalization (through some textual or graphical description) of a conceptualization. An ontology can be used to share knowledge by using similar vocabulary, semantics, and relationships among concepts of a particular domain. In fact, ontologies are very practical for explaining meta data terms and organizing domain knowledge in a structured and standard way. This type of standardization facilitates reuse and enables applications to cooperate with one another more efficiently. Recently, classification or conceptual models have been promoted as ontology. Ontology contains most of the features of entity relationship models.

The typical feature of an ontology is that it is based on a given logic theory. Thus, their interpretation is not left to the users that read the diagrams or to the database management systems that implements them; it is specified explicitly by set of inference rules. The semantics provides the rules for interpreting the syntax. It does not provide the meaning directly but constraints the possible interpretations of what is declared.

Furthermore, ontologies are presented in a specific language. In fact, there are a large variety of languages for presenting them, for example Web ontology language (OWL) and Resource Description framework (RDF) which allow the definition of taxonomies and relations between concepts. Apart from RDF and OWL, one should mention F-logic as a logic-based ontology representation.

2.1.3 Classification

The word "taxonomy" come from the Greek words "taxis" (meaning order or arrangement) and "nomos" (law or science). It is an ordered set of taxons (classes). Typically, these are organized by subtype-supertype relationships, also called parent-child relationships [114, 94, 24, 43]. For example "Agricultural forest" *is subtype of* "Agriculture".

A taxonomy is a type of classification or directory that is used by a library for cataloging books or information, by company for presenting products for sale, or by the web for indexing information for easy navigation, e.g., Google, Yahoo, DMOZ, and LCC etc. These classifications are hierarchies of folders identified by labels. The semantics of these folders is given by the items they ultimately contain. Obviously, each independent entity tends to develop its own directory based on its own needs and tastes.

The culture of classification first introduced by library science, shows how to classify documents under classification labels. In life science, classifications are used to present the tree of species.

Recently, this information has been stored in XML or RDF files.

2.1.4 Databases

In databases, data is stored in predefined tables. A database specifies the names of the tables as well as their types: the names and types of the columns of each table. A database also includes a key for each table: a subset of the columns that uniquely identifies each row. Finally, a column in a table may be specified as a foreign key pointing to a column in another table. This is used to keep referential constraints among various entities. Finally, it is worth mentioning widely used languages for specifying relational schemas, such as Structured Query Language (SQL).

These support many modeling capabilities, such as user-defined types, aggregation, generalization, etc. Furthermore, RDF, SKOS and OWL have stored data in triple storage where data is stored as subject, predicate and object. To manipulate this kind of data, the SPARQL[1] query language is used. There are some existing tool which are used for creating databases,

[1] http://www.w3.org/TR/rdf-sparql-query/

for example, PostgreSql[2], Sql, and Oracle[3] etc. for relational databases, and Jena[4], and Sesame[5], etc. for RDF storage.

2.1.5 Terminology

Terminology considers terms and their use. It consist of words and compound words that work in specific contexts. It should not be confused with "terms" in colloquial usages, the shortened form of technical terms (or term of art), which are defined within a discipline or specialty field. For example, Terms of fisheries mean all terms from the domain of fisheries are included with its labels and their definition so that people can understand the terms or concepts. It does not have any kinds of relationships like thesauri (BT, NT, RT, and UF) or ontologies (*is-a* and *part-of*). This is mainly used for documentation and promoting correct usage. It is not limited to a single language, it does not have any particular structures. It mainly consists of a text file with term description.

2.2 Challenges of Matching

The manipulation of vocabularies is a very difficult task [114, 17] due to different factors involved that create heterogeneity problems.

2.2.1 Factors of heterogeneity problem

- time
- place
- structure

[2] http://www.postgresql.org/

[3] http://www.oracle.com/index.html

[4] http://jena.sourceforge.net/

[5] http://www.openrdf.org/

- culture diversity
- different vocabulary specialists

Firstly, vocabulary changes with times. For example a word “kedara” means in Bangla language “chair”. Now people do not use “kedara”, everybody use “chair”. Secondly, vocabularies change with place. For example, “India” has one language in every 50 miles. Thirdly, vocabularies are not written in specific formats or there are no universal formats. Fourthly, vocabularies change with culture. For example, English people use “centre” whereas American people use “center”. Lastly, written vocabularies can be different for different specialists with different views.

2.2.2 Different heterogeneity

The main purpose of matching vocabularies is overcoming the heterogeneity problem. The problem does not lie solely in the difference of ultimate goals of the applications according to which they have been designed, or in the expression formalisms in which the vocabularies have been encoded. Defining factors creates many heterogeneity problems. We present here some typical heterogeneity problems [114, 8, 26].

- **Syntactic heterogeneity** occurs when two vocabularies are not expressed in the same syntax as the vocabulary language. This generally happens when two vocabularies compare, for instance, a classification with a conceptual model. This also happens when two vocabularies are modeled by using different knowledge representation formalisms, for instance, RDF, OWL, or SKOS. This kind of mismatch is generally tackled at the theoretical level by establishing equivalences between constructs of different languages. Thus, it is sometimes possible to translate vocabularies between different vocabulary languages while preserving the meaning

- **Lexical heterogeneity** occurs due to variations in label names (descriptor terms) when referring to the same entities in different vocabularies [15, 102]. This can be caused by the use of different natural languages, e.g., "mum" vs "mamma".

- **Semantic heterogeneity** occurs due to structure factors [15, 5]. In general, it occurs due to the use of different expressions for defining concepts and their related relationships, e.g. "Reading" is a city in "England" or "Reading" is one kind of activity. This conceptualization mismatch depends on modeled concepts. Finally, in the context of conceptual differences, we can identify three important reasons for these to hold, namely difference in coverage, difference in granularity and difference in perspective.

- **Pragmatic heterogeneity** is concerned with how entities are designed by vocabularies specialists. Indeed, entities which have exactly the same interpretation are often interpreted by specialists with regard to the context. One example is how they are ultimately used. This kind of heterogeneity is difficult for the computer to detect and even more difficult to solve, because it is out of its reach. The intended use of entities has a great impact on their interpretation, therefore, matching entities which are not meant to be used in the same context is often error-prone. Given the limited grasp that a computer can have on these issues, we do not deal with semiotic heterogeneity here.

- **Metadata heterogeneity** is concerned with how data is presented in the metadata registry, with entities which have the same names but different expressions [101]. This kind of heterogeneity problem occurs in bibliographic data expression. For example, in scientific papers author names are expressed in different styles: "A.Powel" or "Powel, Andry". There is no specific format for this.

There are several existing heterogeneity problems, for example: instance heterogeneity, multi-lingual heterogeneity, etc. We mainly focus on the conceptual problems between two CVs.

2.3 Problem details

The matching operation determines the alignment for a pair of controlled vocabularies (CVs) CV1 and CV2. The concept of CV matching is based on the hidden semantic matching idea described in [114, 8, 47]. The key intuition behind matching controlled vocabularies is the determination of mapping by computing syntactic [18] and semantic relations [16] which hold between the entities of any two given CVs. The matching task is the main focus of this thesis work.

Given two CVs, a corresponds is 4-tuple $\langle ID_{i,j}, c_i, d_j, R \rangle$, $i = 1, ..., N_C$; $j = 1, ..., N_D$ where $ID_{i,j}$, is a unique identifier of the given mapping element; c_i is the i-th node of the CV1, N_C is number of nodes in the CV1, d_j is the j-th node of the CV2, N_D is the number of nodes in the CV2 and R is a specific relation (e.g., exact match ($\equiv$), more general ($\sqsupseteq$), less general ($\sqsubseteq$), and not match ($\perp$) which may hold between the concepts at nodes c_i and d_j.

- Exact match: when two concepts are equivalent.
- More general match: when two parent concepts (BT are matched) are matched we call this a more general match.
- Less general match: when two children concepts (NT are matched) are matched we call this a less general.
- Not Matched: when two concepts are not matched.

CV1
CV2
Matcher
Output

Figure 2.1: CV Matching

Therefore, in light of the above discussion, CV matching defines the following problem: given two CVs T_C and T_D compute the $N_C \times N_D$ mapping element $ID_{i,j}$,c_i,d_j,R with $c_i \in T_C$, $i = 1, ..., N_C$, $d_j \in T_D$, $= 1, ..., N_D$ and R relation holding between *concepts at node* c_i, d_j. Since we look for the $N_C \times N_D$ correspondence, the cardinality of mapping between elements can be determined to be $1 : N$. If necessary, these can also be decomposed straightforwardly into mapping elements with 1:1 cardinality.

For example, take the two concepts "Agriculture Industry" from the AGROVOC and "Agriculture manufacture" from the CABI. According to some linguistic approaches [102, 101] their labels measure 0.3478 (from levenshteinDistance matcher). This matching algorithm uses a threshold of 0.5 as on indicator for the resulting matching, i.e., the algorithm considers all the pairs of entities with a confidence measure higher than 0.5 as correct correspondences. Thus our hypothetical matching algorithm should return the following correspondence:

$\langle$"$AgricultureIndustry$", "$Agriculturemanufacture$", $equal\rangle$

However, according to another matching algorithm, they may not be equivalent. These variations depend on relationships and different approaches to matching.

2.4 Summary

In this chapter, we have described different kinds of KOS models. They are dissimilar in structure and presentation languages. Then, we have shown the different heterogeneity factors which create problems. Finally, we have described our problems on the basis of the state-of-the-art.

Chapter 3

Aligning CVS using a facet based approach

The central notion behind the Semantic Web is its ability to uniquely identify resources (with URIs) and languages (e.g. RDF/S, OWL) to formally represent knowledge (i.e. ontologies, which can simplistically be considered the taxonomies of classes representing objects, and of their interrelationships) [84, 13]. These taxonomies contain domain knowledge; the domain is represented by a set of words and phrases used to describe concepts. A vocabulary is said to be controlled if it stores domain-specific chosen words, synonyms, word sense definitions (i.e. glosses) and relations between word senses and concepts [130]. In a Controlled Vocabulary (CV), we denote words as "blocks from which sentences are made", a synonym (it is binary relationship) of "a term is a word refers to the same concept of that term", a sense as "a meaning of a concept" and a concept as "an abstract idea inferred or derived from specific instances". The importance of CVs can hardly be underestimated; generally, each company or research group has its own information source, e.g., databases, schemas and structures. Each of these sources has its own set of individual CVs, creating a high level of heterogeneity. On one hand this is desirable, as it allows the involved parties to structure knowledge in a way which best fits their needs, e.g., for specific inter-office applications. On the other hand, individuals or companies also sometimes need a unified knowledge base (made up of different information sources) in order to satisfy their goals. This integration process requires a mapping between different CVs. Mapping between two CVs is generally a critical challenge for semantic interoperability [17]. These CVs are frequently used a lots as background knowledge for this data integration [44, 36]. What is more, classifications matched using CVs are lightweight ontologies, also called Formal Classifications (FC). In an FC, lexical labels are translated to logical labels that remove ambiguities of natural language. For further reading, we refer the reader to [51, 38]. In our case, we are interested in the correspondence between concepts from two CVs, e.g., concept-to-concept mapping which includes word-to-word mapping, or synonym-to-synonym mapping [63, 104]. This mapping cannot be accomplished solely by a lexical comparison of two concepts using element level matchers [35, 93, 18, 74, 62, 120, 115] such as those included in SMOADistance, HammingDistance, JaroMeasure, SubStringDistance, N-gram, JaroWinKlerMeasure, and LavesteinDistance; we also need to consider the existing semantics. In light of the above discussion, the objective of this work is to determine a fully-automated mapping between two CVs; this work may be useful for navigating vocabularies, information extraction and linking data. Our work is published in [8]. In this chapter, we described about the facet in the sense we use it. We also describe controlled vocabulary matching. Finally, we describe an algorithm for matching.

3.1 Faceted Controlled vocabulary

3.1.1 Facet

A facet is like a diamond that consists of different faces. Its distinct features allow thesauri, classifications or taxonomies to be organized in different ways. The facet is also clearly defined, mutually exclusive, and composed of collectively exhaustive aspects of properties or characteristics of a domain. For example, a collection of rice might be classified using cultural and seasonal facets.

A Facet is constructed according to the following two steps [44, 4]:

1. **Domain analysis :** Analysis of the term is done by consulting domain experts. This process is called the idea plane, the language independent conceptual level, where simple concepts are identified. Each identified concept is expressed in the verbal plane in a given language. For example, in English, we try to articulate the idea coextensively, namely by identifying a term which exactly and unambiguously expresses the concept.

2. **Term collections and organization :** Terms are collected and homogenous terms are ordered according to their characteristics, and (in hierarchies) in a meaningful sequence. The set of homogenous terms form a facet. For example, “cow” and “milk” form a facet called “Dairy System” (these entities have a *part-of* relationship to “Dairy System”).

The above steps construct a faceted knowledge organization system and correspond to background knowledge, namely the a priori knowledge which must exist in order to make semantics effective. Notice that the grouping of terms in step 2 has real world semantics, namely they are ontologies, clas-

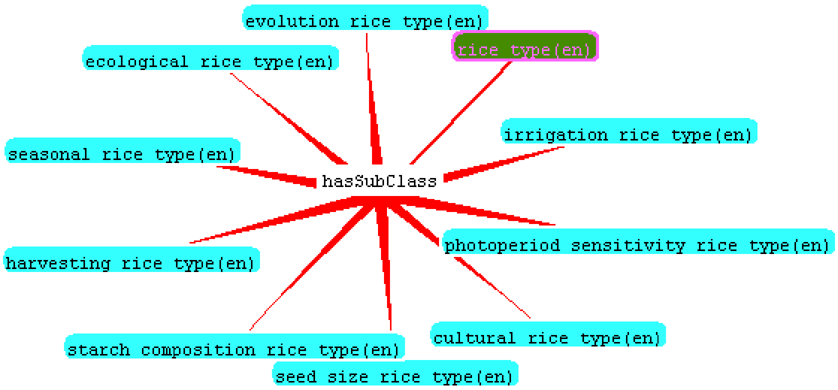


Figure 3.1: Rice Type

sification, and thesauri which are formed using *partOf*, *is-a*, *isSubClassOf*, and *instanceOf* relationships.

To properly identify a facet we need to consider the following:

Specific characteristics of a domain's topics can be seen as independent modularization of that domain. For instance, “dairy product” can be seen in “nutrition”.

S.R. Ranganathan [107, 108] was the first to present the notion “facet” in library and information sciences (LIS). He proposed five different aspect to consider for building a facet, denoted PMEST: Personality (P), Matter (M), Energy (E), Space (S) and Time (T). However, his student Bhattarcharyya [11] proposed a refinement which consist of four main categories, called DEPA: Discipline (D) (what we now call a domain), Entity (E), Property (P), and Action (A).

DEPA can be described as follows:

Discipline (Domain): this includes established fields of studies (e.g., Library Science, Mathematics, and Physics), applications of traditional pure disciplines (e.g., Engineering, and Agriculture), any aggregates of such fields (e.g., Physical Sciences and Social Sciences), as well as more modern terms, fields like music, sports, computer science, and so on.

Entity: the elementary category Entity is manifested in conceptual existence. Basically the concept represents the core idea of a domain treated as under this element category. For example: *rice is an entity or concept in the agriculture domain.*

Property: this includes both quantities and qualitative characteristics. For example, measure, weight, taste, etc.

Action: *every concept should be considered with the notion of "doing". It includes processes and steps of doing. An action can manifest itself as either "Self-action" or "External action". Self-action is an action done by some agent (explicitly or implicitly) on or by itself.* For example, imagination, interaction, reaction, reasoning, thinking, etc. An external action is an action done by some agent (explicitly or implicitly) to a concept of any of the elementary categories described above. For example, organization, cooperation, classification, cataloging, calculation, design, etc.

To build a concept facet [100, 15], we take a discipline and then an entity from the DEPA model. Other properties will not be considered in this case. This process can be called semantic factoring. For example, we choose the domain or discipline as Agricultural science. In this domain *rice* is an entity or concept. Different kind of rices exist in the world. Figure 3.2 shows [19] a distinct module of rice types divided into seasonal rice type, cultural rice type, seed size rice type, and so on. These types depend on cultural, size, seasonal and others factors, each of which can be considered a different facet.

Figure 3.2 shows one facet, the seasonal rice type. Seasonal rices are mostly cultivated in Asian countries like India, Bangladesh, Nepal, and

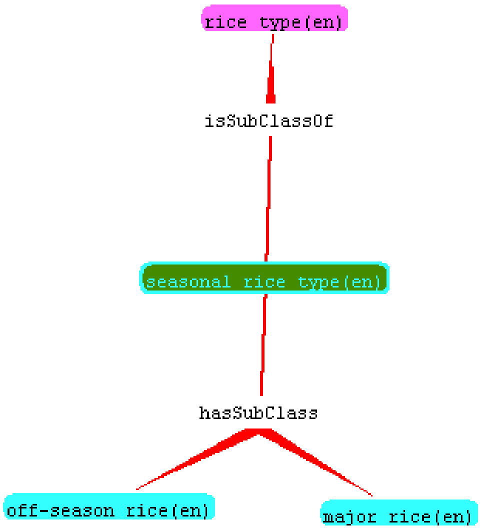


Figure 3.2: Seasonal Rice Type

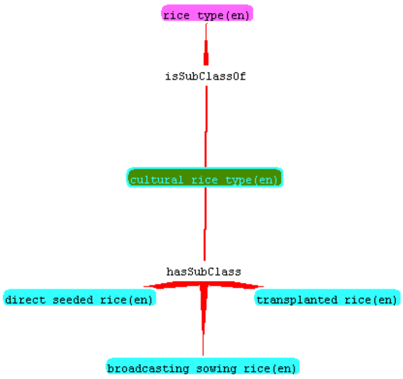


Figure 3.3: Cultural Rice Type

Pakistan. These kinds of rices are planted during the rainy season and cultivated after two or three months. Their cultivation is completely dependent on time. Figure 3.3 shows another facet, the cultural rice type. This class of rice is mostly cultivated in Thailand. Some seeds are planted once. The rice then grows from the seeds directly; this kind of rice is called direct seeded rice. On the other hand, some seeds planted two times. The first, in one place, is for growing a part of the seeds and the second, in the paddy, is for full growing; these kinds of rice are called transplant rice.

There are some common properties of facets:

Hospitalities new terms are added without any difficulties in the hierarchical structure. Terms with in the structure are clearly defined, mutually exclusive and collectively exhaustive.

Compactness facet based systems need less visualization than other hierarchical knowledge organization systems to classify the universe of knowledge. There is no explosion of the possible combinations as the basic elements (facets) are taken in isolation.

Flexibility traditional hierarchical knowledge organization systems are mostly unbending in their construction, whereas facet based systems are flexible by nature.

Reusability a facet based classification for a particular domain can be reused for developing other related domains.

The Methodology a strong methodology for the analysis and categorization of concepts with the existence of reliable rules for synthesis is provided.

Homogeneity this represents a set of concepts that must be homogenous according to their characteristics.

In some sense all properties are included in building concept facets. More precisely, homogeneity, the methodology and structure are more applicable in our case.

3.2 Controlled Vocabulary Matching

Our problem revolves around the concept of CV matching based on the semantic matching idea described in [47, 114, 128, 8]. The key intuition behind matching controlled vocabularies is the determination of mapping by computing syntactic and semantic relations which hold between the entities of any two given CVs [47, 124]. Let us consider matching 4-tuples $\langle ID_{i,j}, c_i, d_j, R \rangle$, $i = 1, ..., N_C$; $j = 1, ..., N_D$ where $ID_{i,j}$, is a unique identifier of the given mapped element; c_i is the i-th node of the CV1, N_C is number of nodes in the CV1, d_j is the j-th node of the CV2, N_D is the number of nodes in the CV2 and R specifies a semantic relation which may hold between the concepts at nodes c_i and d_j. Therefore, in light of the above discussion, CV matching is defined as the following problem: given two CVs T_C and T_D compute the $N_C \times N_D$ mapped element $ID_{i,j}$, c_i, d_j, R with $c_i \in T_C$, $i = 1, ..., N_C$, $d_j \in T_D$, $= 1, ..., N_D$ and R being the strongest semantic relation holding between *concepts at nodes* c_i and d_j. Since we are looking for the $N_C \times N_D$ correspondence, the cardinality of mapping between elements can be determined to be $1 : N$. If necessary, these can also be decomposed straightforwardly into mapping elements with 1:1 cardinality.

From Figure 1, we can find the relationship between cereal and food if we have a mapped vocabulary.

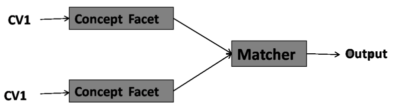


Figure 3.4: CV Matching

3.3 Concept Facet Matcher

A Concept Facet (CF) contains [8] distinct features for each concept: it includes combined relations, CF= $\langle lg, mg, R\rangle$, where lg identifies less general concepts (one or more), mg identifies more general concepts (one or more) and R identifies related concepts (one or more). In order to realize a matching between two vocabularies (CV1, CV2), we consider the CFs from all of the two CVs concepts: for every CF of CV1, we check for a match with all CFs of CV2. These concept facets are stored in tables for matching purposes. The methodology of the matching algorithm, applied to every concept, is represented by the following picture.

The matching between two concept facets follows the top-down approach and uses several lexical comparison algorithms [35, 93, 74, 18] (SMOADistance, HammingDistance, JaroMeasure, SubStringDistance, N-gram, JaroWinKlerMeasure, and LavesteinDistance). Firstly, we start by comparing the more general concepts; if they match (they have the same lexicalizations or they are synonyms) we assume that the concepts under investigation belongs to same concept (they match). Secondly (either we found a match or not), we start comparing the less general concepts. Based on the results of two mentioned matching, we may obtain an exact match (in case more general and less general concepts match), partial match (in case of only one match), or no match. Related concepts of CFs are used to validate previous results.

In short, we can express our CF matching algorithm in the following way:

Algorithm 1 buildCFacet(CV)

```
for $i = 0$ to $CV$ do
    store $cF \leftarrow (Mg,Lg,$R)
end for
return cF
```

In algorithm 1, we take each controlled vocabulary and store each concept information in cF. cF contains more general concepts (BT), less general concepts (NT) and related concepts (RT).

Algorithm 2 MatchingFacet(CV1,CV2)

```
cF1=BuildCFacet(CV1)
cF2=BuildCFacet(CV2)
for $i = 0$ to $cF1.size$ do
    for $j = 0$ to $cF2.size$ do
        cfmatcher=elementLevelMatcher($cF1$,cF2)
    end for
end for
```

In algorithm 2, we compare two concept facets using element level matchers and store all matching information in cfmatcher.

3.4 Summary

In this chapter, we have shown our proposed system for automatic vocabulary matching using concept facets. We are convinced that it helps provide better information searching, browsing, and extraction in agriculture and related domains. There are some open research issues: the semantic heterogeneity between two controlled vocabularies in a single domain; the multi-word concepts; the possibility of automatically linking non-matched concepts to external reliable resources such as public thesauri, encyclopedias or dictionaries.

Chapter 4

An Architecture and System for Aligning CVs

In this chapter, we illustrate the system architecture for aligning CVs and a human readable format to show and browse results so that users can understand the mapping and usage of the mapping in real life. Our target users are librarians and AGROVOC and CABI users. Some of them do not know much about semantic mapping, or the usage of the mapping. In this chapter, we describe the real time mapping prototype and the usage of mapping for searching agricultural and related documents.

4.1 Overview of System

Concept Facet Matcher (CFM) is an infrastructure for aligning controlled vocabularies and publishing them it in human readable format for use in semantic web applications, browsing agricultural information, and indexing documents. CFM is an automatic controlled vocabulary matching system that helps actualize interoperability between CVs.

Below, we describe in more detail the architecture of the overall mapping system, a running online prototype for showing mappings, and an architecture for semantic search to show the usages of the mapping files.

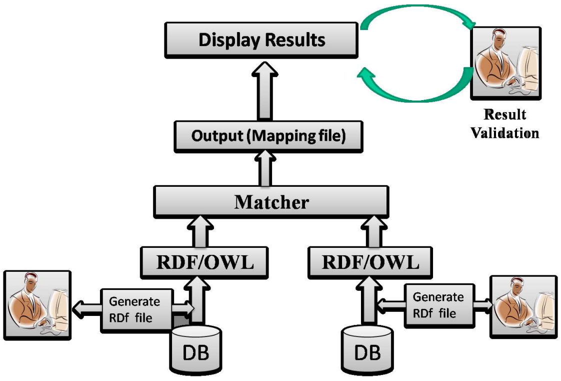


Figure 4.1: Overview of the system

4.1.1 Data formats

There are different types of data formats. We consider three data formats for our system.

1. Resource Description Framework (RDF)/ RDF Schemas (RDFs): RDF [42] is a triple organization model which resemble (only the shape of) semantic networks.

   (a) A fact is expressed as a triple of the form (*Subject, Predicate, Object*). It is like a short English sentence.

   (b) Subjects, predicates, and objects are names for entities, whether concrete or abstract, in the real world.

   (c) Names are in the format of URIs, which are opaque and global.

   The lacks of these network is the possibility to layer different levels of abstraction, by specifying classes of resources and by arranging these classes under a taxonomical relation. Another important feature lacking from RDF is the possibility to impose constraints over the applicability of properties.

RDF Schemas (RDFS) provide these features, leveraging RDF to a knowledge representation language with capabilities similar to semantic networks. Like in XML Schemas, RDFS Schemas are arranged in a modular way, which has been inherited from the adoption of an Object Oriented paradigm to knowledge representation. RDFS approach however differs from typical OO design. Rather than define (the intension of) classes in terms of the properties of its instances and then let objects be instantiated upon a given class according to its properties and the constraints which are bound on them (constrained approach), the RDF vocabulary description language describes properties in terms of which classes they can be applied to and let users declare objects (resources in RDF) of the domain without necessarily worrying which class(es) they belong to. This is an information which can be asserted in a later time, or which may have already been asserted somewhere else. This is in line with the nature of the Web, where information is distributed and potentially underspecified.

2. OWL: Web Ontology Language (OWL) [73] is a language for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web. OWL has been designed to meet the need for a web ontology language. OWL is one of the most important parts of the growing stack of W3C recommendations related to the Semantic Web. OWL increases the RDFS vocabulary with resources for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.

   OWL contains the following features:

Class A class defines a group of individuals that belong together because they share some properties. For example, *seasonal rice*, and *Harvesting rice* are both members of the class "rice". Classes can be organized in a taxonomic order or a hierarchical classification using *SubClassOf*. *Thing* is considered the super class in OWL.

rdfs:subClassOf Class hierarchies may be created by making one or more statements that a class is a subclass of another class. For example, the class "Rice" could be stated to be a *subClassOf* the class Crops. From this a reasoner can deduce that if an individual is a "rice", then it is also a "crops".

rdf:Property Properties can be used to state relationships between individuals or between individuals and data values. Examples of properties include *hasSubClass, hasColor, and hasPlace.*

rdfs:subPropertyOf Property hierarchies may be created by making one or more statements that a property is a *SubPropertyOf* one or more other properties.

rdfs:domain A domain of a property limits the individuals to which the property can be applied. If a property relates an individual to another individual, and the property has a class as one of its domains, then the individual must belong to the class.

rdfs:range The range of a property limits the individuals that the property may have as its value. If a property relates an individual to another individual, and the property has a class as its range, then the other individual must belong to the range class.

Individual Individuals are instances of classes, and properties may be used to relate one individual to another.

3. Databases: Each CV has its own database and different database

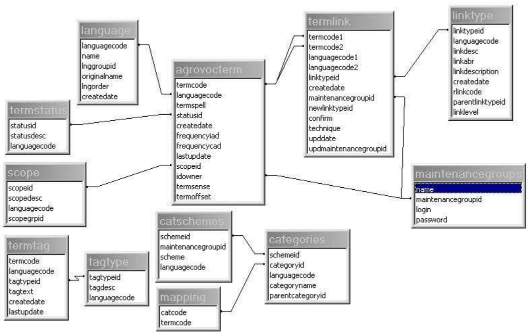


Figure 4.2: Agrovoc database structure

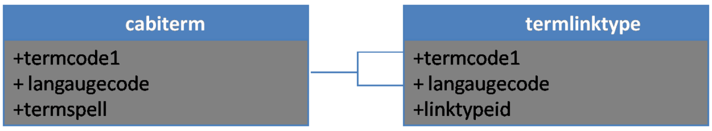


Figure 4.3: CABI database structure

format. For example, the AGROVOC and the CABI have their own databases. We obtained the existing database format for AGROVOC from the FAO web site and a text file from the CABI which we parsed to make it quite in similar format to AGROVOC.

In Figure 4.2 shows the AGROVOC database format[1]. We have considered two tables from the database: agrovocterm, and termlink.

CABI does not have multi-lingual facilities so we have to considered only English terms for our mapping purpose. Figure 4.3 shows the CABI database and here below, we have describe the tables in details.

4.1.2 Generation of RDF or OWL format by an Expert

An expert generates an owl file from a database [82] using Jena [41], which is a Java framework for building Semantic Web applications. It provides

[1] ftp://ftp.fao.org/

agrovocterm	
termcode	The Code assigned to the term. This code is the same for all languages
languagecode	the language code assigned to the term being described. References the “language” table.
statusid	The Status ID of the term such as “Deleted”, “Proposed”, “Terms with Relation”, “Non-descriptors with Relation”, etc. See the “termstatus” table
scopeid	The Scope ID of the term (Geographic or Taxonomic). See the “scope” table for details.
termspell	The lexicalization of the term in the specific language.
createdate	Date of creation of the term
lastupdate	When the term was last modified
idowner	Reference to the owner of the term (see table “maintenancegroups”)
frequencyiad	(old) No longer used
frequencycad	(old) No longer used
termsense	Reference for the refinement tool

a programmatic environment for RDF, RDFS, OWL, and SPARQL and includes a rule-based inference engine.

4.1.3 Matching System

Our matching system is based on element level matchers and consists of eight matchers [120, 115, 43].

- Hamming Distance measures the minimum number of substitutions required to change one string into the other, or the number of errors that transformed one string into the other. For example, “toned” and “roses” is 3.

termlink	
termcode1	The code assigned to the term related to another term (termcode2).
termcode2	Refers to the code of the term termcode1 is related to. Relationship could be BT, NT, RT, etc.
linktypeid	Represents the type of relationship between termcode1 and termcode2. Eg. 50 is BT, 60 is NT, etc. Refer to the "linktype" table for full details.
createdate	Date of creation of the term

cabiterm	
termcode	The Code assigned to the term
languagecode	Language code ; by default, English
termspell	The lexicalisation of the term.

- Levenshtein Distance is a metric for measuring the amount of difference between two sequences (i.e., an edit distance). For example, "kitten" and "sitting" is 3.

- JaroMeasure Distance is a measure of similarity between two strings. It mainly used in the area of record linkage (duplicate detection).

- NeedlemanWunch2 Distance is known by various names, Needleman-Wunch, Needleman-Wunch-Sellers, Sellers and the Improving Sellers algorithm. This is similar to the Levenshtein distance.

- SubString Distance measures the ratio of the longest common substring of two strings with respect to their length.

- N-gram Distance is a subsequence of n items from a given sequence. The items in question can be phonemes, syllables, letters, words or base pairs according to the application.

termlink	
termcode1	The code assigned to the term related to another term (termcode2).
termcode2	Refers to the code of the term termcode1 is related to. Relationship are BT, NT, RT, etc.
linktypeid	Represents the type of relationship between termcode1 and termcode2. Eg., 50 is BT, 60 is NT, etc.

- Smoa Distance is a function of their commonalities (in terms of substrings) as well as of their differences.
- JaroWinkler Distance is a measure of similarity between two strings. It uses for comparing the short strings such as person names. The score is normalized such that 0 equates to no similarity and 1 is an exact match

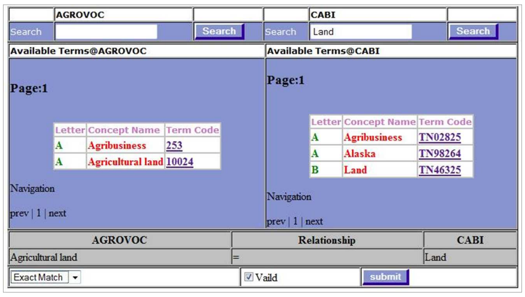
We describe these element level matchers in detail in Chapter 1.

4.1.4 Matching output

The matching output is in RDF and SQL formats. We use these matching results for display purposes.

4.2 Human Readable format for displaying the results

In this section, we present our implemented prototype for mapping and an architecture for a semantic search engine.



Search: Land

Available Terms@AGROVOC

Page:1

Letter	Concept Name	Term Code
A	Agribusiness	253
A	Agricultural land	10024

Navigation

prev | 1 | next

Available Terms@CABI

Page:1

Letter	Concept Name	Term Code
A	Agribusiness	TN02825
A	Alaska	TN98264
B	Land	TN46325

Navigation

prev | 1 | next

AGROVOC	Relationship	CABI
Agricultural land	=	Land

Exact Match ☑ Vaild

Figure 4.4: Human readable format for the mapping system

4.2.1 Search and showing mapping

In our mapping prototype, we can see two input boxes (Figure 4.4) for searching information from the AGROVOC and the CABI databases. For example, let us say a user wants to search "land" in the CABI database. After typing the search keyword "land", he presses the search button to see the results. He then sees "Agribusiness", "Alaska" and "land". By clicking the termcode of "land", he can see corresponding mapping concepts from the AGROVOC database, if they exist. In this example, he sees that "land" is mapped to "Agricultural land".

4.2.2 Validator

In the mapping system, a domain expert acts as a Validator. The Validator checks concepts and their corresponding relationships. If she thinks that concept-to-concept mapping is ok then she clicks on the "valid" checkbox and submits the information for storage in the database. If she thinks that a concept-to-concept mapping is not correct then she clicks in list box to the right mapping relationships and submits the information for storage.

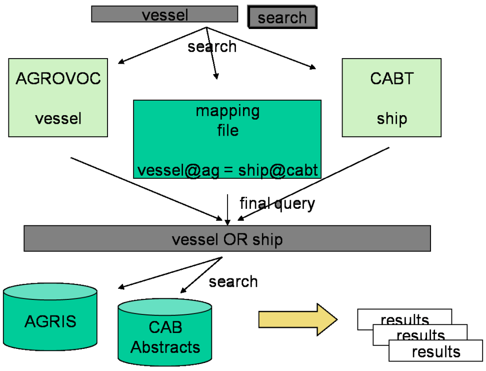


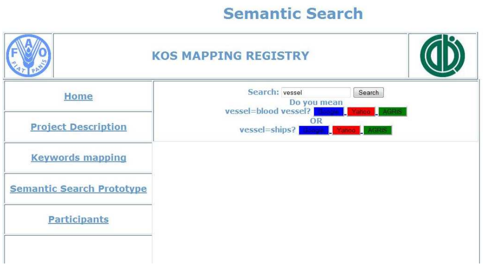
Figure 4.5: An architecture for Semantic Search

4.2.3 Usage of the Mapping File

One of the uses of CV mapping is semantic search. A semantic search is a process used to improve online searching by using mapped data from semantic networks to disambiguate queries in order to generate more relevant results. It not only allows for searching concept information but also gives information about relevant concepts. It also helps to get more information by integrating different data sources. The usage of mapping files can improve the recall of search without breaking constraints of semantic search. For example, say, there are two concepts “swine” and “pigs”, which are mapped in the mapping file. When a user searches for information on “swine”, she gets information about not only swine but also pigs. It makes searching more complete. We can use this information for cataloguing and document classification.

Figure 4.5 shows our semantic search prototype architecture. We have considered two databases and their existing mapping. If we want to make a query about “vessel” in our system, we first search the two databases and mapping files. In this case “vessel@ag” is mapped to “ship@cabt”. The query returns query “vessel OR ship”.

Otherwise, it gives suggestions to the user that perhaps he means “ves-



Semantic Search

KOS MAPPING REGISTRY

Search: vessel Search

Do you mean

vessel=blood vessel? Google Yahoo AGRIS

OR

vessel=ships? Google Yahoo AGRIS

Figure 4.6: Online prototype for semantic search using mapping files

sel"= "blood vessel" or "vessel"= "ship". We send the query to the AGRIS search engine (Agricultural related search engine), CABAbstract (Search Engine for CABI), Google and Yahoo in order to get documents about the given queries.

4.3 Summary

In this chapter, we have shown our proposed system for automatic vocabulary matching using concept facets. We have described data formats and the functionality of that system.

Chapter 5

Evaluation

The widespread diffusion of approaches for vocabulary matching shows the need for evaluation of these methods. Extensive experimental comparison of algorithms has been provided by the series of OAEI workshops and contest, and by similar initiatives, though very few works have dealt until now with comparison of real large-scale ontologies. Matching systems are difficult to compare, but we believe that thesauri matching or CV matching field can evolve only if evaluation criteria are provided [64, 65]. These should guide system architecture to access strengths and weaknesses of their systems as well as help application developers in choosing the most appropriate algorithms. In this chapter, we discuss in details evaluation procedure and apply criteria to some test cases.

5.1 Vocabulary

We have chosen two thesauri as our CVs. The thesauri used for this matching task are the United Nations Food and Agriculture Organization AGROVOC and Commonwealth Agricultural Bureaux International (CABI) thesauri. We selected these two thesauri because they are widely used and have not been completely mapped by anyone before.

5.1.1 AGROVOC

AGROVOC is a multilingual controlled vocabulary [21] designed to cover the terminology of all subject fields in agriculture and related domain (e.g. forestry, fisheries, food, etc). The AGROVOC thesaurus was developed by UN FAO and the Commission of the European Communities in the early 1980s. Since then it has been updated continuously by FAO and local institution in member countries. It is mainly used for indexing and retrieving data in agricultural information system both inside and outside FAO. It is updated by FAO roughly every three months. These work coordinate by ICRISAT[1].

Comparision		
Characteristic	AGROVOC	CAB
Tree leaves	29172	47805
Term count	18200	32884
Single words	6842	11720
MultiWords	11358	21161
Hierarchy depth	7	14
multiple BT	2546	1207
redundant BT	57	76

Table 5.1: Statistics of AGROVOC and CABI

There are several projects that use AGROVOC. Its website is aims[2]. There are some existing mappings [85] between AGROVOC and Chinese Agricultural thesaurus (manual), AGROVOC and the German National Library's Schlagwortnordatei (manual), AGROVOC and GAMET (automatic), and AGROVOC and NAL (automatic). AGROVOC is available in many different formats including ISO 2709 (format for bibliographic information interchange), SKOS, OWL and XML; all formats are generated

[1] http://www.icrisat.org/

[2] http://aims.fao.org/

from a native MySQL format. The current version of AGROVOC can be browsed online. It contains four types of relationships (BT, NT, RT and UF) [117].

Relationship	Broader terms	Narrow terms	Related terms	Used for
AGROVOC	228466	228424	326389	54370
CABI	15154	15841	41239	7094

Table 5.2: Relationship Comparison of the AGROVOC and the CABI

5.1.2 CABI

CABI (Commonwealth Agricultural Bureaux International) is a multilingual controlled vocabulary designed to cover the terminology of all subject fields in agriculture: forestry, horticulture, soil science, entomology, mycology, parasitology, veterinary medicine, nutrition and rural studies. The CABI thesaurus was developed by CABI which is a not-for-profit, science-based development and information organization. The CABI traces its origins back to 1910. It started as an entomological research committee and developed into a commonwealth organization before becoming a truly international service in agricultural information, pest identification and biological control. There are nine local CABI centers providing services in 70 countries. It is regularly updated. The current version was released in January, 2009. It covers all English terms as well as Spanish and Portuguese equivalents for most English terms. There are many current projects using the CABI thesaurus. It can be accessed online at www.cabi.org. The CABI thesaurus has four types of relationships (BT, NT, RT, and UF) derived from the ISO standard. We obtained data in text format and converted it to OWL and SQL formats for experimental purposes [118].

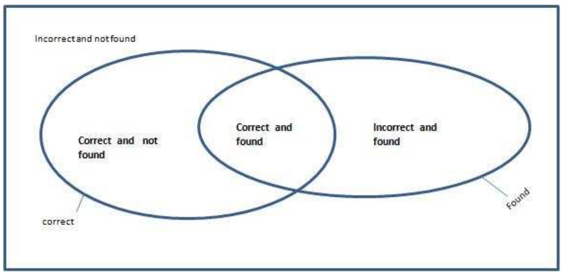


Figure 5.1: Measurement

5.2 Evaluation Measure

In order to evaluate the results of matching algorithms it is necessary to present them with CVs to be matched and to compare the alignment produced automatically with one produced by a domain expert. This section deals with the question of how to measure the results returned by vocabulary matchers. It considers different possible measures for evaluating matching algorithms and systems. These include both effectiveness and efficiency measures.

5.2.1 Quantity of measure

The best known mechanism that measure the performance of matching approaches lies in the calculation of precision and recall [114, 128, 125]. The mechanism originates in information retrieval and has been adapted to vocabulary matching and ontology matching. If we call the set of all alignment relations that are submitted by a participant "Found", and the set of all alignment relations we would like to receive (i.e., all correct alignment relations) "Correct", Precision and Recall can be defined as follows:

Precision=$\frac{|found \cap correct|}{|found|}$

Recall=$\frac{|found \cap correct|}{|correct|}$

Figure 5.1 (adapted from William's thesis [121]) illustrates the definitions. In practise, the computation of Precision and Recall require the assessment of all relations in the set of found relations and the determination of the cardinality of the set of all Correct relations.

The assessment of all found relations requires human assessors to decide whether tens of thousands of alignment relations are correct or incorrect. The experience of the OAEI has shown that a voluntary human assessor can judge around 250 alignment relations per hour for at most a few hours. This means that 10,000 alignments cost around 40 man-hours. For most large organizations that want to know the quality of an ontology alignment system, this is a feasible investment. For evaluation of such a matching task as the OAEI's this is not feasible. For the comparative evaluations of multiple systems we also have to assess multiple sets of found relationships.

Correct evaluation requires the human construction of the whole desired alignment by hand. Human construction of the entire alignment is even more costly than assessment of all found relations, because it involves searching for good alignment relations, which is more difficult than simply judging the validity of a set of given relations. To describe this situation we can look at the human construction of the alignment between Chinese Agriculture Thesaurus (CAT), which consists of 64,638 concepts and AGROVOC. This alignment is directional from CAT to AGROVOC (and hence incomplete) and consists of 24,686 alignment relationships. Chang Chung of the Chinese Academy of Agricultural Sciences (CAAS) revealed at the Eighth Agricultural Ontology Service (AOS) [77] meeting that the construction took 15 PhD students (in relevant fields like biology) 24 man-hours each over a six month period. The students were paid per alignment and follow a strict protocol. They constructed at most around 150 alignment relationships per hour.

If you are not interested in the evaluation as such, but in a complete alignment, automatic ontology alignment might not be necessary, because the total investment for the manual construction of an alignment is, for many purposes, not significantly larger than that of verifying an automatically constructed alignment. Provided that time, money and access to adequately educated people are not an issue, manual ontology alignment might be worth the investment.

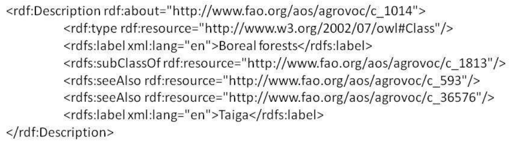
To make the computation of precision and recall feasible for our task, we performed sample evaluations. Sample evaluations assume that measurements on a randomly drawn sample can be extrapolated to the entire thesauri. The larger the sample, the less the estimation based on the sample will deviate from the true value on the entire thesauri. In our case, we extrapolated the performance of a system on a small set of alignment relations to all relevant alignments. We worked with small subsets of all Found and Correct relationships from which we generalized to the entire set of found or correct relations.

In order to draw samples from the set of all Correct alignment we have to draw from the set of all alignment relations and filter out the incorrect alignments. Clearly, some parts of the cartesian product of the sets of terms from two thesauri will contain more correct alignment relations than others. So in order to use our time optimally we looked for correct alignment relations in the areas that were more likely to contain such relationships.

5.3 Evaluation of Outcome using different methodology

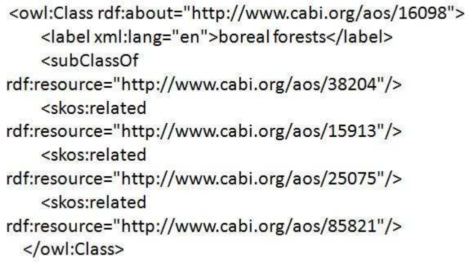
5.3.1 Facet based approach

We described our facet based [8, 4] approach for mapping in Chapter 4. We considered two concept facets for matching using string based methodologies.



```
<rdf:Description rdf:about="http://www.fao.org/aos/agrovoc/c_1014">
            <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
            <rdfs:label xml:lang="en">Boreal forests</rdfs:label>
            <rdfs:subClassOf rdf:resource="http://www.fao.org/aos/agrovoc/c_1813"/>
            <rdfs:seeAlso rdf:resource="http://www.fao.org/aos/agrovoc/c_593"/>
            <rdfs:seeAlso rdf:resource="http://www.fao.org/aos/agrovoc/c_36576"/>
            <rdfs:label xml:lang="en">Taiga</rdfs:label>
</rdf:Description>
```

Figure 5.2: Sample Input File of AGROVOC



```
 <owl:Class rdf:about="http://www.cabi.org/aos/16098">
      <label xml:lang="en">boreal forests</label>
      <subClassOf
rdf:resource="http://www.cabi.org/aos/38204"/>
      <skos:related
rdf:resource="http://www.cabi.org/aos/15913"/>
      <skos:related
rdf:resource="http://www.cabi.org/aos/25075"/>
      <skos:related
rdf:resource="http://www.cabi.org/aos/85821"/>
   </owl:Class>
```

Figure 5.3: Sample Input file of CABI

Experiment and Evaluation

To start our experiment, we used the following methodologies.

First we converted two files into OWL format.

Second we read these files using the semantic toolkit Jena and stored the concepts in triple storage. Figure 5.2 and 5.3 show the input files.

Third we considered only English language labels because our matching system only handles English labels.

Fourth , we obtained different results from the different matchers. We averaged the results from all the matchers. We used 0.56 as our given threshold.

Experiments were performed on a laptop with an Intel Core 2 Duo T5750 processor and 4GB RAM running Windows Vista using a 32-bit Java machine. It used only 3 GB RAM because a 32-bit OS architecture does not support 4GB RAM.

After running the experiments, we obtained the results displayed in table 5.3.

Parameter	Experiment 1	Experiment 2
Exact Match	5976	6021
Partial Match	164255	164278
No Match	69800745	69800732

Table 5.3: Facet based experiment

It was very difficult to evaluate the results. Output was produced in text format. Our domain expert did not feel comfortable evaluating the results in text format so we provided her in XLS format so that she could evaluate them quickly. For our evaluation, we built a system using RAP-API[3], PHP[4], MySQL, and Jena. We provided a mapping sample which contained 200 mappings to a domain expert at FAO, UN.

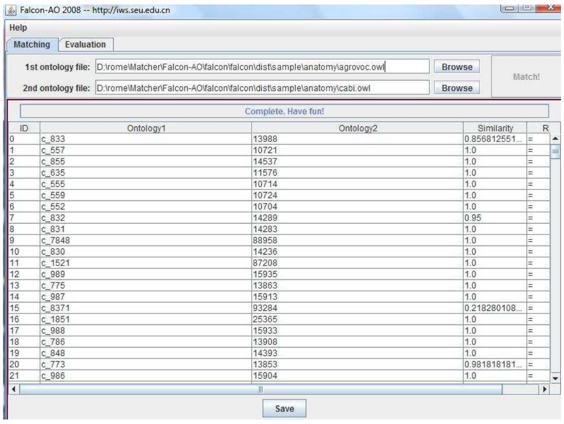
Our domain expert evaluated the results using the following criteria:

- a concept is exactly matched to another concept if their strings similarity is 1.0;
- a concept is partially matched if their strings similarity is less than 1.0;
- otherwise, concepts are not matched.

In our evaluation system, expert search for the concept and click on the concept so that she can see the corresponding mapping. If the expert thinks that the results are correct then she presses the submit button so that the information is stored in the database. Otherwise, the expert can correct the results using list boxes which contain information about partial matching, exact matching, and non-matching.

[3] http://sourceforge.net/projects/rdfapi-php/

[4] http://php.net/



1st ontology file: D:\rome\Matcher\Falcon-AO\falcon\falcon\dist\sample\anatomy\agrovoc.owl

2nd ontology file: D:\rome\Matcher\Falcon-AO\falcon\falcon\dist\sample\anatomy\cabi.owl

Complete. Have fun!

ID	Ontology1	Ontology2	Similarity	R
0	c_833	13988	0.856812551...	=
1	c_557	10721	1.0	=
2	c_855	14537	1.0	=
3	c_635	11576	1.0	=
4	c_555	10714	1.0	=
5	c_559	10724	1.0	=
6	c_552	10704	1.0	=
7	c_832	14289	0.95	=
8	c_831	14283	1.0	=
9	c_7848	88958	1.0	=
10	c_830	14236	1.0	=
11	c_1521	87208	1.0	=
12	c_989	15935	1.0	=
13	c_775	13863	1.0	=
14	c_987	15913	1.0	=
15	c_8371	93284	0.218280108...	=
16	c_1851	25365	1.0	=
17	c_988	15933	1.0	=
18	c_786	13908	1.0	=
19	c_848	14393	1.0	=
20	c_773	13853	0.981818181...	=
21	c_986	15904	1.0	=

Figure 5.4: Output of Falcon tool

5.3.2 Mapping using standard tool

In this experiment, we used FALCON-AO which is an automatic tool for aligning Ontologies. There are two matchers integrated into Falcon-AO: one matcher is based on linguistic matching techniques for ontologies, called LMO; the other matcher is based on graph matching techniques for ontologies, called GMO. In Falcon-AO, GMO takes the alignments generated by LMO as external input and outputs additional alignments. Reliable alignments are obtained through LMO as well as GMO. The matching reliability is obtained by observing the linguistic comparability and structural comparability of the two ontologies being compared. We chose Falcon-AO because it had given the best results according to OAEI's evaluation [125].

Experiment Setup and Evaluation

In order to do the experiment using FALCON-AO, both vocabularies had to be converted into RDF/OWL. Each concept became an $owl : CLASS$ and *broader/hypernym* relations were converted to $rdfs : subClassOf$ property statements. We considered only English concepts from AGROVOC in order to avoid multilingual problems.

For linguistic analysis, this tool used 0.9 as high similarity between two concepts and 0.035 as low similarity between two concepts. For structural similarity, it considered 0.95 as highest similarity between concepts and 0.5 the lowest similarity measure. It considered 0.0075 the threshold.

We initialized the path of input files and output files in the *"falcon.properties"* file. The experiments were performed on the same laptop.

After running the experiment, we got the results which are shown in table 5.4.

Parameter	Experiment1	Experiment 2
Exact Match	8795	8795
Partial Match	334255	334258
No Match	NA	NA

Table 5.4: Experiment Result

We found many ambiguities in our results. We evaluated our results using the same evaluation tool.

5.3.3 Using background knowledge

In our mapping experiment, we used the semantic matching system S-Match [114, 47] which implements the minimal semantic matching algorithm [45] developed by *the University of Trento.* Further information available at www.unitn.it. The semantic matching algorithm implemented in S-Match [128] consists of four steps.

First input sources in natural language are enriched with logical formulas using concepts drawn from a linguistic resource.

Second the formulas are contextualized to reflect the position of the concept in the initial data.

Third all atomic concepts identified in the source and target thesauri, are matched using background knowledge and other techniques, like string matching.

Fourth complex concepts from source and target thesauri are matched using a satisfiability solver and axioms collected in the third step. As a source of background knowledge for the first and the third steps we used WordNet, a generic linguistic resource and its extended version, made available by the Stanford WordNet project [89]. WordNet provides a good coverage of the general parts of the language and its slowly changing core. AN extended version of WordNet contains about 4 times more concepts than the original WordNet 2.1. For example, we extracted 78551 (WordNet:19,075) multiwords and 1271,588 (WordNet: 755,306) hypernym relations. The extended version is generated automatically and was 84 percent accurate.

Experiment Setup and Evaluation

We conducted two sets of experiments; Table 5.5 summarizes the parameters. We made the following variations during our experiments.

Input data for Smatch		
Parameter	1	2
AGROVOC version	2007-08-10	2007-08-10
CABI version	2009-11-01	2009-11-01
AGROVOC term-leaves	35036	35036
CABI term-leaves	29172	29172
Coversion	hierarchy	hierarchy
Knowledge base	WordNet 2.1	SWN 400.000
Matching Algorithm	Mini S-Match	Mini S-Match

Table 5.5: Experiments Parameters

First conversion from thesauri formats led to different results. The most important parameters that influenced the final results include: how to import relations, how to resolve ambiguities arising during the conversion process and which knowledge base to use. We imported only BT and NT relationships for establishing a hierarchy of concepts. During the import we found a number of terms which had multiple broader terms. Such concepts could be placed in two (or more) places in the final hierarchy. Instead of removing BT relationship until only one remains, we left these terms under their broader terms to increase matching chances.

Second we could preserve the hierarchy of terms using BT and NT relationships, or we can match term to term without considering the hierarchy.

Third we used different knowledge bases:WordNet version 2.1 and a 400.000 concept version of the Stanford WordNet Project.

Fourth we could choose between standard semantic matching and minimal semantic matching.

Fifth the input sources were changed for technical reasons. According to FAO experts, the structure and content of the 2009 version of AGROVOC is greatly improved a lot in comparison with 2007. However, the 2009 version was not available during the first experiment, so due to the amount significant changes it was decided to proceed with a new version.

The matching consisted of four steps: pre-processing (or concept at node computation), contextualization (or concept at node computation), element-level matching and structure-level matching. Below we will present some parameters and figures related to these stages of the matching process.

Table 5.6 summarizes the quantitative results of the preprocessing stage. Using general-purpose knowledge bases such as WordNet on domain-specific input resulted in a large amount of unrecognized words. For these words the matcher had to rely only on string-based matching techniques. Using extended WordNet from the Stanford WordNet Project results in slightly improved coverage. Differences in coverage also depended on the differences in thesauri versions and on the conversion parameters.

Parameter	1	2
Knowledge base	WordNet 2.1	SWN 400.000
Unrecognized words in AGROVOC	16080	14934
Unrecognized words in CABI	18235	16890

Table 5.6: Preprocessing stage figures

Table 5.7 summarizes the results of the experiments. Using the extended knowledge base in the element-level matching step increases mapping size. A relatively small number of equivalence relationships were noted in the first experiment. In the second experiment, where BT/NT relations were not used for conversion and only plain terms were matched, the number of discovered equivalence links is significantly larger. In the latter case the algorithm was able to establish an equivalence relation directly between two terms, while in the former cases it failed to establish the relation when intermediate terms were present in the hierarchy. We hypothesize that if the pairs of terms in question are the same, this could be due the lack of background knowledge. That is, in the former cases, a proper relation was not established between the intermediate terms, thus preventing the establishment of a relation between the end terms. Another possibility is that this is a consequence of using a minimal match algorithm [45]. Namely, the relation was established from one term to another, but either remained as a derived one found in the maximized mapping (unlikely, given that the amount of EQ in the maximized mapping is roughly the same), or again, lack of background knowledge prevented the establishment of a relation between intermediate terms, in turn preventing the establishment of a relation between the end terms. We report here both maximized and minimized mapping sizes due to their different purposes. The minimized mapping contains a sort of "compressed information", leaving out many links, (which could, however be derived). Therefore it is useful for exploration and validation as it minimizes the effort required. If used with applications, however, the consuming application should be aware of the semantics of minimal mapping. The maximized mapping has traditional semantics and is ready for immediate consumption by applications. The difference between minimized and maximized mapping sizes is as larger as a factor of 17 times.

Parameter				
	min	max	min	max
Mapping Size Relations	432475	5282852	4353322	5191637
EQ (equivalence)	3698	3564	3603	3468
DJ(disjointness)	125439	3811923	124648	3777493
MG (more general)	84759 204665	83931	173992	
LG (less general)	218579	1262700	223140	1236684

Table 5.7: Experiment Results

The experiments were conducted on a laptop with an Intel Core 2 Duo T9600 processor and 4G RAM running Windows 7 x64 using a 64-bit Java machine. The run times should be considered as approximate, because although S-Match currently runs single-threaded and there were 2 processors available with one available almost exclusively for the JVM, the matching process was not the only process in the OS and other (lightweight) activities were permitted during the experiments.

Evaluation is an ineluctable part of many experiments. In matching experiments, evaluation is not a simple task. For large matching tasks, such as this one, many of the more precise techniques based on a manual examination are not applicable due the size of the data.

Parameter				
	min	max	min	max
Overall	25.8065	31.4496	21.7391	21.7391
Positive	18.6047	14.0814	10.4895	14.6154
Negative	97.1831	52.1495	94.7368	99.1304

Table 5.8: Precision for minimized and maximized mapping

To evaluate the quality of links discovered by the matching algorithm, we needed a golden standard to compare the mapping to. Such a mapping is usually created by an expert in the domain of the resources being matched and not only requires significant effort, but in many cases is impossible to create. Expert time is a very valuable resource and there is but a little of it available. This limited us in choosing an evaluation method. We chose to evaluate a random sample of links from the mapping. We used a sample size of 200 links. In the following we assume that the mapping being evaluated contains links with 4 relations: $EQ(equivalence)$, $DJ(disjointness)$, $LG(lessgenerality)$, $MG(moregenerality)$. The part of the mapping consisting of EQ, LG and MG links is called the positive part. The rest, namely DJ links, is called the negative part. Traditionally, the most interesting part of the mapping is the positive part, with equivalences being the most desired links. However, one should consider the value of the mapping together with its intended use, keeping the target application in mind. For example, traditionally DJ relations are discarded as not being of interest. However, if the mapping is used for search purposes, DJ relations could be used to prune the search space and therefore shorten search times. Similar reasoning can be applied to less or more general links for narrowing or broadening search in a manner similar to the way BT/NT relations work.

5.3.4 Limitations

There were some limitations found during our experiments:

Structure Problem: AGROVOC and CABI had different structures. For example AGROVOC was in SQL format. On other hand, we received only a text file for CABI, which did not adequately cover all concepts. The provided CABI file did not contain chemical and scientific concepts.

Term Variants: In AGROVOC, we found "frog farms" which should have been "frog farming" because "frog farms" is used for "frog culture" and BT is "aquaculture". Also, we found the abbreviated term "Uht milk" (one kind of milk product) which should have been "UHT milk". There were some ambiguous term which had different meanings, for example "cutting" (i.e., slicing of bread or meat) or "cuttings" (i.e.,propagation material). Furthermore, there were some terms spells whose meaning is to difficult to capture, for example "1,1-dimethylpiperidinium", "1,2-dibromoethane", "2.4.4-T", "2.4.5-TP 2.4-D", "2.4 DES", "2.4 dinitrohenol". Similarly, CABI contained the term "4-H Clubs". These terms did make sense during any mapping experiments.

Domain expert: To evaluate our results, we were able to find one domain expert from FAO but we did not get any domain expert from CABI. The results may have been different if we had another domain expert.

Lack of consistency: Since the relationships in thesauri lack precise semantics, they are applied inconsistently, both creating ambiguity in the interpretation of the relationships and resulting in an overall internal structure that is irregulated and unpredictable.

Limited automated processing: Traditional thesauri are designed for indexing and query formulation by people and not for automated processing. The ambiguous semantics that characterizes many thesauri makes them unsuitable for automated processing.

5.4 Summary

In this chapter, we have described our test cases in detail. We ran these test cases and obtained some results which we have described in the chapter. Also, we have described the evaluation procedure and evaluated results. Further more, we have described the limitations which AGROVOC and CABI dictionaries imposed on the experimentation.

Chapter 6

Conclusion and Future Trends

6.1 Summary of the Chapters

In this dissertation, we have given a detailed account of the state-of-the art in ontology matching, vocabulary matching, and already ongoing matching projects. We proposed a novel approach to controlled vocabulary matching, called facet based matching, illustrated its technical details, and presented some evaluation. Specifically, the main findings of each chapter are summarized one by one in sequence. Finally, future trends in the matching field are outlined in this chapter.

We showed the basic problem of controlled vocabulary matching, possible solutions and outline for our work in the introduction chapter.

We showed that there are various existing roles of controlled vocabularies in semantic matching and ontology matching systems. Also, we pointed out that there are several applications and data formats of CVs. We briefly discussed modern CVs and traditional CVs. Furthermore, we provided a systematic view of matching techniques and methods (Chapter 1). In addition, we showed that there are several existing ongoing matching projects and evaluated results in Chapter 1. We showed that there are various existing ways of expressing knowledge found in diverse applications. These ways of expressing knowledge can be viewed as different forms of CVs that may need to be matched (Chapter 2). Unlike many other works, we aimed to treat the matching problem in a unified way and provide a common roof under the heading of ontology matching for many existing instantiations of this problem, such as schema matching, catalog matching, etc. In fact, schema matching is usually performed with the help of techniques aiming to retrieve the meaning encoded in schemas. On the other hand, ontology matching systems primarily try to exploit knowledge explicitly encoded in ontologies. In real world applications, schemas and ontologies usually have both well-defined and obscure terms, and well-defined and obscure contexts in which they occur; therefore, solutions to both problems would be beneficial. We introduced several justifications for heterogeneity in order to help the design of a matching strategy. Finally, we precisely defined the ontology matching problem. Having analyzed in detail the state-of-the art we proposed an approach to CV matching called facet based matching (Chapter 3). This was done based on what we found to be good practices in the previous approaches and what we found missing in them, thereby bridging that gap. We discussed with the help of examples and pseudo-code, the main steps of the algorithm that implements the facet based approach. We demonstrated how to deal, in a fully automated way and without background knowledge, with matching tasks using the facet based approach.

We demonstrated a complete system architecture of CV matching in Chapter 4.

We discussed some evaluation criteria for comparison of the results of matching algorithms (Chapter 5). We described our experience with building a large test case for the evaluation of quality results produced by matching systems. It is worth noting that this is a time-consuming and error-prone effort, however, and that having large real world data sets for evaluation of the quality of matching results is among the more important and under developed themes of CVs matching.

We performed an evaluation of the facet based matching approach, giving a proof of the concept that is practically useful. As our comparative evaluation shows it is very difficult to know a priori the quality one can expect in a matching system. Matching tasks are so different that a system can perform very well on some, usually small test cases, while not so well on others, usually large-scale test cases. Analysis of the mistakes made by a system points to a number of further possible improvements.

We would like to make two final notices. The first notice concerns some assumptions and limitations of the proposed solution. In particular, the proposed solution naturally assumes that the vocabularies to be matched have a meaningful overlap, that these are worth matching. The proposed approach reduces the conceptual heterogeneity only to a certain extent, though, for example, cases such as geometries axiomatized with points as primitive objects, and geometries axiomatized with spheres as primitive objects are not handled. Furthermore, although we have aimed at producing a generic matching solution, a lot of work still needs to be done.

The second notice is that although the semantic heterogeneity problem has been known and worked on for decades, vocabulary matching, which is a plausible solution to it. Therefore, besides the development of a facet matching approach, many efforts have been invested in understanding the relationship to vocabulary matching problems as well as in the rationalization of the state-of-the art.

6.2 Future Trends

There are several works that can be extended to the field of matching. We believe that these are important contributions to our work which will be carried out in the future.

1. To build the extended knowledge base (EKB)

A Knowledge base is a storage of knowledge which may be modelled as a controlled vocabulary, classification, schema, taxonomy, ontology, etc. At the moment, there is no universal knowledge base system to which we can turn. For the time being, WordNet is a working alternative. But it does not cover all the information needed in specific domain, for example, in the *Agriculture* or *Medicine* domains. On the other hand, it is not domain specific. One of its advantages is that we can extend it according to our needs. We strongly felt it necessary during our matching experiment to use the SMatch tool, which used WordNet as background knowledge. WordNet covers only 30 percent of the Agricultural domain. This problem can be solved by extending our knowledge base to use different thesauri within the same domain. Creating and managing new KBs will be time consuming and laborious work. However, from our experience we know that CABI, AGROVOC, ASFA [9], and NAL all cover more or less the same domain of information. CABI and AGROVOC overlap in 70 percent of their terms while AGROVOC and NAL overlap in 60 percent of their terms. Since ASFA is a sub-branch thesaurus of AGROVOC, we can say ASFA is a child of AGROVOC. It has a lot of similarity among the terms.

There are several issues that we need to take into account for building an EKB:

1. *Data format.* The first issue is a common data format. There is no unique ISO format for all thesauri. However, we can take the SKOS format as a starting point since present vocabularies are presented in SKOS format.

2. *Proposed EKB.* Figure 6.1 shows the diagram of a proposed EKB.

3. *Maintenance.* Another issue is maintenance of the EKB. FAO voluntarily manages the different “International Information System for

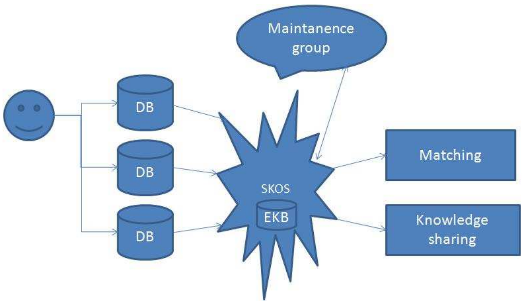


Figure 6.1: Proposed EKB

the Agricultural Sciences and Technology (Agris)" centers [6], that maintain AGROVOC. We can use the same ideology for maintaining the EKB by collaborating with different institutes.

Realization of a EKB will bring tremendous changes not only to the matching fields but also to the cataloging, classification and information navigation fields. It will increase the accuracy rates of all existing mapping systems. Furthermore, it will help distribute the knowledge among different organizations. Finally, it will have more impact on the research community.

2. Integrating Mapping into Modern Controlled Vocabulary (Concept Server)

Knowledge exchange and improving worldwide access to information in the agricultural domain by developing knowledge management resources, standards and tools is one of the main activities through which FAO aims to combat hunger and poverty in the world. One of the most important resources for covering the terminology of all subjects of interest to FAO (agriculture, forestry, fisheries, food and related domains, e.g., environment) is AGROVOC, the multilingual agricultural thesaurus, developed by FAO and the Commission of the European Communities in the early 1980s. Since then it has continuously been updated by FAO in collaboration with partner organizations in different countries, and is now available online in 19 languages.

In light of the rapid developments in information management and the possibilities available for exploiting semantic technologies, FAO has been working on converting the AGROVOC thesaurus to a concept server. The main objective of the AGROVOC concept server (CS) is to create a collaborative references platform and a one-stop shop for a pool of commonly used concepts related to agriculture terms, definitions, and relationship between terms in multiple languages and derived from various sources.

Consequently, the main characteristics [81] of the CS, compared to the traditional AGROVOC thesaurus are the following:

- It is a concept-based, modularized and extensible system.
- It gives the possibility to realize term and language specific relationships which offers for much more flexibility on the linguistic level.
- It allows for the representation of more semantics in terms of concept and term relationships and other constraints and definitions provided by the OWL modeling language.
- It caters to distributed maintance for improved workflows and better domain coverage.

Over the years, the initial idea of the agricultural ontology service (AOS) developed [21] into something much bigger. The agricultural ontology service, which the concept server is now an integral part of, also includes domain ontologies, registries of mappings, URN services to name but a few of its features. This service will host a wide variety of elements and services which are necessary to realize interoperability in the agricultural domain, and which will be made available to users in the international community for better harmonization of data and realization of better development tools.

The concept server [81] as described above represents the core of the AOS.

- A knowledge organization registry will be maintained in order to register trusted and well developed knowledge organization system (ontologies, thesauri, etc., whether based on the CS model or created using other models) within the agricultural community.
- A registry of mappings will be maintained through which mappings between featured KOSes will be made available for use in other systems for disambiguation, translation and other purposes.

Our thesis has been an investigation of the part related to uploading the KOS mapping into the CS.

3. Semantic Search using mapping files

Some of the important use of a mapping are making search queries faster, harvesting different information from heterogenous sources and presenting this information arranged according to its semantic meaning. We have proposed a prototype for semantic matching using AGROVOC and CABI mapping files (Figure 5.6 from Chapter 5). There are some existing mapping files (e.g., AGROVOC-NAL, AGROVOC-CAT) at FAO [85]. In the future, we will combine these files for better semantic search and navigation of agricultural information.

6.3 Conclusion

In this thesis, we have tried to solve the problem of vocabulary matching using a large number of datasets. We evaluated three matching techniques using these datasets. The majority of this work was done under the supervision of the FAO and the CABI. At the moment, a prototype is running at the FAO. Some work still needs to be done in order to fine-tune the system. In the future, we will integrate this system into the AGROVOC Concept Server.

Bibliography

[1] Project jxta. *see http://www.jxta.org.*

[2] Stanford peers project, see http://www-db.stanford.edu/peers/.

[3] T. Kamps A. Faatz and R. Steinmetz. Background knowledge,indexing and matching interdependencies of document management and ontology maintenance. In *Proceedings of the First Workshop on Ontology Learning(OL-2000) in conjunction with the 14th European Conference on Artificial Intelligence (ECAI 2000)*, Berlin, November 2000.

[4] M. Sini A. Morshed and J. Keizer. Aligning controlled vocabularies using a facet based approach. *Technical report at FAO, the Food and Agricultural Organization of UN(FAO),Rome,Italy*, November 2009.

[5] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *In Proceedings of the 16th International Conference on Computational Linguistics*, pages 16–22, 1996.

[6] AIMS. see http:aims.fao.org.

[7] A.Morshed. Controlled vocabulary matching in distributed system. *26th British National Conference on Databases,UK*, July 2009.

[8] A.Morshed and M.Sini. Creating and aligning controlled vocabularies. In *Advance Technology for Digital Libraries,AT4DL,Trento, Italy*, 2009.

[9] ASFA. see http://www4.fao.org/asfa/asfa.htm.

[10] F. Aubry and A. Todd-Pokropek. Mimos: a description framework for exchanging medical image processing results. In *Medinfo*, volume 10, pages 891–895, 2001.

[11] Bhattachary.G. Popsi:its foundamentals and procedure based on a general theory of subject indexing language. In *Libary Science with a slant to Documentation*, volume 16, pages 1–34, 1979.

[12] P. Bouquet, A. Dona', and L. Serafini. Contextualized local ontology spesification via ctxml. *Mean-02 - AAAI Workshop on Meaning Negotiation. Edmonton, Alberta, Canada*, 2002.

[13] P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: a new approach and an application. *In Proc. of the 2nd International Semantic Web Conference (ISWO'03). Sanibel Islands, Florida, USA*, October 2003.

[14] Karin Koogan Breitman, Carolina Howard Felicssimo, and Marco A. Casanova. Cato - a lightweight ontology alignment tool. In Orlando Belo, Johann Eder, Joo Falco e Cunha, and Oscar Pastor, editors, *CAiSE Short Paper Proceedings*, volume 161 of *CEUR Workshop Proceedings.* CEUR-WS.org, 2005.

[15] Paul. Buitelaar. *CoreLex:Systematic Polysemy and Underspecification.* Forest P.,U.S., Brandeis University, Waltham MA, USA, December 1998.

[16] D. Calvanese, G. De Giacomo, and M. Lenzerini. Description logics for information integration. In *Computational Logic: Logic Programming and Beyond*, Lecture Notes in Computer Science, pages 41–60. Springer, 2002.

[17] D. Calvanese, G. De Giacomo, and M. Lenzerini. A framework for ontology integration. In *The Emerging Semantic Web*. IOS press, 2002.

[18] William W. Cohen, P. Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the XVIII International Joint Conferences on Artificial Intelligence (IJCAI) - Workshop on Information Integration on the Web (IIWeb)*, pages 73–78, Acapulco, México, 9-10 August 2003.

[19] AGROVOC concept server. see http://naist.cpe.ku.ac.th/agrovoc/.

[20] C.Tatsiopoulos and B. Boutsinas. Ontology mapping based on association rule mining. In *11th International Conference on Enterprise Information*, Milan, Italy 2009.

[21] A. Liang F. Fisseha J. Keizer D. Soergel, B. Lauser and S. Katz. Reengineering thesauri for new applications: the agrovoc example. *Journal of Digital Information*, (4), 2004.

[22] D.Aumueller, H.-Hai Do, S.Massmann, and E.Rahm. Schema and ontology matching with coma++. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 906–908, New York, NY, USA, 2005. ACM.

[23] Flamenco demos. see http://flamenco.berkeley.edu/demos.html.

[24] Open directory project. http://www.dmoz.org/.

[25] D.Soegel. Indexing languages and thesauri:construction and maintence. MelVille Publishing Co., 1974.

[26] Marc Ehrig. Ontology alignment - bridging the semantic gap. 2006.

[27] E.McCulloch. Digital direction thesauri:practical guidance for construction. volume 54, 2005.

[28] Clusty:MetaSearch Engine. see http://clusty.com/.

[29] Google:Search engine. See http://google.com/.

[30] LookSmart:Search engine. See http://www.looksmart.com/.

[31] Swoogle:Semantic Search Engine. see http://swoogle.umbc.edu/.

[32] Vivisimo:MetaSearch Engine. see http://vivisimo.com/.

[33] Yahoo:Search engine. See http://yahoo.com/.

[34] EuroWordNet. see http://www.illc.uva.nl/EuroWordNet/.

[35] J. Euzenate and P. Shaviko. *Ontology Matching*. Springer, 1st edition, 2007.

[36] P. Shvaiko F. Giunchiglia and M. Yatskevich. Discovering missing background knowledge in onology matching. In *17th European Conference on Artificial Intelligance (ECAI 2006)*, volume 141, pages 382–386, 2006.

[37] F.Giunchiglia, I.Zaihrayeu, and U.Kharkevich. Formalizing the get-specific document classification algorithm. In *ECDL*, volume 4675 of *Lecture Notes in Computer Science*, pages 26–37. Springer, 2007.

[38] M.Marchese F.Giunchiglia and I.Zaihrayeu. Encoding classifications into lightweight ontologies. *Data Semantics VIII*, pages 57–81, 2007.

[39] F.Ibekwe-SanJuan. Construction and maintaining knowledge organization tools a symbolic appraoch. volume 62, 2006.

[40] Flickr. see http://www.flickr.com/.

[41] A Semantic Web Framework for Java. http://jena.sourceforge.net/.

[42] Resource Description Framework. see http://www.w3.org/RDF/.

[43] The free encyclopedia. See http://en.wikipedia.org/wiki.

[44] F. Giunchiglia, B.Dutta, and V.Maltese. Faceted lightweight ontologies. In *LNCS*, 2009.

[45] F. Giunchiglia, V. Maltese, and A. Autayeu. Computing minimal mappings. Technical report, DISI, University of Trento, 2008.

[46] F. Giunchiglia and P. Shvaiko. Semantic matching. *"Ontologies and Distributed Systems" workshop, IJCAI*, 2003.

[47] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: An algorithm and an implementation of semantic matching. *In Proceedings of ESWS'04*, 2004.

[48] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic schema matching. In *CoopIS*, 2005.

[49] F. Giunchiglia and M. Yatskevich. Element level semantic matching. In *Meaning Coordination and Negotiation workshop, ISWC*, 2004.

[50] F. Giunchiglia, M. Yatskevich, and E. Giunchiglia. Efficient semantic matching. In *ESWC*, 2005.

[51] F. Giunchiglia and I. Zaihrayeu. Lighweight ontologies. *Technical report at DIT, the University of Trento, Italy*, October 2007.

[52] F. Giunchiglia, I. Zaihrayeu, and M.Marchese. Towards a theory of formal classification. *Proceedings of the AAAI-05 Workshop on Contexts and Ontologies*, 2005.

[53] T.R Gruber. A translation approach to portable ontology specification. *Knowledge.Acquis*, 5(2):199–220, 1993.

[54] Nicola Guarino. Helping people (and machines) understanding each other: The role of formal ontology. In *CoopIS/DOA/ODBASE (1)*, page 599, 2004.

[55] A. Halevy, Z. Ives, I. Tatarinov, and P. Mork. Piazza: Data management infrastructure for semantic-web applications. *Proceedings of the International World-Wide Web Conference, WWW-03*, 2003.

[56] H.Corey and B.Tillett. Library of congress controlled vocabularies and their application to the semantic web.

[57] H.Zhu and S. Madnick. A lightweight ontology approach to scalable interoperability. *Working paper CISL, The Massachusetts Institute of Technology,Cambridge, MA ,USA*, June 2006.

[58] Gong Cheng ingsheng Jian, Wei Hu and Yuzhong Qu. Falcon-ao: Aligning ontologies with falcon. In *In Proceedings of the International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT)*, pages 85–91, 2005.

[59] P. Shvaiko J. Euzenat. Ten challenges for ontology matching. In *In Proceedings of The 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, 2008.

[60] P. Bernstein J. Madhavan and E. Rahm. Generic schema matching using cupid. In *In Proceedings of the International Conference on Very Large Data Bases (VLDB)*, page 4858, 2001.

[61] A.Gilchrist J.Aitchison and Bawden. Thesaurus construction and use:a practical manual. 4th ed., page 240, London, 2006. Aslib.

[62] M. A. Jaro. Advances in record linking methodology as applied to the 1985 census of tampa florida. *Journal of the American Statistical Society*, 64:1183–1210, 1989.

[63] Mustafa Jarrar. Towards the notion of gloss, and the adoption of linguistic resources in formal ontology engineering. In ACM, editor, *Proceeding of the 15th International World Wide Web Conference, WWW2006.*, Edinburgh, Scotland, May 2006.

[64] J.Euzenat. An api for ontology alignment. volume 3298, pages 698–712, Hiroshima, Japan,, 2004. The Semantic Web - ISWC 2004: Third International Semantic Web Conference, Springer. November 7-11.

[65] J.Euzenat. Semantic precision and recall for ontology alignment evaluation. pages 348–353, Hyderabad, India, January 6-12,, 2007. IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence. January 6-12.

[66] Bangyong Liang Jie Tang and Juanzi Li. Multiple strategies detection in ontology mapping. In *The 14th International World Wide Web Conference*, 2005.

[67] A. Valarakos K. Konstantinos and G. Vouros. Automs: Automated ontology mapping through synthesis of methods. In *International workshop on ontology matching*, Georgia,, USA, November 2006.

[68] A. Kementsietsidis, M. Arenas, and R. Miller. Data mapping in peer-to-peer systems. *ICDE*, 2003.

[69] A. Kementsietsidis, M. Arenas, and R. J. Miller. Mapping data in peer-to-peer systems: Semantics and algorithmic issues. *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, June 2003.

[70] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 170–178, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[71] S. Sceffer L. Serafini, S. Zanobini and P. Bouquet:. Matching hierarchical classifications with attributes. In *ESWC*, pages 4–18, 2006.

[72] Extensible Markup Language. see http://www.w3.org/XML/.

[73] Web Ontology Language. see http://www.w3.org/TR/owl-features/.

[74] I. V Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8):707–710, 1996.

[75] Wen-Syan Li and Chris Clifton. Semantic integration in heterogeneous databases using neural networks. In *In Proceedings of the Very Large Data Bases Conference (VLDB)*, page 4984, 1994.

[76] Wen-Syan Li and Chris Clifton. Semint: a tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data and Knowledge Engineering*, 33(1):4984, 2002.

[77] Sini M. Chang C. Li S. Lu W. He C. Liang, A. and J. Keizer. The mapping schema from chinese agricultural thesaurus to agrovoc. In Proceedings of the fifth Conference of the European Federation for Information Technology in Agriculture, Food and Environment and

the thirdWorld Congress on Computers in Agriculture and Natural Resources, 2005.

[78] National Agriculture Library. see http://agclass.nal.usda.gov/.

[79] Library Congress Author List. see http://www.loc.gov/,.

[80] Giorgio Terracina Luigi Palopoli and Domenico Ursino. Dike: a system supporting the semi-automatic construction of cooperative information systems from heterogeneous databases. *Softwarepractice and experinece*, 33(9):847884, 2003.

[81] G. Salokhe J. Keizer M. Sini, B. Lauser and S. Katz. The agrovoc concept server: rationale, goals and usage. Emerald Group Publishing Limited, 2008.

[82] A.Th. Schreiber J. Wielemaker M. van Assem, M.R. Menken and B. Wielinga. a method for converting thesauri to rdf/owl. *In Proc. of the ThirdInternational Semantic Web Conference,*, 2004.

[83] Anu Joseph Macgregor, George and Dennis Nicholson. A skos core approach to implementing an m2m terminology mapping server. In *the international conference on semantic web and digital libraries*, 2007.

[84] Bernardo Magnini, Luciano Serafini, and Manuela Speranza. Making explicit the semantics hidden in schema models. *In: Proceedings of the Workshop on Human Language Technology for the Semantic Web and Web Services, held at ISWC-2003, Sanibel Island, Florida*, October 2003.

[85] KOS/Ontology mapping. see http://aims.fao.org/.

[86] Google Maps. see http://maps.google.com/.

[87] R. Matthew, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *Proceedings of the Second International Semantic Web Conference*, Sanibel Island, Florida., October 2003.

[88] D. L. McGuinness, P. Shvaiko, F. Giunchiglia, and P. Pinheiro da Silva. Towards explaining semantic matching. In *International Workshop on Description Logics at KR'04*, 2004.

[89] George Miller. *WordNet: An electronic Lexical Database*. MIT Press, 1998.

[90] A. Dawson N. Dennis and A. Shiri. Hilt: A pilot terminology mapping service with a ddc spine. In *Cataloging and Classification Quarterly*, number 3, page 8, 2006.

[91] I. Niles and A. Pease. Towards a standard upper ontology. In *In Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS)*, page 29, 2001.

[92] Natalya Noy and Mark Musen. Anchor-prompt: Using non-local context for semantic matching. In *In Proceedings of the Workshop on Ontology and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*, page 6370, 2001.

[93] Natalya F. Noy. Semantic integration: a survey of ontology-based approaches. *SIGMOD Rec.*, 33(4):65–70, 2004.

[94] Library of Congress Classification system. see http://www.loc.gov/.

[95] Library of Congress Online Catalog. see http://catalog.loc.gov/.

[96] The Library of Congress.Thesaurus for the Global Legal Information Network(GLIN). http://www.loc.gov/lexico/servlet/lexico/.

[97] National Institute of Health National Center for Biotechnology Information. National library of medicine.

[98] Karlsruhe Ontology and Semantic Web Framework. see http://kaon.semanticweb.org/.

[99] L. Serafini P. Bouquet, B. Magnini and S. Zanobini. A sat-based algorithm for context matching. In *In Proceedings of the International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT)*, page 6679, 2003.

[100] P. Haase P. Buitelaar, P. Cimiano and M. Sintek. Towards linguistically grounded ontologies. In *6th Annual European Semantic Web Conference (ESWC2009)*, pages 111–125, June 2009.

[101] M. Teresa Pazienza, S. Sguera, and A. Stellato. Let's talk about our "being": A linguistic-based ontology framework for coordinating agents. *Applied Ontology, special issue on Formal Ontologies for Communicating Agents*, 2(3-4):305–332, 2007.

[102] Maria Teresa Pazienza and Armando Stellato. An environment for semi-automatic annotation of ontological knowledge with linguistic content. volume 4011, pages 442–456. The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings. Lecture Notes in Computer Science, Springer, 2006.

[103] Gio Wiederhold Prasenjit Mitra and Martin Kersten. A graphoriented model for articulation of ontology interdependencies. In *1In Proccedings of the International Conference on Extending Database Technology (EDBT)*, page 86100, 2000.

[104] L. Prevot, S. Borgo, and A. Oltramari. Interfacing ontologies and lexical resources. Jeju Island, South Korea,, 2005. OntoLex2005 - Ontologies and Lexical Resources.

[105] The United Nations Standard Products and Services Code(UNSPSC). See http://www.unspsc.org/.

[106] Open Video Project. see http://www.open-video.org/.

[107] S.R Ranganathan. Element of library classification. Asia Publishing house.

[108] S.R Ranganathan. Prolegomena to library classification. Asia Publishing house.

[109] HILT Final Report. see http://hilt.cdlr.strath.ac.uk/hilt3web/finalreport.html.

[110] E.Miller R.Guha, R.McCool. Semantic search.

[111] E. Rahm S. Melnik and P. Bernstein. Rondo: A programming platform for model management. In *In Proceedings of the International Conference on Management of Data (SIGMOD)*, page 193204, 2003.

[112] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[113] Hector Garcia-Molina Sergey Melnik and Erhard Rahm. Similarity flooding: a versatile graph matching algorithm. In *In Proceedings of the International Conference on Data Engineering (ICDE)*, page 117128, 2002.

[114] Pavel Shvaiko. Iterative schema-based semantic matching(phd thesis). *Technical Report DIT-06-102*, December 2006.

[115] Stamou G. Stoilos, G. and S Kollias. A string metric for ontology alignment. In *In Proceedings of the 4rd International Semantic Web Conference*, volume 3729, page 624637, Galway, Ireland, November 2005.

[116] MeSH: the National Library of Medicine's controlled vocabulary thesaurus. see http://www.nlm.nih.gov/mesh/.

[117] Agrovoc thesaurus. see http://www.fao.org/agrovoc/.

[118] CAB thesaurus. see http://www.cabi.org/.

[119] R. Fikes V. K. Chaudhri, A. Farquhar, D. Peter Karp, and James Rice. OKBC: A programmatic foundation for knowledge base interoperability. In *AAAI/IAAI*, pages 600–607, Madison, WI, July 1998.

[120] P. Rosso V. Mascardi, A. Locoro. Automatic ontology matching via upper ontologies: A systematic evaluation. In *IEEE Transactions on Knowledge and Data Engineering*, June 2009.

[121] Willem Robert van Hage. Evaluating ontology-alignment techniques. *PhD thesis, Vrije Universiteit,Netherland*, 2008.

[122] Piek Vossen. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks.* Kluwer Academic Publishers, Dordrecht,, 1998.

[123] H. Wache, V. ogele, T. Visser, U. Stuckenschmidt, H. Schuster, G. Neumann, and H. ubner. Ontology-based integration of information - a survey of existing approaches, 2001.

[124] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[125] L.Finch H. Kolb W.Hage, M.Sini and G.Schreiber. The oaei food task:an analysis of a thesaurus alignment task.

[126] W.Hoschek. A unified peer-to-peer database framework for xqueries over dynamic dystributed content and its application for scallable service discovery. *Disseration, CERN*, 2002.

[127] Rudolf Wille. Conceptual graphs and formal concept analysis. In *International Conference on Conceptual Structures*, pages 290–303, 1997.

[128] Mikalai Yatskevich. Schema-based semantic matching: Algorithms, a system and a testing methodology(phd thesis). *Technical Reports - DIT-05-047*, March 2008.

[129] YouTube. see http://www.youtube.com/.

[130] I. Zaihrayeu, L. Sun, F. Giunchiglia, W. Pan, Q. Ju, M. Chi, and X. Huang. From web directories to ontologies: Natural language processing challenges. In *ISWC/ASWC*, 2007.

[131] Ilya Zaihrayeu. Query answering in peer-to-peer database networks. *Technical Report DIT-03-012*, March 2003.