Context-Sensitive Ontology Matching in Electronic Business

Author Name
Jingshan Huang
School of Computer and Information Sciences
University of South Alabama, Mobile, AL 36688, U.S.A.
huang@usouthal.edu

Jiangbo Dang
Siemens Corporation, Princeton, NJ 08540, U.S.A.
jiangbo.dang@siemens.com

ABSTRACT
In today's global economy, electronic business has offered great advantages to enhance the capabilities of traditional businesses. In order to satisfy the imposed requirement for businesses to coordinate with each other, electronic business partners are chosen to be represented by service agents. These agents need to understand each others' service descriptions before successful coordination happens. Ontologies developed by service providers to describe their service can render help in this regard. Unfortunately, due to the heterogeneity implicit in independently designed ontologies, distributed agents are bound to face semantic mismatches and/or misunderstandings. We introduce an innovative algorithm, Context-Sensitive Matching, to reconcile heterogeneous ontologies. Our algorithm takes into consideration contextual information, via inference through a formal, robust statistical model based on confidence interval. In addition, an Artificial Neural Network is utilized to learning weights for different semantic aspects. At last, an agglomerative clustering algorithm is adopted to generate the final matching results.

INTRODUCTION
A wide range of online business activities for products and/or services are referred to as electronic business, which is increasingly being utilized by many different types of enterprises due to its potential to provide new opportunities and unparalleled efficiencies. In most cases electronic business is associated with buying and selling over the Internet, or conducting transactions involving the transfer of ownership or rights to use goods or services through a computer-mediated network. Broadly speaking, electronic business can be regarded as any business process that relies on an automated information system, which typically incorporates Web-based technologies. Thus, electronic business enables companies to link their internal and external data processing systems in a more efficient, effective, and flexible way. As a result, such companies will be more agile and responsive to their customers.

Because of the fact that electronic business is usually conducted using the dynamic environment of the Internet and the World-Wide Web, it is advantageous to introduce software agents into the electronic business area. Considering the two fundamental characteristics of software agents, i.e.,

autonomy and proactiveness, if services or business partners are represented by agents, it might enable us to increase the extent to which the data process is automated.

Previous research has found that exposing formerly internal activities to external business collaborators can yield increased value. Undoubtedly, there is value in accessing the service provided by a single agent through a semantically well-founded interface; at the same time, greater value is bound to be derived through enabling a flexible composition of electronic businesses, which not only creates new services, but also potentially adds value to existing ones (Singh, & Huhns, 2005). Before the communication and integration of electronic business activities can possibly happen, mutual understanding of semantics for interacting services is an indispensible precondition during such coordination process.

Ontologies serve as a declarative model for the knowledge and capabilities possessed by an agent or of interest to an agent. Not only are ontologies a core technology in the Semantic Web and Web 2.0, but they have also get deeply woven into the modern business world, as indicated by the vast amount of research in Enterprise Engineering and Enterprise Modeling. In essence, ontologies form the foundation upon which machine-understandable service descriptions can possibly be obtained and, as a result, automatic coordination among agents is then made possible. By providing a more comprehensible and formal semantics, the use of and reference to ontologies can help the functionalities and behaviors of agents to be formally and explicitly described, advertised, discovered, and composed. Eventually, each pair of ontology-conforming agents would be able to interoperate, even though it has not been specifically designed to do so.

However, because it is impractical to force all agents to adopt a global, "all-in-one" ontology that describes every concept that is or might be included as part of the services represented by these agents, ontologies from different agents typically have heterogeneous semantics. Due to this inherent characteristic, it is unavoidable for agents to reconcile their individual ontologies and form a mutual understanding before they interact with each other. Only via this means will agents be able to comprehend and/or integrate the information from different sources, and enhance process interoperability thereafter. In other words, during ontology management, of which the matching among heterogeneous ontologies is one of the most critical components, ontologies should be made dynamic, i.e., they should be associated with a certain degree of context-awareness. This being said, clues drawn from context should be taken into consideration during ontology reconciliation, if a more accurate, meaningful matching result is expected. In this chapter, we present an innovative algorithm, Context-Sensitive Matching, to reconcile ontologies from heterogeneous sources.

BACKGROUND

We give a brief review of the state-of-the-art ontology-matching techniques; in particular, we analyze the pros and cons of the existing two categories of matching algorithms: rule-based and learning-based algorithms. In addition, we also present an overview of current research in ontology and context, confidence interval applications, and ontology-based e-services.

Ontology Matching

Ontology matching is the process of determining correspondences between concepts from heterogeneous ontologies. Such correspondences include many relationships, for example, *equivalentWith*, *subClassOf*, *superClassOf*, and *siblings*. According to the classification in (Doan, & Halevy, 2005), most ontology-schema-matching techniques (Euzenat, & Shvaiko, 2007) can be divided into two categories: rule-based approaches and learning-based approaches.

Rule-Based Schema Matching

The rule-based solutions consider schema information only. Different algorithms have different methods of specifying a set of rules (usually domain-independent, although could be designed to include domain features); then these rules are applied to the available schema information, such as concept names, properties, data types, relationships, and other constraints, etc, to match schemas of interest. Different algorithms distinguish from each other by using different specific rules. However, they usually have the same advantage of relatively fast running speed. Also, they share the same disadvantage of ignoring the additional information possibly brought by instance data associated with schemas, when these instance data are available.

In (Noy, & Musen, 2000), Noy, N.F. and Musen, M.A. describe PROMPT, a semiautomatic approach to ontology alignment. By performing some tasks automatically and guiding the user in performing other tasks for which intervention is required, PROMPT helps in understanding ontologies covering overlapping domains.

Castano, S., Ferrara, A., and Montanelli, S. present H-MATCH in (Castano, Ferrara, & Montanelli, 2003). The authors divide the semantics of a concept into its linguistic and contextual parts. The former captures the meaning of terms used as concept names, while the latter evaluates the semantic affinity between two concepts by taking into account the affinity between their contexts, which are concept properties and relationships.

In (Dou, McDermott, & Qi, 2003), Dou, D., McDermott, D., and Qi, P. view ontology translation as ontology merging and automated reasoning, which are in turn implemented through a set of axioms. They obtain the merger of two related ontologies by taking the union of the terms and the axioms defining them, then adding bridging axioms through the terms in the merge. The language used in this approach, Web-PDDL, has the right degree of flexibility.

Similarity Flooding (SF) (Melnik, Garcia-Molina, & Rahm, 2002) is a matching algorithm based on a fixpoint computation that is usable across different scenarios. SF takes two graphs as input, and produces as output a mapping between corresponding nodes. This work defines several filtering strategies for pruning the immediate result of the fixpoint computation.

Cupid (Madhavan, Bernstein, & Rahm, 2001) is an algorithm for generic schema matching outside of any particular data model or application. It discovers mappings between schema elements based on their names, data types, constraints, and schema structure. Cupid has a bias toward leaf structures where much of the schema content resides.

S-Match (Giunchiglia, Shvaiko, & Yatskevich, 2005; Giunchiglia, Shvaiko, & Yatskevich, 2009) views match as an operator that takes two graph-like structures and produces a mapping between the nodes of the graphs. Mappings are discovered by computing semantic relations, which are determined by analyzing the meaning that is codified in the elements and the structures of the schemas. (Giunchiglia, Yatskevich, & McNeill, 2007) presents structure preserving match, which preserves a set of structural properties of the graphs being matched. An approximate structure matching algorithm is described, based on a formal theory of abstraction, and built upon tree edit distance measures.

Hu, B., Dasmahapatra, S., and Lewis, P. (Hu, Dasmahapatra, & Lewis, 2007) explore the ontology matching in a dynamic and distributed environment where on-the-fly alignments are needed. Their approach exploits imperfect consensuses among heterogeneous data holders by combining the logic formalisms with Web repositories.

In (Todorov, & Geibel, 2008), the authors design a procedure for mapping hierarchical ontologies populated with properly classified text documents. Through the combination of structural and instance-based approaches, the procedure reduces the terminological and conceptual ontology heterogeneity, and yields certain granularity and instantiation judgments about the inputs.

Learning-Based Schema Matching

The learning-based solutions consider both schema information and the associated instance data. Various kinds of machine learning techniques have been adopted in ontology-matching area. The most common ones include text content classification, k-nearest neighbor, Naive Bayes, and decision tree techniques. While taking advantages of extra clues contained in instance data, learning-based solutions are prone to run a longer time than rule-based solutions do (mostly because of the data training phase). Also, the difficulty in getting enough and/or good-quality data is a potential problem.

In (Doan et al., 2003), Doan, A. et al. describe GLUE that employs machine learning techniques to find semantic mappings between ontologies. A Metalearner is used to combine the predictions from both Content Learner and Name Learner; a similarity matrix is then built; and common knowledge and domain constraints are incorporated through a Relaxation Labeler. In addition, GLUE has been extended to find complex mappings.

Williams, A.B. and Tsatsoulis, C. (Williams, & Tsatsoulis, 1999) present their theory for learning ontologies among agents with diverse conceptualizations to improve group semantic concept search performance. The authors introduce recursive semantic context rule learning and unsupervised concept cluster integration to address the issue of how agents teach each other to interpret and integrate knowledge.

Soh, L.K. describes a framework for distributed ontology learning in a multiagent environment (Soh, 2002). The objective is to improve communication and understanding among the agents while agent autonomy is still preserved. Each agent maintains a dictionary for its own experience and a translation table, and the concept learning and interpretation are based on a description vector.

(Ding, Peng, & Pan, 2004; Ding et al., 2005; Pan et al., 2005) are a series of work in ontology matching based on a Bayesian (BN) approach. The methodology is built on BayesOWL. The algorithm learns probabilities using the naive Bayes text classification technique; then these probabilities and original ontologies are translated into the BN structures; finally, the algorithm finds new mappings between concepts.

(Afsharchi, Far, & Denzinger, 2006) presents a general method for agents using ontologies to teach each other concepts to improve their communication and thus cooperation abilities. An agent gets both positive and negative examples for a concept from other agents; it then makes use of one of its known concept learning methods to learn the concept in question, involving other agents again by taking votes in case of knowledge conflicts.

Madhavan, J. et al. (Madhavan et al., 2005) use a corpus of schemas and mappings to augment the evidence about the schemas being matched. The algorithm exploits a corpus in two ways. It first increases the evidence about each element by including evidence from similar elements in the corpus; then it learns statistics about elements and their relationships and uses them to infer constraints to prune candidate mappings.

(Wang et al., 2007) tackles the challenge of aligning multiple concepts simultaneously. Two statistically-grounded measures (Jaccard and Latent Semantic Analysis) are explored to build conversion rules that aggregate similar concepts, and different ways of learning and deploying the multi-concept alignment are evaluated.

In order to solve the problem of low precision resulted from ambiguous words, Gracia, J. et al. (Gracia et al., 2007) introduce techniques from Word Sense Disambiguation. They validate the mappings by exploring the semantics of the ontological terms involved in the matching process. They also discuss techniques to filter out mappings resulting from the incorrect anchoring of ambiguous terms.

Lambrix, P., Tan, H., and Xu, W. (Lambrix, Tan, & Xu, 2008) describe Support Vector Machine (SVM)-based algorithms to align ontologies using literature. The authors have discovered: (1) SVM-Single and Naïve Bayes obtain similar results; (2) the combinations of terminological using WordNet (TermWN) with SVM-Single and with SVM-Plural lead to a large gain in precision compared to TermWN and SVM-Plural.

Ontology and Context

Context refers to the conditions, constraints, and circumstances that are relevant to the conceptual model of interest. While an ontology is regarded as an explicit encoding of a domain model, a context can be viewed as an explicit encoding of a domain model that is expected to be local and may contain one party's subjective view of the domain (according to the Context & Ontologies workshop series). In the Workshop on Context, Information and Ontologies (CIAO 09), a slightly different definition was used as "Context needs to be defined semantically, specifically in terms of an ontology" An extensive survey of the term "context" can be found in (Tan, Goh, & Lee 2009), with specific reference to context-aware computing to facilitate Business-to-Business (B2B) collaboration. Both contexts and ontologies play a crucial role in knowledge representation and reasoning. All papers listed below make use of context information in ontologies.

In (Bouquet, 2007), an ontology schema is viewed as a context, i.e., as a partial and approximate representation of the world from a software agent's perspective. A schema cannot be assigned any arbitrary interpretation, as the meaning of the expressions used to label nodes (and possibly arcs) may be constrained by shared social conventions or agreements expressed in some lexical or domain ontologies. Accordingly, the author proposes that a schema matching method can be viewed as an attempt of coordinating intrinsically context-dependent representations by exploiting socially negotiated constraints on the acceptable interpretations of the labels as codified in shared artifacts like lexicons or ontologies.

Paulheim, H., Rebstock, M., and Fengel, J. show that community-driven referencing can be realized using a context-sensitive referencing service in a way that the user administration is transparent to the referencing system (Paulheim, Rebstock, & Fengel, 2007). The authors demonstrate that a context-sensitive semantic referencing service, combined with users' ratings, can be used for providing community-based semantic referencing. Both are feasible approaches for ontology mapping disambiguation, each having its advantages and drawbacks.

Heckmann, D. et al. (Heckmann et al., 2007) revisit the top-level ontology Gumo for the uniform management of user and context models in a Semantic Web environment. They discuss design decisions, while putting the focus on ontological issues. The structural integration into user model servers, especially into the U2M-serModel&ContextService, is also presented. The authors show ubiquitous applications using the user model ontology Gumo, together with the user model

markup language UserML. Finally, they ask how data from Web 2.0 (especially from a social tagging application) as a basis for user adaptation and context-awareness could influence the ontology.

The authors in (Hu, Qu, & Cheng 2008) propose a divide-and-conquer approach to matching large ontologies. They develop a structure-based partitioning algorithm, which partitions entities of each ontology into a set of small clusters and constructs blocks by assigning RDF Sentences to those clusters. Then, the blocks from different ontologies are matched based on precalculated anchors, and the block mappings holding high similarities are selected. Finally, two powerful matchers, V-DOC and GMO, are employed to discover alignments in the block mappings. Comprehensive evaluation on both synthetic and real world data sets demonstrates that this approach both solves the scalability problem and achieves good precision and recall with significant reduction of execution time.

In (Panayiotou, & Bennett 2008), the authors define the notion of cognitive learning context that refers to multiple and possibly inconsistent ontologies about a single topic. They discuss how this notion relates to the cognitive states of ambiguity and inconsistency. This work shows that discrepancies in viewpoints can be identified via the inference of conflicting arguments from consistent subsets of statements. Two types of arguments are discussed, i.e., arguments inferred directly from taxonomic relations between concepts, and arguments about the necessary and jointly sufficient features that define concepts.

(Li et al., 2009) presents a dynamic multistrategy ontology alignment framework, named RiMOM. The key insight in this framework is that similarity characteristics between ontologies may vary widely. The authors propose a systematic approach to quantitatively estimate the similarity characteristics for each alignment task and propose a strategy selection method to automatically combine the matching strategies based on two estimated factors. In the approach, they consider both textual and structural characteristics of ontologies, and their system was among the top three performers in the benchmark data sets in the 2006 and 2007 campaigns of the Ontology Alignment Evaluation Initiative (OAEI).

Seddiqui, M.H. & Aono, M. (Seddiqui, & Aono 2009) assume that an ontology is typically given in RDF (Resource Description Framework) or OWL (Web Ontology Language) and can be represented by a directed graph. Their proposed algorithm, Anchor-Flood algorithm, boasting of $O(nlogn)$ computation on the average (contrasting to a typical $O(n^2)$ complexity), starts off with an anchor, a pair of "look-alike" concepts from each ontology, gradually exploring concepts by collecting neighboring concepts, thereby taking advantage of locality of reference in the graph data structure. Moreover, since they only focus on segment-to-segment comparison, regardless of the entire size of ontologies, this algorithm not only achieves high performance, but also resolves the scalability problem in aligning ontologies.

Najar, S. et al. (Najar et al., 2009) review several context models proposed in different domains: content adaptation, service adaptation, and information retrieval. According to their insight, the authors propose an ontology-based context model focusing on the business processes domain. The framework analyzes and compares different context models. Such a framework aims to help understanding and analyzing different models, and consequently, the definition of new ones. The framework is based on the fact that context-aware systems use context models in order to formalize and limit the notion of context, and the observation that relevant information differs from a domain to another and depends on the effective use of such information.
Automated Semantic Matching of Ontologies with Verification (ASMOV) is an algorithm proposed in (Jean-Mary, Shironoshitaa, & Kabukaa 2009). It uses lexical and structural

characteristics of two ontologies to iteratively calculate a similarity measure between them, derives an alignment, and then verifies it to ensure that it does not contain semantic inconsistencies. Experimental results are presented that measure the algorithm's accuracy using the Ontology Alignment Evaluation Initiative (OAEI) 2008 tests, and that evaluate its use with two different thesauri: WordNet and the Unified Medical Language System (UMLS). These results show the increased accuracy obtained by combining lexical, structural, and extensional matchers with semantic verification, and demonstrate the advantage of using a domain-specific thesaurus for the alignment of specialized ontologies.

A framework is presented in (Mayer, Neumayer, & Rauber, 2009) to (semi-) automatically determine the context of creation and usage of digital objects, which is achieved through the analysis of the relationship of digital objects across a number of dimensions. Various facets of context along with different dimensions are automatically detected, and then are combined in pivot-table inspired views at multiple levels of granularity, which in turn allow the extraction of the most appropriate connections to other digital objects. In addition, the authors claim that this contact can be used for a wide range of applications.

Confidence Interval Applications

A confidence interval is a single observation of a random interval, calculated from a random sample by a given procedure, so that the probability that the interval contains an unknown population parameter θ is $(1-\alpha)$, which is also known as the confidence level or confidence coefficient. For example, if a confidence level of 95% is expected, then α equals 0.05. Confidence intervals have been widely applied in many different domains.

In (Altman, 1998), Altman, D. describes the number needed to treat (NNT) as a useful way of reporting the results of randomized, controlled trials. In a trial comparing a new treatment with a standard one, the NNT is the estimated number of patients who need to be treated with the new treatment rather than the standard treatment for one additional patient to benefit. It can be obtained for any trial that has reported a binary outcome. Altman, D. demonstrates how confidence intervals for this measure are calculated. As the author shows, a confidence interval for an absolute risk reduction (ARR) from, for example, -5% to +25%, inverts to a confidence interval that goes from a number needed to treat to benefit (NNTB) of 4, through infinity to a number needed to treat to harm (NNTH) of 20.

The location quotient (LQ) ratio is a useful measure that allows quantification and comparison of health and other outcomes across defined geographical regions. Beyene, J. and Moineddin, R. (Beyene, & Moineddin, 2005) present statistical methods that can be used to construct confidence intervals for LQs. The delta and Fieller's methods are generic approaches for a ratio parameter, whereas the generalized linear modelling framework is a useful re-parameterization that is particularly helpful for generating profile-likelihood-based confidence intervals for the LQ. In this work, the authors carry out a simulation experiment to assess the performance of each of the analytic approaches, with a health utilization data set used for illustration.

Yan, Y., Osadciw, L.A., and Chen, P. (Yan, Osadciw, & Chen 2008) propose a multistep statistical procedure to determine the confidence interval of the number of features that should be retained in appearance-based face recognition. MIZM (Modified Indifference Zone Method) is adopted to estimate the confidence interval of the number of features. MIZM overcomes the singularity problem in face recognition and extends the indifference zone selection. The simulation results on the ORL, UMIST, and FERET databases show that the overall recognition performance based on MIZM has improved. The relatively small number of features also

indicates the efficiency of the proposed feature selection method.

In (Brannath, Mehta, & Posch, 2009), the authors provide a method for obtaining confidence intervals, point estimates, and p-values for the primary effect size parameter at the end of a two-arm group sequential clinical trial. The method is based on applying an adaptive hypothesis testing procedure to a sequence of dual tests derived from the stage-wise adjusted confidence interval. Extensive simulation experiments, supported by an empirical characterization of the conditional error function, demonstrate that for all practical purposes the coverage is exact and the point estimate is median unbiased. The methodology is illustrated by an application to a clinical trial of deep brain stimulation for Parkinson's disease.

Ontology-Based E-Services

The application of ontologies in e-service environments has been studied widely. In (Honavar et al., 2001), Honavar, V. et al. describe several challenges in information extraction and knowledge acquisition from heterogeneous, distributed, autonomously operated, and dynamic data sources when scientific discovery is carried out in data-rich domains. They outline the key elements of algorithmic and systems solutions for computer-assisted scientific discovery in such domains, including ontology-assisted approaches to customizable data integration and information extraction from heterogeneous and distributed data sources. Ontology-driven approaches to exploratory data analysis from alternative ontological perspectives are also discussed.

With the advent of Semantic Web, knowledge-based interoperability in Virtual Enterprises faces a new technological shift, in which ontologies and Semantic web technologies play a major role. Exploiting the explicit semantic description of the domain of discourse allows reasoning and automatically acquiring semantic relations between two different domains of discourses. Such semantic relations would be further applied in converting data between such domains. (Silva, Rocha, & Cardoso 2003) proposes MAFRA–Mapping FRAmework to cover phases of the ontology mapping process, including analysis, specification, representation, execution, and evolution. The execution strategy and methodology are the focus of this paper. The MAFRA Toolkit has been applied in the European project Harmonise, which aims to provide solutions for (semi-) automatic interoperability between major operators in e-tourism.

An ontology-based information retrieval model for the Semantic Web is presented in (Song et al., 2005). The authors generate an ontology through translating and integrating domain ontologies. The terms defined in the ontology are used as metadata to markup the Web's content; these semantic markups are semantic index terms for information retrieval. The equivalent classes of semantic index terms are obtained by using a description logic reasoner. It is claimed that the logical views of documents and user information needs, generated in terms of the equivalent classes of semantic index terms, can represent documents and user information needs well, so the performance of information retrieval can be improved when a suitable ranking function is chosen.

Tijerino, Y. et al. introduce an approach, TANGO, to generate ontologies based on table analysis (Tijerino et al., 2005). TANGO aims to understand a table's structure and conceptual content; discover the constraints that hold between concepts extracted from the table; match the recognized concepts with ones from a more general specification of related concepts; and merge the resulting structure with other similar knowledge representations. The authors claim that TANGO is a formalized method of processing the format and content of tables that can serve to incrementally build a relevant reusable conceptual ontology.

Web services are increasingly utilized by organizations that want to improve responsiveness and efficiency. While they may be used in an isolated way, the need of integrating them as part of workflow processes is increasingly felt, and the creation of applications composed of dynamically selected basic services entails facing two essential issues: how to efficiently discover Web services and how to allow and facilitate their composition. In (Negri et al., 2006), the authors propose an agent-based framework representing an attempt of giving an answer to such problems. Its peculiar characteristic and strength is the integration of the agent technology with other key emerging technologies, i.e., Semantic Web, Web service, rule engine, and workflow technologies. The multiagent system, which constitutes the backbone of the framework, represents the "glue" that holds these pieces together and makes them perform properly. The framework has been experimented and evaluated in the realization of a prototype of an e-travelling system.

Ontologies are often used to improve search applications. Quality of ontology plays an important role in these applications. An important body of work exists in both information retrieval evaluation and ontology quality assessment areas. However, there is a lack of task- and scenario-based quality assessment methods. In (Strasunskas, & Tomassen, 2008), the authors discuss a framework to assess the fitness of ontologies for use in ontology-driven Web search. They define metrics for ontology fitness to particular search tasks and metrics for ontology capability to enhance recall and precision. Further, they discuss the applicability of the proposed framework and the value of ontology quality in ontology-driven Web search.

Context-awareness (CA) has been applied in different domains, particularly in ubiquitous computing, to provide better value-added services. CA has also been used, albeit sporadically, in business related applications. Tan, P.S., Goh, A.E.S., & Lee, S.S.G. (Tan, Goh, & Lee 2008) discuss their research effort in enhancing Business-to-Business (B2B) collaborations through context awareness, specifically to support the formation of short and dynamic connectivity between partners collaborating in a supply chain. Through an understanding of context awareness and how companies evaluate and select prospective suppliers, a B2B Context Model is proposed to discover and match suitable partners. Note that the prospective partners are represented as services in a service-oriented architecture (SOA) paradigm.

MAIN FOCUS OF THE CHAPTER

Issues, Controversies, Problems

There are many different definitions for an ontology. In this chapter, the one in (Singh, & Huhns, 2005) is adopted. Note that "properties and attributes" below are also known as "relationships."

> An ontology is a computational model of some portion or domain of the world. The model describes the semantics of the terms used in the domain. It is often captured in some form of a semantic network—a graph whose nodes are concepts or individual objects and whose arcs represent relationships or associations among the concepts. The network is augmented by properties and attributes, constraints, functions, and rules, which govern the behavior of the concepts.

The following qualitative equation shows the components of an ontology:

$$Ontology = Concepts + Relationships + Constraints. \quad (1)$$

The meaning of a concept, i.e., its semantics, is therefore determined by three aspects: (1) the name of the concept, (2) the properties of the concept, and (3) the relationships of the concept. These three features together specify a conceptual model for each concept from the viewpoint of an ontology designer.

Considering the fact that anyone can design ontologies according to his/her own conceptual view of the world, ontological heterogeneity among different parties becomes an inherent characteristic. The heterogeneous semantics occurs in two ways. (1) Different ontologies could use different terminologies to describe the same conceptual model. That is, different terms could be used for the same concept, or an identical term could be adopted for different concepts. (2) Even if two ontologies use the same name for a concept *C*, the associated properties and the relationships with other concepts for *C* are most likely to be different. Essentially, out of the aforementioned three semantic aspects, both properties and relationships belong to an ontology concept's contextual information. Any successful strategy to handle the ontology heterogeneity issue has to take into consideration the clues drawn from the context. Only by this means can a meaningful and helpful matching result be expected.

Besides the ontological heterogeneity issue discussed above, there are many other difficulties in matching ontologies. The following challenges are discussed in (Ding et al., 2005): one-to-many matching (where a term in one ontology may match with a few terms in the other ontology); uncertain matching (where a term in one ontology may have a similar meaning to another term in another ontology but the meaning of terms may not be an exact match); and structural difference (two terms with the same or similar meaning are structured differently in different ontologies). Other major issues (as well as sub-issues) of ontology matching are also discussed in (Euzenat, & Shvaiko, 2007) and (Shvaiko, & Euzenat, 2008). For example, one of the sub-issues in ontology matching is how to aggregate the similarity values of different comparators (similarity in concept names, similarity in concept properties, and similarity in concept relationships, etc.), for which we propose to use the Artificial Neural Network approach. Note that in (Do, & Rahm, 2002) and (Sayyadian et al., 2005), the authors present COMA and eTuner, respectively, to handle this issue. Another issue is that of alignment extraction that is concerned with the selection of the matching entities (concepts) given the similarity values between many different pairs of entities (concepts). We propose the agglomerative hierarchical clustering approach to address this issue. Due to the length limitation, a complete review of existing work and an explanation of difficulties in the alignment extraction issue can be found in Section 5.7 of (Euzenat, & Shvaiko, 2007).

Solutions and Recommendations

Solution Overview

We present an innovative algorithm, Context-Sensitive Matching (CSM), to reconcile ontologies from heterogeneous sources. In brief, CSM takes three steps to match ontologies. (1) Based on the insight of the important role played by contextual information in ontology matching, our algorithm first attempts to identify the domain information of ontology concepts of interest. Such identification is based on the inference on contextual information through a formal, robust statistical model. (2) Upon obtaining the estimated domain knowledge, CSM then applies an Artificial Neural Network technique to learning and adjusting the different weights for different semantic aspects, i.e., concept name, concept properties, and concept relationships. (3) According to these learned weights, an agglomerative clustering algorithm is adopted to generate clusters of equivalent (or similar) concepts from different ontologies.

Statistical Model to Acquire Domain Information

As for the concepts of interest, which domain they belong to is an important piece of contextual information. Consider two concepts from two completely different domains, it is meaningless to match up these concepts because there is very little, or even no, intersection between them. For example, concept *Bat* may be within an animal domain, or it may appear in a sports domain. It does not make much sense to match the animal *Bat* with the sports-related *Bat*. Another example is *Bachelor*: it may refer to a kind of academic degree, or it may talk about a single male.

Therefore, the first step in CSM is to identify the domain information of concepts in question. Notice that if a concept is considered separately, it may not be straightforward to identify its domain information, simply because any concept is constrained by its neighboring concepts, more or less. Our idea is to consider a sample of all related concepts; based on the domain information from these sample concepts having relationships with the concept of interest, *C*, we calculate the likelihood of the concept *C* belonging to a specific domain at a confidence level.

It is well known that the most common relationships in most real-world ontologies are *subClassOf* and *superClassOf*. For a concept *C* of interest, we extract all of its direct parents, ancestors, siblings, direct children, and descendants. A subset of all these "surrounding" concepts serves as our sample in the statistical model. Based on the following information:

- the sample size: n
- the number of the candidate domain names: m
- the number of the sample surrounding concepts that belong to the i^{th} domain: x_i
- the adjusted sample proportion with regard to the i^{th} domain: $p_i^* = \dfrac{x_i + 2}{n + 4}$
- the confidence level: $(1-\alpha)$

we calculate the simultaneous $(1-\alpha)$ confidence interval estimate for the likelihood, p_i, that concept *C* belongs to the i^{th} domain as

$$p_i^* - z_{\alpha/2m}\sqrt{\frac{p_i^*(1-p_i^*)}{n+4}} < p_i < p_i^* + z_{\alpha/2m}\sqrt{\frac{p_i^*(1-p_i^*)}{n+4}}, \qquad (2)$$

where $z_{\alpha/2m}$ is the critical value from the normal density table in statistics textbooks, (McClave, & Sincich, 2002; Rencher, 1997) for example. The Formula (2) was first proposed in (Agresti, & Coull, 1998). Researchers have shown that this confidence interval works well even for small sample size and extreme sample proportion (i.e., p_i^* is close to 0 or 1). In other words, true proportion for the actual population can be estimated well.

To sum up, in case where there is ambiguity regarding which domain the concept of interest belongs to, we adopt **a formal, robust statistical model** to infer such important contextual information. By taking into consideration the surrounding concepts which embody the most common relationships (*subClassOf* and *superClassOf*), instead of the concept in question alone, our model integrates the contextual information into the ontology-matching process as much as possible, while obtaining **statistically valid** domain knowledge, and therefore associates the ontology matching with high degree of context-awareness.

Artificial Neural Network to Learn Weights for Semantic Aspects

As discussed earlier, the semantics of an ontology concept is determined by three aspects: (1) concept name, (2) concept properties, and (3) concept relationships. These three features together specify a conceptual model for each concept from the viewpoint of an ontology designer. Any ontology-matching algorithm, either rule-based or learning-based, needs to handle some or all of these three semantic aspects, by different rules or machine learning techniques.

The rule-based algorithms usually share the disadvantage of ignoring the additional information from instance data. In addition, it is unavoidable to determine the corresponding weights for

different semantic aspects, reflecting their different importance (or contributions) in ontology matching. Many existing rule-based algorithms make use of human heuristics to predefine these weights. On the other hand, while taking advantage of extra clues contained in instances, the learning-based algorithms are likely to be slower. Moreover, the difficulty in getting enough good-quality data is a more severe problem. Compared with schemas, instance data usually exhibit much less variety. Therefore, most existing learning-based algorithms make use of instance data, more or less. In our opinion, machine learning techniques are essential in ontology matching; however, at the same time, it is preferable to avoid the problem of lacking instance data, either in quality or in quantity, which is *very common* for real-world ontologies (verified in the later section of this chapter, "Evaluation on Real-World Ontologies"). The learning process in our proposed CSM algorithm is therefore carried out at the schema level, instead of the instance level.

Given a pair of ontologies, it is reasonable to assume that the contributions of different semantic aspects to ontology understanding should be independent of specific concepts, although it is recognized that much design diversity might exist. In fact, different contributions, which are the foundation for different weights, are characteristics of ontologies viewed as a whole. That is, during ontology matching, weights are features with regard to ontologies, rather than individual concepts. Therefore, it is possible to *learn these weights for all concepts by training examples from a subset of concepts*.

(1) Concept Similarity

(1.1) Similarity in Concept Names

The similarity s_1 between a pair of concept names is a real value in [0, 1]. If two names have an exact string matching or are synonyms of each other in WordNet (Miller, 1995), then s_1 has a value of 1; otherwise, s_1 is calculated according to

$$s_1 = 1 - \frac{d}{l}, \qquad (3)$$

where *d* stands for the edit distance between two strings, and *l* for the length of the longer string.

(1.2) Similarity in Concept Properties

Given two lists of concept properties (including those inherited from ancestors), p_1 and p_2, their similarity s_2 is a real value in the range of [0, 1], and s_2 is calculated according to

$$s_2 = \frac{n}{m}, \qquad (4)$$

where *n* is the number of pairs of properties matched, and *m* is the smaller cardinality of lists p_1 and p_2. In order for a pair of properties, one from p_1 and the other from p_2, to be matched, their data types should be the same or compatible (*float* and *double* for example), and their property names should be similar with each other.

(1.3) Similarity in Concept Relationships

As mentioned before, the most common relationships are *subClassOf* and *superClassOf* (this is verified in the later section of this chapter, "Evaluation on Real-World Ontologies"). To obtain a better matching result, not only the direct parents of a concept, but also all of its ancestors are considered as well, i.e., concepts along the path from this concept up to the common built-in root "Thing." Given two lists of concept ancestors, a_1 and a_2, their similarity s_3 is a real value in the range of [0, 1], and is obtained by first calculating the

similarity values for pairwise concepts (one from a_1, the other from a_2, considering all combinations), then assigning the maximum value to s_3. Notice that this is a recursive procedure but is guaranteed to terminate, because 1) the number of concepts is finite; and 2) it is assumed that "Thing" is a common root for two ontologies being matched.

(2) Concept Similarity Matrix

After s_1, s_2, and s_3 between two concepts, C_1 and C_2, are calculated, the overall similarity value, s, is obtained as the weighted sum of s_1, s_2, and s_3:

$$s = \sum_{i=1}^{3} (w_i s_i), \qquad (5)$$

Where $\sum_{i=1}^{3} w_i = 1$, and s in [0, 1]. Notice that w_i's are randomly initialized, and will be adjusted through a learning process that is discussed in the next section. For two ontologies being matched, O_1 and O_2, the similarity values are calculated for pairwise concepts (one from O_1, the other from O_2, considering all combinations). Then an $n_1 \times n_2$ matrix M is built to record all values calculated, where n_i is the number of concepts in O_i. The cell $[i,j]$ in M stores the similarity value between the i^{th} concept in O_1 and the j^{th} concept in O_2.

(3) Weight Learning via Artificial Neural Network

The learning problem is designed as follows.

- Task *T*: match two ontologies
- Performance measure *P*: *Precision*, *Recall*, *F-Measure*, and *Overall*, with regard to manual matching
- Training experience *E*: a set of equivalent concept pairs by manual matching
- Target function *V*: a pair of concepts $\rightarrow \Re$
- Target function representation: $\hat{V}(b) = \sum_{i=1}^{3} (w_i s_i)$

An artificial neural network is chosen as the learning technique, because: instances are represented by attribute-value pairs; the target function output is a real-valued one; and fast evaluation of the learned target function is preferable.

(3.1) Network Design

A two-layer 3×1 network is adopted in CSM, as shown in Fig.1. The input is a vector $\vec{s}$, which consists of s_1, s_2, and s_3, representing the similarity in name, properties, and relationships, respectively, for a given pair of concepts. The output is s, the overall similarity value between these two concepts, calculated according to Formula (5). Notice that a linear function might not be powerful enough to reflect the true relationships among w_i's. However, Mitchell (1997) states that "the delta rule converges toward a best-fit approximation to the target concept even when the training examples are not linearly separable." If more relationships among ontology concepts are to be considered, then one or more layers of hidden units might need to be added to express a rich variety of nonlinear decision surfaces.

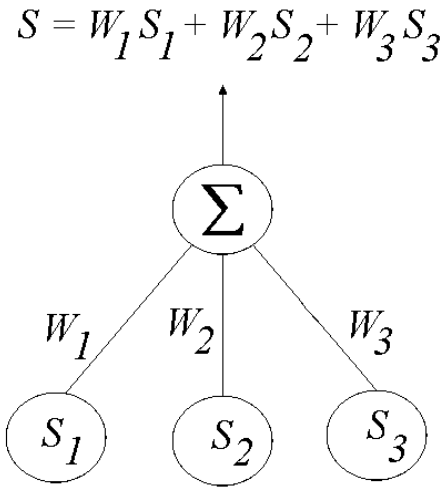

Fig.1: Neural Network Structure

Initially, a concept similarity matrix M is obtained for O_1 and O_2, with w_i's being initialized randomly. A set of concepts is randomly picked up from O_1, and their equivalent concepts are found in O_2 by a manual matching. Each of such manually matched pairs will be processed by CSM, and the similarity values in name, properties, and relationships for these two concepts are calculated and used as a training example to the network in Fig.1.

(3.2) Hypothesis Space and the Searching Strategy

In this learning problem, it is assumed that the hypothesis space is a three-dimensional space consisting of w_1, w_2, and w_3. For every weight vector $\vec{w}$ in the hypothesis space, the learning objective is to find the vector that best fits the training examples. Gradient descent (delta rule) is adopted as the training rule, and the searching strategy is to find the hypothesis, i.e., weight vector, that minimizes the training error with regard to all training examples. According to (Mitchell 1997), a standard definition of the training error E of a hypothesis is given by

$$E(\vec{w}) \equiv \frac{1}{2}\sum_{d \in D}(t_d - o_d)^2 , \qquad (6)$$

where D is the set of training examples, t_d is the target output for training example d, and o_d is the network output for d. The above formal definition is customized according to the characteristics of the learning problem here. For any training example d, instead of a given target value t_d, some other values are needed. The intuition is, a given pair of manually matched concepts corresponds to a cell $[i,j]$ in M, therefore, the value of cell $[i,j]$ should be the maximum one in both row i and column j. Suppose the maximum value for row i and column j are t_r and t_c, respectively, then the customized description of E is

$$E(\vec{w}) \equiv \frac{1}{2}\sum_{d \in D}[(t_r - o_d) + (t_c - o_d)]^2 . \qquad (7)$$

Accordingly, the weight update rule for gradient descent in CSM is

$$\Delta w_i \equiv \eta \sum_{d \in D}[(t_r - o_d) + (t_c - o_d)]s_{id} , \qquad (8)$$

where η is the learning rate, and s_{id} is the s_i value for a specific training example d.

Agglomerative Clustering to Generate Matching Results

Upon obtaining the learned weights for three semantic aspects (name, properties, and relationships), the similarity matrix is recalculated between every two ontologies. An agglomerative clustering algorithm is then adopted to form a set of *superconcepts*. Here, the superconcept is defined as a set of original concepts. Within each superconcept, all *components*, i.e., original concepts, are from different ontologies; at the same time, they are equivalent to each other. Our goal is to find all superconcepts across a set of ontologies. Because the number of superconcepts is not known prior to the matching process, an agglomerative clustering algorithm fits our needs.

In the following procedure, similarity between a pair of clusters, (*a*) and (*b*), is denoted by *s*[(*a*), (*b*)], which is calculated as the average similarity between all pairs of concepts from cluster (*a*) and cluster (*b*), i.e., $s[(a),(b)] = \frac{1}{uv}\sum_{i=1}^{u}\sum_{j=1}^{v} s[(a_i),(b_j)]$, where ($a_i$) and ($b_j$) are component concepts in cluster (*a*) and cluster (*b*), respectively; and *u* and *v* are the numbers of concepts in cluster (*a*) and cluster (*b*), respectively.

Input:

Ontologies O_1, O_2,…, and O_k

M_{ij}'s ($i, j \in [1,k]$ and M_{ij} is the recalculated similarity matrix between O_i and O_j)

Output:

A set of superconcepts

Begin

1) Each original concept forms a *singleton* cluster
2) Find a pair of clusters, (*a*) and (*b*), such that their similarity *s*[(*a*), (*b*)] = max (*s*[(*m*), (*n*)])
3) If *s*[(*a*), (*b*)] > *similarity threshold*, go to step 4, otherwise go to step 7
4) Merge (*a*) and (*b*) into a new cluster (*a*+*b*)
5) For all ontologies containing (*a*) **and** (*b*), update their matrices by deleting the row and column corresponding to (*a*) and (*b*); for other ontologies whose matrices contain (*a*) **or** (*b*), recalculate the row/column corresponding to (*a*) or (*b*), using the similarity between the new cluster, (*a*+*b*), and any existing cluster (*c*):
$$s[(a+b),(c)] = \frac{1}{2}(s[(a),(c)] + s[(b),(c)])$$
6) Repeat steps 2 and 3
7) Output current clusters as the set of superconcepts

end

Pseudocode for Agglomerative Clustering

The above procedure shows that the key to correctly obtain a set of superconcepts depends on whether or not a suitable *similarity threshold* can be determined. This is not trivial at all, and the following strategy is taken to tackle this challenge. First of all, let the number of concepts in O_i be n_i ($i \in [1,k]$). Without loss of generality, suppose $n_1 \geq n_j$ ($j \in [2,k]$). The number of total clusters (superconcepts) should then be within the range of $[n_1, \sum_{i=1}^{k} n_i]$. Possible values of

threshold are real numbers in [0, 1]. With the decrease of threshold value, the number of superconcepts will decrease as well. Let us pay attention to two extreme situations. 1) If threshold is set to 1, then no pair of concepts will be regarded as equivalent ones, and no new clusters are to be generated. Therefore, there will be $\sum_{i=1}^{k} n_i$ resultant superconcepts. 2) On the other hand, if threshold is set to 0, then every concept in O_j ($j \in [2,k]$) finds its equivalent one in O_1, and there will be n_1 superconcepts. The number of superconcepts changes with the changing of threshold value. This results in a certain shape of curve. If after an initial drop, there emerges a plateau, followed by a second drop, then it is reasonable to conclude that threshold can possibly be assigned the value corresponding to the beginning of this plateau (Fig.2). The intuition is, the semantic similarity between non-equivalent concepts and that between equivalent concepts are different, and this difference could be remarkable enough to form a plateau. In addition, the starting point of the plateau indicates the point from which the superconcept number starts to converge.

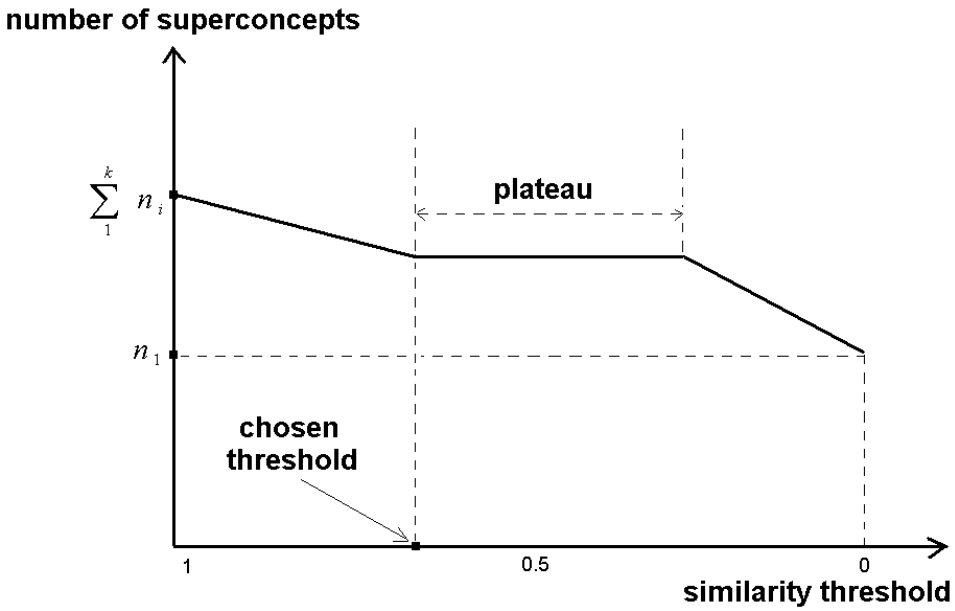

Fig.2: Evolution of Superconcept Numbers with Changing of Similarity Thresholds

Evaluation on Real-World Ontologies

Our hypothesis is, by integrating the critical contextual information into the ontology-matching process, it is supposed to handle well the heterogeneity implicit in independently designed ontologies, so that distributed agents are able to reconcile heterogeneous semantics during their communication and coordination with each other in electronic business. In order to verify this hypothesis, two sets of real-world ontologies are adopted to evaluate our proposed CSM algorithm.

(1) Two Sets of Test Ontologies

The first set contains nine real-world ontologies, all built and maintained by professionals. They belong to three different domains, i.e., “Building,” “Material,” and “Education,” and can be used

in different types of electronic business applications, real estate and Web-based education for example.

1. **space**: http://212.119.9.180/Ontologies/0.3/space.owl
2. **ops**: http://moguntia.ucd.ie/owl/Operations.owl
3. **swap**: http://svn.mindswap.org/pychinko/pychinko/allogtests/mindswapRealized.rdf
4. **mgm**: http://ontologies.isx.com/onts/2005/02/isxbusinessmgmtont.owl
5. **gfo**: http://www.onto-med.de/ontologies/gfo.owl
6. **akt**: http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/akt_ontology_LITE.owl
7. **aktive**: http://www.mindswap.org/2004/SSSW04/aktive-portal-ontology-latest.owl
8. **ita**: http://www.mondeca.com/owl/moses/ita.owl
9. **Mid**: http://reliant.teknowledge.com/DAML/Mid-level-ontology.owl

Our second set of test ontologies are chosen from Swoogle (Swoogle Site 2010), a Google-like search engine specifically designed for the Semantic Web. These six ontologies are in either "Enterprise" or "E-business" domain.

1. **ontology_1**: http://wiki.infowiss.net/Spezial:Exportiere_RDF/E-Business_/_E-Commerce
2. **ontology_2**: http://wiki.infowiss.net/Spezial:Exportiere_RDF/E-Business
3. **ontology_3**: http://openean.kaufkauf.net/id
4. **ontology_4**: http://tw.rpi.edu/wiki/Special:ExportRDF/Middle-tier_database_caching_for_e-business
5. **ontology_5**: http://ontoware.org/swrc/swrc_v0.3.owl
6. **ontology_6**: http://derpi.tuwien.ac.at/~andrei/cerif.rdfs

(2) Measures

In the research area of ontology matching, there are four commonly adopted measures, with regard to the performance of (semi)automatic matching algorithms.

- *Precision p*: the percentage of the correct predictions over all predictions, representing the **correctness** aspect of the matching performance.
- *Recall r*: the percentage of the correct predictions over correct matching, estimating the **completeness** aspect of the matching performance.
- *F-Measure f* ($= \frac{2rp}{r+p}$): also known as *Harmonic Mean*, aiming to consider both *Precision* and *Recall*, and **avoid the bias** from adopting *Precision* or *Recall* alone
- *Overall o* ($= r(2 - \frac{1}{p})$): a measure on the **post-match effort**, i.e., how much human effort is needed to remove false matches and add missed ones.

(3) Experimental Results

In our experiment, we match up pairwise ontologies from the aforementioned sources, and then the matching results from CSM are compared with those from two experts. Finally, we calculate all four measures. The detailed results and a summary for the first set are demonstrated in Table 1 and Table 2, respectively. Due to the length limitation, we only show the summary of experimental results for the second set of test ontologies in Table 3.

Table 1: Evaluation of Pairwise Ontology Matching (the First Set)

	space	**ops**	**swap**	**mgm**	**gfo**	**akt**	**aktive**	**ita**	**Mid**
space		p = 0.67 r = 0.65 f = 0.66 o = 0.33	p = 0.71 r = 0.67 f = 0.69 o = 0.40	p = 0.83 r = 0.74 f = 0.78 o = 0.59	p = 0.69 r = 0.68 f = 0.68 o = 0.37	p = 0.72 r = 0.71 f = 0.71 o = 0.43	p = 0.71 r = 0.69 f = 0.70 o = 0.41	p = 0.70 r = 0.64 f = 0.67 o = 0.37	p = 0.68 r = 0.67 f = 0.67 o = 0.35
ops			p = 0.72 r = 0.71 f = 0.71 o = 0.43	p = 0.77 r = 0.74 f = 0.75 o = 0.52	p = 0.71 r = 0.66 f = 0.68 o = 0.39	p = 0.69 r = 0.68 f = 0.68 o = 0.37	p = 0.68 r = 0.65 f = 0.66 o = 0.34	p = 0.73 r = 0.64 f = 0.68 o = 0.40	p = 0.72 r = 0.70 f = 0.71 o = 0.43
swap				p = 0.72 r = 0.71 f = 0.71 o = 0.43	p = 0.76 r = 0.67 f = 0.71 o = 0.46	p = 0.71 r = 0.69 f = 0.70 o = 0.41	p = 0.77 r = 0.74 f = 0.75 o = 0.52	p = 0.67 r = 0.66 f = 0.66 o = 0.33	p = 0.75 r = 0.73 f = 0.74 o = 0.49
mgm					p = 0.67 r = 0.65 f = 0.66 o = 0.33	p = 0.77 r = 0.74 f = 0.75 o = 0.52	p = 0.83 r = 0.73 f = 0.78 o = 0.58	p = 0.81 r = 0.74 f = 0.77 o = 0.57	p = 0.83 r = 0.72 f = 0.77 o = 0.57
gfo						p = 0.80 r = 0.70 f = 0.75 o = 0.53	p = 0.79 r = 0.69 f = 0.74 o = 0.51	p = 0.83 r = 0.74 f = 0.78 o = 0.59	p = 0.82 r = 0.73 f = 0.77 o = 0.57
akt							p = 0.73 r = 0.69 f = 0.71 o = 0.43	p = 0.81 r = 0.72 f = 0.76 o = 0.55	p = 0.76 r = 0.74 f = 0.75 o = 0.51
aktive								p = 0.83 r = 0.71 f = 0.77 o = 0.56	p = 0.75 r = 0.72 f = 0.73 o = 0.48
ita									p = 0.77 r = 0.70 f = 0.73 o = 0.49
Mid									

Table 2: Summary of Experimental Results (the First Set)

	High	Low	Average	Median	Standard Deviation
Precision	0.83	0.67	0.75	0.74	0.05
Recall	0.74	0.64	0.70	0.70	0.03
F-Measure	0.78	0.66	0.72	0.72	0.04
Overall	0.59	0.33	0.46	0.44	0.08

Table 3: Summary of Experimental Results (the Second Set)

	High	Low	Average	Median	Standard Deviation
Precision	0.86	0.66	0.74	0.75	0.04
Recall	0.73	0.65	0.71	0.72	0.04
F-Measure	0.77	0.67	0.73	0.71	0.05
Overall	0.57	0.34	0.49	0.48	0.07

(4) Result Analysis

The characteristics of the first set of test ontologies are summarized in Table 4 (the data for the second set is not included due to the length limitation). In these randomly chosen ontologies, the most common relationships are *subClassOf* and *superClassOf* (from 67% to 79%). In addition, only two ontologies have instance data associated with schemas, with fairly low percentages (24% and 14%) over the whole file. **The data in Table 4 verify our earlier claims, i.e., 1) *subClassOf* and *superClassOf* are the most important relationships, and 2) obtaining enough good-quality instances is difficult.**

Table 4: Characteristics of Test Ontologies (the First Set)

Features	**space**	**ops**	**swap**	**mgm**	**gfo**	**akt**	**aktive**	**ita**	**Mid**
Max Depth	8	8	7	9	11	8	6	8	10
Concept Number	90	91	61	72	127	81	62	67	117
Relationship Number	158	139	87	109	162	116	85	101	203
sub/superClassOf Number	115	110	64	75	117	83	57	73	146
sub/superClassOf Percentage	73%	79%	73%	69%	72%	71%	67%	73%	72%
Instance Data Percentage	0%	0%	0%	0%	0%	24%	0%	0%	14%

The aforementioned experimental results demonstrate that the proposed methodology is in fact effective for matching ontologies. In particular, the average *Precision* and *Recall* are 0.74–0.75 and 0.70–0.71, respectively. Consequently, distributed agents would be enabled to efficiently reconcile heterogeneous semantics during their communication and coordination with each other in electronic business. Therefore, the proposed methodology will help with the interoperability between electronic business applications by greatly reducing human efforts in an otherwise manual ontology-matching process.

FUTURE RESEARCH DIRECTIONS

Despite of the large number of researchers involved, the heterogeneity problem in ontologies remains a challenging issue to handle. Some future research directions are envisioned here. The focus of CSM has been placed on locating equivalent concepts (superconcepts), leaving other mapping tasks as future work, for example, to discover parent-child pairs. Another possible future work is to include other relationships besides *subClassOf* and *superClassOf* into the matching process. There is no doubt that these two relationships are the most common and therefore the most important relationships to be taken into consideration. However, other relationships will definitely provide additional clues during the matching process. On the other hand, we need to be careful in dealing with such tradeoff between the extra effort and the (possibly) marginal gains. Lastly, we plan to compare our experimental results to those obtained using the more prominent ontology matchers introduced in the related work or the top matchers in the Ontology Alignment Evaluation Initiative (OAEI[1]). Some of these matchers are Falcon-AO, ASMOV, RiMOM, and AnchorFlood.

CONCLUSION

Electronic business has provided great opportunities to the current global economy by enhancing the capabilities of traditional businesses. In many cases, electronic business partners are chosen to be represented by service agents for the sake of better satisfying the imposed requirement for businesses to coordinate with each other. These service agents need to understand each others' service descriptions before successful coordination may happen, and ontologies developed by service providers can render help in this regard. Unfortunately, due to the heterogeneity inherent in independently designed ontologies, it is unavoidable for distributed agents to face semantic mismatches and misunderstandings. We propose an innovative algorithm, Context-Sensitive Matching, to reconcile heterogeneous ontologies. Our approach takes into consideration contextual information, via inference through a formal, robust statistical model; in addition, an Artificial Neural Network is utilized to learning weights for different semantic aspects; and finally, an agglomerative clustering algorithm is adopted to generate the final matching results. We have evaluated our methodology on real-world ontologies, followed by detailed, in-depth analysis of the experiment results. The evaluation affirms the promising performance of our approach. Therefore, we have successfully verified our hypothesis, that is, by integrating the critical contextual information into the ontology-matching process, it is supposed to well handle the heterogeneity implicit in independently designed ontologies, so that distributed agents are able to reconcile heterogeneous semantics during their communication and coordination with each other in electronic business.

[1] http://oaei.ontologymatching.org/

REFERENCES

Afsharchi, M., Far, B.H., & Denzinger, J. (2006). Ontology-guided learning to improve communication between groups of agents. In *proc. the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 06)*, Hakodate, Japan.

Agresti, A., & Coull, B.A. (1998). Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52, 119-126.

Altman, D.G. (1998). Confidence intervals for the number needed to treat. *BMJ Online Journal*, 317, 1309-1312.

Beyene, J., & Moineddin, R. (2005). Methods for confidence interval estimation of a ratio parameter with application to location quotients. *BMC Medical Research Methodology*, 5(32).

Bouquet, B. (2007). Contexts and ontologies in schema matching. In *proc. the third International Workshop on Contexts and Ontologies: Representation and Reasoning (C&O: RR 07)*, Roskilde University, Denmark.

Brannath, W., Mehta, C.R., & Posch, M. (2009). *Biometrics, the International Biometric Society*, 65(2), 539-546(8).

Castano, S., Ferrara, A., & Montanelli, S. (2003). H-MATCH: an algorithm for dynamically matching ontologies in peer-based systems. In *proc. the first VLDB International Workshop on Semantic Web and Databases (SWDB 03)*, Berlin, Germany.

Ding, L., Kolari, P., Ding, Z., Avancha, S., Finin, T., & Joshi, A. (2005). Using ontologies in the semantic web: a survey. *Technical report at University of Maryland, Baltimore County*.

Ding, Z., Peng, Y., & Pan, R. (2004). A Bayesian approach to uncertainty modeling in OWL ontology. In *proc. the International Conference on Advances in Intelligent Systems - Theory and Applications*, Luxembourg.

Ding, Z., Peng, Y., Pan, R., & Yu, Y. (2005). A Bayesian methodology towards automatic ontology mapping. In *proc. Technical Report WS-05-01 of AAAI Workshop on Contexts and Ontologies: Theory, Practice, and Applications*, Pittsburgh, PA.

Do H.H., & Rahm, E. (2002). Coma - a system for flexible combination of schema matching approaches. In *proc. 28th International Conference on Very Large Data Bases (VLDB 02)*, pp. 610–621, Hong Kong, China.

Doan, A., Madhavan, J., Dhamankar, J., Domingos, P., & Halevy, A. (2003). Learning to match ontologies on the Semantic Web. *The VLDB Journal*, *12(4)*.

Doan, A., & Halevy, A.Y. (2005). Semantic integration research in the database community: a brief survey. *AI Magazine*, *26(1)*, 83-94.

Dou, D., McDermott, D., & Qi, P. (2003). Ontology translation on the Semantic Web. In *proc. the International Conference on Ontologies, Databases, and Applications of Semantics*, Lecture Notes in Computer Science, Berlin: Springer-Verlag.

Euzenat, J., & Shvaiko, P. (2007). *Ontology Matching*. Springer-Verlag, Berlin Heidelberg (DE).
Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2005). Semantic Schema Matching. In *proc. the Thirteenth International Conference on Cooperative Information Systems (CoopIS 05)*, Agia Napa, Cyprus.

Giunchiglia, F., Yatskevich, M., & McNeill, F. (2007). Structure preserving semantic matching. In *proc. the Second International Workshop on Ontology Matching (OM 07)*, Busan, Korea.
Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2009). Semantic matching. Encyclopedia of Database Systems, 2561-2566, Springer.

Gracia, J., Lopez, V., Aquin, M., Sabou, M., Motta, E., & Mena, E. (2007). Solving semantic ambiguity to improve Semantic Web based ontology matching. In *proc. the Second International Workshop on Ontology Matching (OM 07)*, Busan, Korea.

Heckmann, D., Schwarzkopf, E., Mori, J., Dengler, D., & KrÄoner, A. (2007). The user model and context ontology GUMO revisited for future Web 2.0 extensions. In *proc. the third International Workshop on Contexts and Ontologies: Representation and Reasoning (C&O: RR 07)*, Roskilde University, Denmark.

Honavar, V., Andorf, C., Caragea, D., Silvescu, A., Reinoso-Castillo, J., & Dobbs, D. (2001). Ontology-driven information extraction and knowledge acquisition from heterogeneous, distributed biological data sources. In *proc. the IJCAI-2001 Workshop on Knowledge Discovery from Heterogeneous, Distributed, Autonomous, Dynamic Data and Knowledge Sources*, Seattle, WA.

Hu, B., Dasmahapatra, S., & Lewis, P. (2007). Emerging consensus in-situ. In *proc. the Second International Workshop on Ontology Matching (OM 07)*, Busan, Korea.

Hu, W., Qu, Y., & Cheng, G. (2008). Matching large ontologies: A divide-and-conquer approach. *Data and Knowledge Engineering, 67(1):140-160.*

Jean-Mary, Y.R., Shironoshitaa, E.P., & Kabukaa, M.R. (2009). Ontology matching with semantic verification. *Journal of Web Semantics*.

Lambrix, P., Tan, H., & Xu, W. (2008). Literature-based alignment of ontologies. In *proc. the Third International Workshop on Ontology Matching (OM 08)*, Karlsruhe, Germany.

Li, J., Tang, J., Li, Y., & Luo, Q. (2009). RiMOM: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering, 21(8):1218-1232*.

Madhavan, J., Bernstein, P.A., & Rahm, E. (2001) Generic schema matching with Cupid. In *proc. the Twenty-seventh VLDB Conference*, Roma, Italy.

Madhavan, J., Bernstein, P.A., Doan, A., & Halevy, A. (2005). Corpus-based Schema Matching. In *proc. the Twenty-first International Conference on Data Engineering (ICDE 05)*, Tokyo, Japan.

Mayer, R., Neumayer, R., & Rauber, A. (2009). Interacting with (semi-) automatically extracted context of digital objects. In *proc. the first Workshop on Context, Information, and Ontologies (CIAO 09)*, Heraklion, Greece.

McClave, J.T., & Sincich, T. (2002). *Statistics (9th ed.)*. Prentice Hall, Upper Saddle River, NJ.

Melnik, S., Garcia-Molina, H., & Rahm, E. (2002). Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In *proc. the Eighteenth International Conference on Data Engineering (ICDE 02)*, San Jose, CA.

Miller, A.G. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39-41.

Mitchell, T. (1997). *Machine Learning*, McGraw-Hill.

Najar, S., Saidani, O., Kirsch-Pinheiro, M., Souveyet, C., & Nurcan, S. (2009). Semantic representation of context models: a framework for analyzing and understanding. In *proc. the first Workshop on Context, Information, and Ontologies (CIAO 09)*, Heraklion, Greece.

Negri, A., Poggi, A., Tomaiuolo, M., & Turci, P. (2006). Agents for e-business applications. In *proc. the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 06)*, Hakodate, Japan.

Noy, N.F., & Musen, M.A. (2000). PROMPT: algorithm and tool for automated ontology merging and alignment. In *proc. the 17th National Conference on Artificial Intelligence (AAAI 00)*, AAAI Press, Menlo Park, CA, USA.

Pan, R., Ding, Z., Yu, Y., & Peng, Y. (2005). A Bayesian network approach to ontology mapping. In *proc. the International Semantic Web Conference (ISWC 05)*, Galway, Ireland.

Panayiotou, C., & Bennett, B. (2008). Cognitive context and syllogisms from ontologies for handling discrepancies in learning resources. In *proc. the third International Workshop on Contexts and Ontologies (C&O 08)*, Patras, Greece.

Paulheim, H., Rebstock, M., & Fengel, J. (2007). Context-sensitive referencing for ontology mapping disambiguation. In *proc. the third International Workshop on Contexts and Ontologies: Representation and Reasoning (C&O: RR 07)*, Roskilde University, Denmark.

Rencher, A.C. (1997). *Multivariate Statistical Inference and Applications*. Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., NY.

Sayyadian, M., Lee, Y., Doan, A., & Rosenthal, A. (2005). Tuning schema matching software using synthetic scenarios. In *proc. 31st International Conference on Very Large Data Bases (VLDB 05)*, pp. 994–1005, Trondheim, Norway.

Seddiqui, M.H., & Aono, M. (2009). An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Journal of Web Semantics, 7(4):344-356.*

Shvaiko, P., & Euzenat, J. (2008). Ten challenges for ontology matching. In *proc. International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE 08)*, pp. 1164–1182, Monterrey, Mexico.

Silva, N., Rocha, J., & Cardoso, J. (2003). E-business interoperability through ontology semantic mapping. In *proc. the Processes and Foundations for Virtual Organizations*, pp. 315–322, Lugano, Switzerland.

Singh, M.P., & Huhns, M.N. (2005). *Service-oriented computing - semantics, processes, agents (1st ed.)*. Wiley, Chichester, England: England Press.

Soh, L.K. (2002). Multiagent distributed ontology learning. In *proc. Working Notes of the second AAMAS OAS Workshop*, Bologna, Italy.

Song, J., Zhang, W., Xiao, W., Li, G., & Xu, Z. (2005). Ontology-based information retrieval model for the semantic web. In *proc. IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE 05)*, Hong Kong, China.

Strasunskas, D., & Tomassen, S.L. (2008). Empirical insights on a value of ontology quality in ontology-driven Web search. In *proc. On the Move to Meaningful Internet Systems (OTM 2008)*, Monterrey, Mexico.

Swoogle Site (2010), http://swoogle.umbc.edu/, March 2010.

Tan, P.S., Goh, A.E.S., & Lee, S.S.G. (2008). A context model for b2b collaborations. In proc. 2008 IEEE International Conference on Services Computing (SCC 08), pp. 108–115, Honolulu, Hawaii, USA.

Tan, P.S., Goh, A.E.S., & Lee, S.S.G. (2009). Context information support for b2b collaboration. *International Journal of Web Engineering and Technology, 5(2):214–245*.

Tijerino, Y., Embley, D., Lonsdale, D., Ding, Y., & Nagy, G. (2005). Towards ontology generation from tables. *World Wide Web: Internet and Web Information Systems*, *8(3):261-285*.

Todorov, K., & Geibel, P. (2008). Ontology mapping via structural and instance-based similarity measures. In *proc. the Third International Workshop on Ontology Matching (OM 08)*, Karlsruhe, Germany.

Wang, S., Isaac, A., Meij, L., & Schlobach, S. (2007). Multi-concept alignment and evaluation. In *proc. the Second International Workshop on Ontology Matching (OM 07)*, Busan, Korea.

Williams, A.B., & Tsatsoulis, C. (1999). Diverse Web ontologies: what intelligent agents must teach to each other. In *proc. AAAI Spring Symposium on Intelligent Agents in Cyberspace*, Stanford University.

Yan, Y., Osadciw, L.A., & Chen, P. (2008). Confidence interval of feature number selection for face recognition. *Journal of Electronic Imaging*, 17(01), 011002.

KEY TERMS & DEFINITIONS

Electronic Business: commonly referred to as "eBusiness" or "e-business," can be regarded as any business process that relies on an automated information system, which typically incorporates Web-based technologies. Electronic business includes a wide range of online business activities for products and/or services. In most cases electronic business is associated with buying and selling over the Internet, or conducting

transactions involving the transfer of ownership or rights to use goods or services through a computer-mediated network.

Service Agent: a software agent (i.e., a piece of software) that acts on behalf of a user or other programs in the Services Computing environment, which is an emerging cross discipline that covers the science and technology of leveraging computing and information technology to model, create, operate, and manage business services.

Ontology: a computational model of some portion or domain of the world. The model describes the semantics of the terms used in the domain. Ontology is often captured in some form of a semantic network, i.e., a graph whose nodes are concepts or individual objects and whose arcs represent relationships or associations among the concepts. The semantic network is augmented by properties and attributes, constraints, functions, and rules, which govern the behavior of the concepts.

Ontology Heterogeneity: an inherent characteristic of ontologies developed by different parties for the same (or similar) domains. The heterogeneous semantics may occur in two ways. (1) Different ontologies could use different terminologies to describe the same conceptual model. That is, different terms could be used for the same concept, or an identical term could be adopted for different concepts. (2) Even if two ontologies use the same name for a concept, the associated properties and the relationships with other concepts are most likely to be different.

Ontology Matching: also known as "Ontology Alignment," or "Ontology Mapping." It is the process of determining correspondences between concepts from heterogeneous ontologies (often designed by distributed parties). Such correspondences include many relationships, for example, *equivalentWith*, *subClassOf*, *superClassOf*, and *siblings*.

Artificial Neural Network (ANN): a mathematical or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. Consisting of an interconnected group of artificial neurons, an ANN processes information via a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Generally speaking, ANNs are non-linear statistical data modeling tools, and can be used to model complex relationships between inputs and outputs, or to find patterns in data.

Agglomerative Clustering: a special category of clustering, which is the assignment of a set of observations into subsets (i.e., clusters) so that observations in the same cluster are similar in some sense. Agglomerative clustering belongs to the hierarchical clustering that creates a hierarchy of clusters represented in a tree structure. Different from divisive clustering, agglomerative clustering starts at the leaves and successively merges clusters together.

Context: while an ontology is regarded as an explicit encoding of a domain model that may be shared and reused, a context can be viewed as an explicit encoding of a domain model that is expected to be local and may contain one party's subjective view of the

domain. In other words, context refers to the conditions, constraints, and circumstances that are relevant to the conceptual model of interest. Both contexts and ontologies play a crucial role in knowledge representation and reasoning.

Confidence Interval: a single observation of a random interval, calculated from a random sample by a given procedure, so that the probability that the interval contains an unknown population parameter θ is $(1-\alpha)$, which is also known as the confidence level or confidence coefficient.

Confidence Interval Estimate: a particular kind of interval estimate of a population parameter θ. Instead of estimating the parameter by a single value, an interval likely to include the parameter θ is given. Confidence intervals are used to indicate the reliability of an estimate. How likely the interval is to contain the parameter is determined by the confidence level or confidence coefficient, which is usually expressed as a percentage, i.e., $(1-\alpha)$.