DEGREE OF DOCTOR OF PHILOSOPHY IN
COMPUTER ENGINEERING AND SCIENCE

DOCTORATE SCHOOL IN
INFORMATION AND COMMUNICATION TECHNOLOGIES

XXIII Cycle

UNIVERSITY OF MODENA AND REGGIO EMILIA

INFORMATION ENGINEERING DEPARTMENT

Ph.D. DISSERTATION

Label Normalization and Lexical Annotation for Schema and Ontology Matching

Candidate:
Serena SORRENTINO
Advisor:
Prof.Sonia BERGAMASCHI
Co-Advisor:
Prof. Sanda HARABAGIU
The Director of the School:
Prof. Sonia BERGAMASCHI

DOTTORATO DI RICERCA IN
COMPUTER ENGINEERING AND SCIENCE

SCUOLA DI DOTTORATO IN
INFORMATION AND COMMUNICATION TECHNOLOGIES

XXIII Ciclo

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

TESI PER IL CONSEGUIMENTO DEL TITOLO DI DOTTORE DI RICERCA

Label Normalization and Lexical Annotation for Schema and Ontology Matching

Tesi di:
Serena SORRENTINO
Relatore:
Prof. Sonia BERGAMASCHI
Co-Relatore:
Prof. Sanda HARABAGIU
Il Direttore:
Prof. Sonia BERGAMASCHI

Keywords:

Schema and Ontology Matching
Lexical Annotation
Word Sense Disambiguation
Schema Label Normalization
Probabilistic Lexical Annotation

Abstract

Schema matching is the problem of finding relationships among concepts across heterogeneous data sources (heterogeneous in format and semantics). The "hidden meaning" associated with *schema labels* (i.e. class and attribute names) may be explicated by *Lexical Annotation* (i.e. annotation with respect to an authoritative lexical thesaurus).

The goal of this thesis is to propose, and experimentally evaluate automatic and semi-automatic methods performing lexical annotation of schema labels. In this way, we may add sharable semantics to legacy data sources. Moreover, annotated labels are a powerful means in order to discover *Lexical Relationships* among structured and semi-structured data sources. Original methods to automatically extract lexical relationships have been developed and their affectiveness for automatic schema matching shown.

In this thesis, a novel automatic lexical annotation method called CWSD (Combined Word Sense Disambiguation), which assigns to each schema label one or more meanings with respect to the well known lexical database WordNet, is introduced.

As the performance of automatic/semi-automatic lexical annotation methods on real-world schemata suffers from the abundance of *non-dictionary words* such as compound nouns, abbreviations, and acronyms, a novel method to perform *Schema Label Normalization* which increases the number of annotable labels has been developed and presented in the thesis.

Moreover, to cope with the uncertainty intrinsic in systems performing "on-the-fly" (i.e. in a fully automatic way) schema matching, I present a method called PWSD (Probabilistic Word Sense Disambiguation) performing probabilistic lexical annotation that allows to discover probabilistic lexical relationships among elements of different schemata. The effectiveness of all the developed methods is proved by experimental results.

Sommario

Schema matching è il processo di identificazione delle relazioni che sussistono fra concetti di sorgenti dati eterogenee (eterogenee sia nella struttura che nella semantica). I significati implicitamente associati alle etichette di uno schema (cioè i nomi delle sue classi e dei suoi attributi), possono essere esplicitati attraverso il processo di *Annotazione Lessicale* (cioè l'annotazione rispetto ad un thesaurus lessicale di riferimento).

L'obbiettivo di questa tesi è di proporre e valutare sperimentalmente metodi automatici e semi-automatici per l'annotazione lessicale che consentano di arricchire semanticamente le sorgenti dati. A partire dalle annotazioni è possibile scoprire *Relazioni Lessicali* fra sorgenti dati strutturate o semi-strutturate. All'interno della tesi vengono proposti metodi innovativi ed automatici per scoperta di relazioni lessicali, e viene descritta la loro efficacia al fine della scoperta dei matching. Inoltre, viene proposto e descritto un metodo chiamato CWSD (Combined Word Sense Disambiguation) il quale associa a ciascuna etichetta di uno schema uno o più significati utilizzando come risorsa lessicale di riferimento il database lessicale WordNet.

Le prestazioni dei metodi di annotazione automatici o semi-automatici quando applicati su schemi dati reali, soffrono della presenza di *non-dictionary words* (cioè di termini non presenti all'interno della risorsa lessicale di riferimento) che possono essere, nomi composti, abbreviazioni o acronimi. Allo scopo di migliorare le prestazioni di tali metodi, viene descritto e proposto un metodo di *normalizzazione delle etichette dello schema* che incrementa il numero delle etichette annotabili.

Infine, nella tesi viene introdotto un metodo di annotazione probabilistica, chiamato PWSD (Probabilistic Word Sense Disambiguation), allo scopo di gestire l'incertezza intrinseca ai sistemi di schema matching "on-the-fly" (cioè sistemi dove i matching vengono scoperti in maniera completamente automatica). A partire dalle annotazioni lessicali probabilistiche, PWSD consente di scoprire fra gli elementi di più schemi, relazioni lessicali probabilistiche.

Tutti i metodi proposti in questa tesi sono stati implementati e valutati sperimentalmente.

Acknowledgments

I owe my deepest gratitude to my supervisor, Professor Sonia Bergamaschi, for her precious support and guide during my Ph.D.

I would like to convey my sincerest gratitude to the Ph.D. and fellow research Laura Po and the Ph.D. student Maciej Gawinecki co-writers of many of my papers, it was great to collaborate with you both. I sincerely thank all the members of the DBGROUP at the University of Modena and Reggio Emilia and in particular my colleagues of the laboratory (in alphabetic order) Francesco Guerra, Mirko Orsini, Laura Po, Silvia Rota and Antonio Sala for creating such a great friendship in the laboratory and during the University life.

I gratefully and sincerely thank the co-tutor of this thesis Professor Sanda Harabagiu for the opportunity she gave me to spend a profitable period abroad at UTD – University of Texas at Dallas. I gratefully acknowledge to the colleagues at UTD Bryan and Kirk for their advice and their willingness to share their bright thoughts with me, which were very fruitful for shaping up my ideas and research.

To my roommate Beatrice and to Gabi, Brianna, Murat, Cristina and Gabriel for their support and for the beautiful time spent together in Dallas.

My parents deserve a special thank for their invaluable support and help throughout all my studies. Francesca and Chiara, thanks for being supportive and caring sisters. I gratefully and sincerely thank Miky, Dany, Alle, Lu, Marce, and all my friends to support and encouragement made during these three years.

Finally, I would like to thank everybody who was important to the successful realization of thesis, as well as expressing my apology that I could not mention personally one by one.

The work on this thesis was partially supported by the MUR FIRB Network Peer for Business (NP4B) project (2006)[1], by the IST FP6 STREP project STASIS[2](2006), and by the "Searching for a needle in mountains of data!" project funded by the Fondazione Cassa di Risparmio di Modena within the Bando di Ricerca Internazionale (2008)[3].

[1] http://www.dbgroup.unimo.it/nep4b

[2] http://www.stasis-project.net

[3] http://www.dbgroup.unimo.it/keymantic

Contents

List of Figures

Chapter 1

Introduction

The advent of the Semantic Web [Shadbolt et al., 2006] associated with the progress of Information and Communication Technologies has made available a large amount of information resources for different database and web site applications. The number of different information resources is rapidly increasing and the *semantic integration* of heterogeneous data sources (i.e. the process of interrelating information from diverse, distributed data sources by exploiting the semantics associated to each source) is becoming a crucial challenge to build large-scale information systems.

Many researchers agree that one of the major bottleneck in semantic integration is the *schema matching problem*. Schema matching is the process that takes two heterogeneous schemata as input and produces as output a set semantic correspondences (called *mappings*) among the schema elements/attributes. Matching techniques are important in many applications, such as ontology integration, data integration, or data warehouse [Bergamaschi et al., 2011a, Euzenat and Shvaiko, 2007, Inmon, 1997]. These applications are characterized by heterogeneous data models that are analyzed and matched either manually or semi-automatically.

Ontology and schema matching are not a new research area, and several approaches and tools have been proposed in the literature since the 1970s [Islam et al., 2008]. Much progress has been made, and nowadays ontology and schema matching have become a vibrant research area. However, despite its pervasiveness, today no satisfactory solution has yet been found: schema and ontology matching is still largely conducted by hand, supported by a graphical user interface that allows the user to interact with the matching system. Obviously, manual matching is expensive, laborious, error-prone, and not scalable. Hence, the development of automatic or semi-automatic methods to assist in the matching process has become crucial for the success of a wide variety of applications.

Automatic or semi-automatic ontology and schema matching has to deal with

problems arising from the heterogeneity of data sources. We distinguish two main types of heterogeneity: *Structural Heterogeneity* and *Semantic Heterogeneity*.

Structural heterogeneity refers to differences among local definitions, such as attribute types, formats, or models. These differences can be relatively easy to solve.

On the other end, semantic heterogeneity refers to differences in the meaning of schema elements. Dealing with semantic heterogeneity in schema matching is a difficult task for different reasons: two elements in two local data sources can have the same intended meaning, but different labels (e.g., "Last Name" and "Surname"); thus, it should be recognized that these two elements actually refer to the same concept. Moreover, two schema elements in two schemata might be named by using the same label, but with different intended meanings (e.g., "peer" has a sense "equal" as well as another sense "member of nobility") [Batini and Lenzerini, 1984].

Another problem is the presence of *complex mappings* between elements of two schemata [Dhamankar et al., 2004]: in contrast with *simple mappings* which represent *one-to-one* relationships (e.g. "address = location"), complex mappings imply *one-to-many* or *many-to-one* relationships between the element of two source schemata (e.g. "name = (firstname, lastname)").

In the literature, it is possible to identify two main schema matching approaches, according to the different information on which the rely on, to cope with semantic heterogeneity [Rahm and Bernstein, 2001]: *Schema-Based* and *Instance-Based* matching. Schema-based matchers consider schema information including the usual properties of schema elements, such as class and attribute labels, data type, relationship types (part-of, is-a, etc.) and schema structure. On the contrary, instance-based matchers only exploit the instance information and depend on the content overlap between two data sources. The main drawback of the latter approach is that instance analysis is computationally a heavy task since it involves a great number of elements [Bergamaschi et al., 2011a]. Moreover, there are several situations where data instances are not available due to security reasons or restricted license authorizations [Clifton et al., 1997].

In this thesis, we focus on schema-based solutions, i.e., matching systems exploiting intensional information, not instance data. However, as it will be described in Chapter 7, when available, instances can be useful to solve those cases where the schema description is poor or/and not informative about the content of the sources.

The DBGroup of the University of Modena and Reggio Emilia (of which I am a member), has developed a Data Integration System called MOMIS[1](Mediator EnvirOnment for Multiple Information Sources), which performs schema match-

[1]See `http://www.dbgroup.unimore.it` for references about the MOMIS project.

ing by exploiting only the schema level information. A key intuition of the DB-Group, to solve the semantic heterogeneity problem, is to look for similarities between schema/ontology labels (i.e. element names, also called terms from now on) by using lexical driven techniques and semantic similarity measures. Starting from the "hidden meaning" associated with schema labels, it is possible to discover semantic correspondences among the elements of different schemata. In order to explicit the "meaning" of schema labels a method for *lexical annotation* (i.e. the explicit assignment of meanings to a label with respect to a thesaurus or a reference lexical database) has to be devised.

In MOMIS a manual method to perform lexical annotation w.r.t. the lexical database WordNet [Miller et al., 1990] has been implemented[2]. However, manual lexical annotation is a time-consuming and not scalable task in large sets of data sources.

For this reason, I focused my research activity during the Ph.D. on the study and development of automatic/semi-automatic lexical annotation methods (annotation methods on the following). To perform automatic/semi-automatic lexical annotation, we need to employ Word Sense Disambiguation (WSD) algorithms which permit to identify the meaning of terms in a context by exploiting computational techniques.

During the first year of my Ph.D. I developed (in collaboration with the DBGroup) a method called CWSD (Combined Word Sense Disambiguation), described in [Bergamaschi et al., 2007c, Bergamaschi et al., 2007d]. CWSD is a method and a tool to perform semi-automatic annotation of structured and semi-structured data sources, which is composed by two main WSD algorithms: SD (Structural Disambiguation) and WND (WordNet domain Disambiguation). CWSD associates more than one meaning to a term and thus differs from the traditional WSD approaches. Instead of being targeted to textual data sources like most of the traditional WSD algorithms, CWSD exploits the structured knowledge of the data sources together with the lexical knowledge supplied by the lexical thesaurus WordNet and its extension WordNet Domains[3].

The strength of a thesaurus, like WordNet, is the presence of a wide network of semantic relationships among word meanings, thus providing a corresponding inferred semantic network of semantic relationships among the labels of different schemata. Its weakness, is that it does not cover different domains of knowledge with the same detail and many domain-dependent words, or *non-dictionary words*, may not be present in it. Non-dictionary words include Compound Nouns (CNs) (e.g., "company address"), abbreviations (e.g., "QTY"), and

[2]With the exclusion of an automatic annotation algorithm which trivially assigns to each label the most frequent meaning.

[3]`http://wndomains.fbk.eu/`

acronyms (e.g., WSD-Word Sense Disambiguation).

The evaluation of CWSD applied on real data sources has highlighted how the result of semi-automatic lexical annotation is strongly affected by the presence of such non-dictionary words in schemata. For this reason, I worked during the second and third year of my Ph.D., on the study and development of a method to expand abbreviations and to semantically "interpret" CNs, described in [Sorrentino et al., 2009, Sorrentino et al., 2010, Sorrentino et al., 2011]. In particular, my research was focused on the study of a method for annotating CNs by enriching WordNet with new meanings (synsets in WordNet) presented in [Sorrentino and Bergamaschi, 2009, Beneventano et al., 2009a]. In the following, I will refer to this method as *Schema Label Normalization*. Schema label normalization helps in the identification of similarities between labels coming from different data sources, thus improving schema matching accuracy.

The use of semi-automatic techniques often requires the human intervention in order to validate or reject the automatic results. However, while this is possible in application scenarios such as data warehouse or data integration with a limited number of sources, it is not feasible in a dynamic context, such as Web source interconnection, which involves a large number of sources dynamically growing. In these cases, the matching needs to be automatically extracted and therefore approximate. In recent years, these observations have brought to the development of *pay-as-you-go* approaches, where the schema matching is performed *on-the-fly*. In these approaches the system starts with very few (or approximate) semantic mappings and these mappings are improved over time as deemed necessary [Sarma et al., 2008].

In this dynamic environment, we can still apply conventional schema matching techniques, but we need to enrich the traditional solutions with the notion of *uncertainty*.

To cope with this problem, the semi-automatic lexical annotation method (CWSD) has been extended: I have taken part to the study and development of a probabilistic lexical annotation method called PWSD (Probabilistic Word Sense Disambiguation), which deals with the uncertainty during the annotation process by associating to each annotation a probability value representing the reliability of the annotation itself [Po et al., 2009, Po and Sorrentino, 2011, Bergamaschi et al., 2009a, Bergamaschi et al., 2009b].

All the methods presented in this thesis have been applied in the context of the MOMIS Data Integration System. However, they can be applied in general in the contexts of schema and ontology matching. Although, there are some differences between schema and ontology matching (alignment) problems (see Chapter 2 for more details), we agree with [Shvaiko and Euzenat, 2005] that the techniques developed for each of them can be of a mutual benefit. Therefore, in this thesis, we discuss schema and ontology matching as the same problem.

Outline of the Thesis
The rest of this thesis is organized as follows:

- *Chapter 2* is dedicated to the definition of the ontology and schema matching problem and to the description of the role of lexical annotation during the matching task.
- In *Chapter 3*, a general description of the MOMIS Data Integration System is provided. Moreover, an early effort to obtain an effective approach for Ontology-Based Data Integration is presented, which is builded on the combination of the MOMIS and STASIS (SofTware for Ambient Semantic Interoperable Services)[4] systems.
- *Chapter 4* focuses on semi-automatic lexical annotation techniques. In particular, I propose the CWSD semi-automatic method in order to annotate structured and semi-structured data sources with respect to WordNet.
- In *Chapter 5* a method to perform schema label normalization is presented. Moreover, describe a stand-alone tool called NORMS (NORMalizer of Schemata) which implements the label normalization functionalities and provides a GUI that supports the designer during the normalization process allowing her/him to enhance the automatic results by correcting potential errors.
- In *Chapter 6*, the CWSD method is extended by introducing the concept of uncertainty and by proposing a probabilistic annotation method called PWSD, which combines, in a probabilistic way, a set of WSD algorithms. PWSD has been implemented in a tool called ALA (Automatic Lexical Annotator) integrated within the MOMIS system.
- Finally, in *Chapter 7*, I make some concluding remarks and discuss the future work.

[4] `http://www.stasis-project.net`

Chapter 2

The Role of Lexical Annotation in Schema and Ontology Matching

In the recent years, the problem of semantic heterogeneity among structured and semi-structured data sources has received much attention by the databases and information integration research communities. This research ranges from techniques for matching database schemata and for aligning ontologies in the context of the Semantic Web.

Several methods and tools developed to address these problems rely, in different ways, on the use of lexical knowledge. The reason is simple: beyond the syntactic and semantic heterogeneity of ontologies and schemata, their elements and properties are labeled by using natural language terms or combination of terms: these labels represent useful (but often implicit) information about the intended meaning and use of the ontology/schema under construction. It is more evident in the case of ontologies and less evident for schemata of legacy information systems.

Once the semantics associated with schema labels (i.e. class and attribute names) is explicated, it is possible to discover semantic correspondences among the elements of different ontologies/schemata. Therefore, it should not come as a surprise that a large number of matching techniques include some lexical resource (e.g., thesauri or dictionaries) as a component, and use them in some intermediate step to *annotate* schema elements and ontology classes/properties with respect to these lexical resources. *Lexical annotation*, denoting such annotation, is a critical and crucial task to develop smart methods for schema/ontology matching.

The goal of this chapter is to discuss the major contributes and approaches to schema/ontology matching proposed in the literature by the ontology and database communities and to provide readers with pointers to the sources for additional information. Moreover, the concept of lexical annotation of schema and ontology labels is introduced.

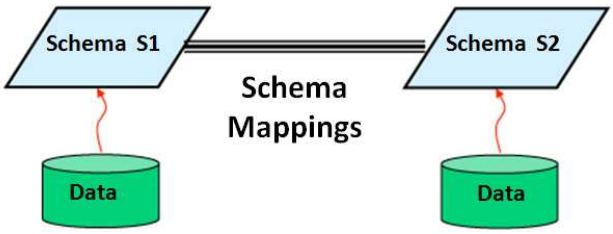

Figure 2.1: Mappings between two schemata.

The rest of the chapter is organized as follows: Section 2.1 defines the concepts of schema and ontology matching, its applications and classifications; Section 2.2 introduces lexical annotation and its role during the matching process; Section 2.4 describes the open problems in the context of lexical annotation and schema matching that will be addressed in the next chapters. Finally, a formal definition of the performance measures used to evaluate the proposed methods in this thesis is provided in Section 2.3.

2.1 Schema Matching

The core of the semantic heterogeneity problem is a process called *schema matching* [Bergamaschi et al., 2011a]. Schema matching has been the focus of research since the 1970s in the artificial intelligence, databases, and knowledge representation communities [Islam et al., 2008].

Definition 1 (Schema Matching). *Schema matching is a process that takes two heterogeneous schemata (e.g., $S1$ and $S2$) as input and produces as output a set of* mappings *(as shown in Figure 2.1). Each mapping indicates that certain elements of a schema $S1$ are related to certain elements of the schema $S2$. Mappings may be obtained by using a set of semantic correspondences (e.g., location = area) between different schemata, as they capture the semantic relationships between concepts.*

Similarly, ontology matching has to deal with multiple, distributed and evolving ontologies. Ontologies can be viewed as schemata for knowledge bases [Shvaiko and Euzenat, 2004]; therefore, techniques developed for schema matching in the great majority of the cases may be applied in the ontology matching context.

A schema is a conceptualization of a domain with a model: Entity-Relationship (ER) model, Object-Oriented (OO) model, XML/XMLSchema, or an ontology graph. In each case, there is a natural correspondence between the building blocks of the representation and the notions of element and structure:

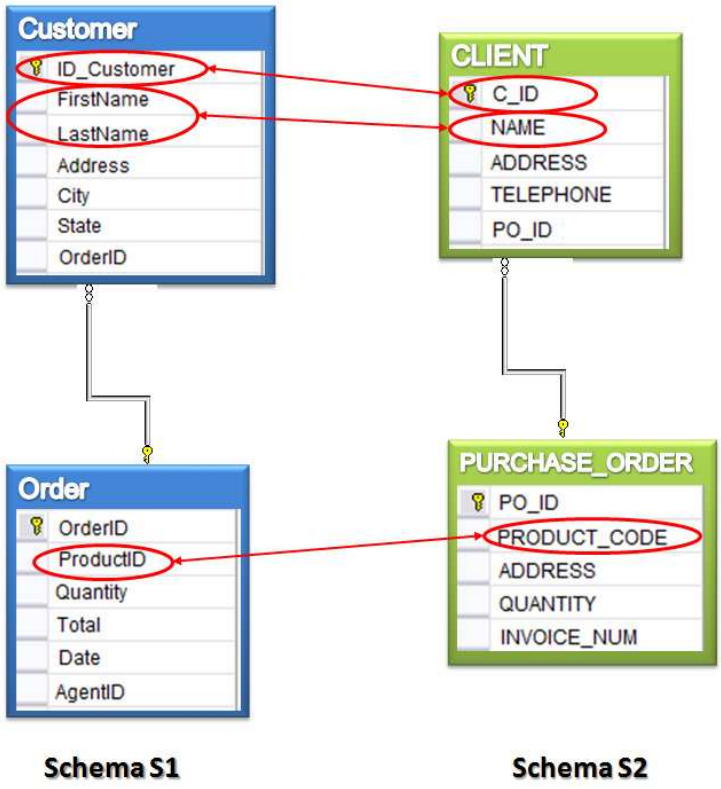

Figure 2.2: Example of mappings among two schemata.

entities and relationships in ER models; objects and relationships in OO models; elements, subelements, and IDREFs in XML; and nodes and edges in ontologies [Rahm and Bernstein, 2001]. In this thesis, I assume the simplified definition proposed in [Rahm and Bernstein, 2001]:

Definition 2 (Schema) *A schema is a set of elements connected by some structure.*

As a consequence, in the following, I will refer to both schema and ontology matching simply as schema matching.

A simple example of schema matching is described in the following: let us suppose that we need to merge two corporations, both of which need to consolidate their relational databases. In this integration scenario,to matching of relational schemata is required [Rahm and Bernstein, 2001]. Let us suppose, for example, that we have to integrate the very simple schemata $S1$ and $S2$ shown in Figure 2.2. Different schemata may use different labels to express the same concept: for instance, as it is shown in Figure 2.2, the "product code" field in schema $S1$ might be represented with the label "productID" in schema $S2$.

Each mapping specifies that certain elements of the table "Customer" logically correspond to (i.e., match) certain elements of the table "Client", where the semantics of the correspondence is expressed by the mapping expression of a mapping element. An example of mapping between the table "Customer" and "Client" is the element relating "Customer.CustomerID" to "Client.C_ID" with the mapping expression "Cust.C_ID = Customer.CustomerID". Another example is the mapping with the expression "Concatenate(Customer.FirstName, Customer.LastName) = Client.Name" describing a mapping between two "Customer" elements and one "Client" element (as shown in Figure).

2.1.1 Schema Matching Application Areas

Schema matching is a prerequisite in many applications, including ontology matching, data integration, and data warehouse [Euzenat and Shvaiko, 2007]. In this section, we describe these three well-known application scenarios where the schema matching problem has been studied for a long time. A fully and complete description of the different schema matching applications is out of the scope of this thesis, and for sake of simplicity we reported only the ones that represent the direct contexts of the methods proposed in this thesis. For further details, please refer to [Euzenat and Shvaiko, 2007, Bergamaschi et al., 2011a].

Data Integration. Data integration is a long-standing research problem and continues to be a challenge in practice.

Definition 3 (Data Integration) *Given a set of two or more source schemata (describing data in a common domain), the goal of a data integration system is the construction of a unified* global view *(in the following also called* global schema *or* mediator*) starting from a set of independently developed schemata, with different structures, terminologies and semantics.*

The data sources contain the real data, while the global view provides a reconciled, integrated, and virtual view of the underlying sources, thus offering a way to deal with the heterogeneity in the sources (see Figure 2.3).

Moreover, the unified view provides a single access point against which user queries are posed to access to the set of data sources. To this goal, data integration systems often provide a uniform query interface to a multitude of data sources. The source databases are interfaced with wrappers if needed, where a wrapper is a format translator that depends by the format of the data source. The general architecture of a data integration system is shown in Figure 2.4.

To answer user queries, integration systems have to reformulate a single query in terms of a set of queries over the original data sources. To do that, such systems

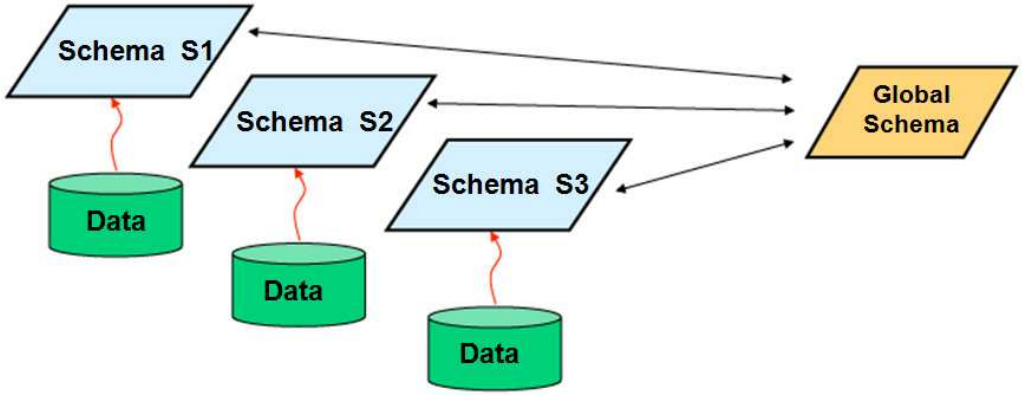

Figure 2.3: Schema matching in Data Integration.

employ a set of semantic matches between the mediated schema and the data source schemata which specify how the local data have to be transformed into the integrated schema.

Modeling the relationships between the sources and the global schema is, therefore, a crucial aspect. Three basic approaches have been proposed to this purpose: (1) *GAV (Global-As-View)* [Ullman, 2000], where "Global" refers to the global/mediated schema; (2) *LAV (Local-As-View)* [Halevy, 2001], where "Local" refers to the local sources/databases; and (3) *GLAV(Global-Local-As-View)* [Friedman et al., 1999].

The GAV approach is based on the idea that the content of each element of the Global Schema should be characterized in terms of a view over the data sources. In this case, the mapping explicitly tells the system how to retrieve the data. Queries are processed by means of unfolding, i.e., by expanding the atoms according to their definitions (so as to come up with local schema relationships). This idea is effective whenever the data integration system is based on a set of sources that is stable; it is possible to say that the GAV approach favors the system in carrying out query processing, because it tells the system how to use the sources to retrieve data. However, extending the system with a new source may be a problem: the new source may have an impact on the definition of various elements of the Global Schema, whose associated views need to be redefined [Lenzerini, 2002].

On the contrary, the LAV approach requires the Global Schema to be specified independently from the data sources, and the relationships between the Global Schema and the sources are established by defining every source as a view over the Global Schema. LAV is based on the idea that the content of each source should be characterized in terms of a view over the Global Schema. Queries are processed by means of an inference mechanism that re-expresses the atoms of the Global Schema in terms of atoms of the local schemata. This approach

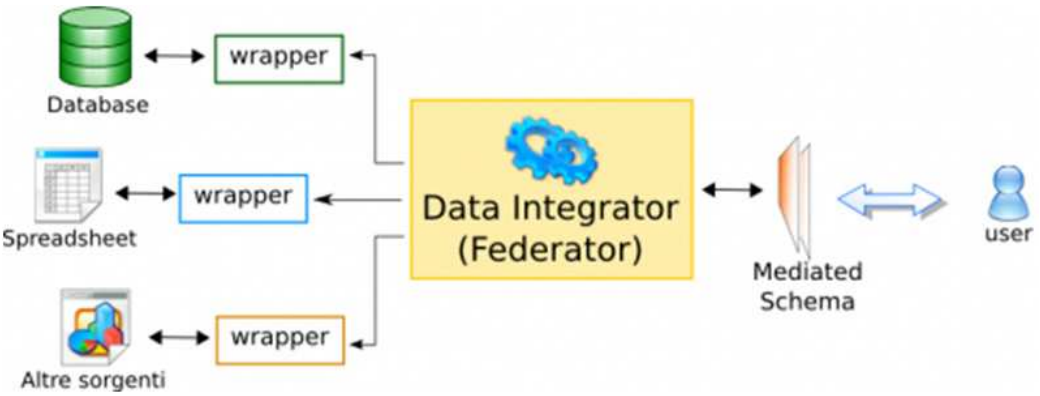

Figure 2.4: Data integration architecture.

is effective whenever the data integration system is based on a Global Schema that is stable and well-established in the organization. A notable case of this type is when the data integration system is based on an enterprise model, or an ontology. The LAV approach favors the extensibility of the system: adding a new source simply means enriching the mapping with a new assertion, without other changes [Lenzerini, 2002].

Finally, the GLAV approach is a mix of the previous approaches: it can be considered as a variation of the LAV approach that allows the head of the view definition to contain any query on the local schemata [Friedman et al., 1999].

The work of this thesis, as it will be described in the following chapters, is in the context of schema matching in data integration. In particular, the proposed methods are described as part of the MOMIS Data Integration System (see Chapter 3) which follows a GAV approach. A number of Data Integration system have been developed in the literature, for more details see [Bergamaschi and Maurino, 2009].

Data Warehouse. In [Inmon, 1997] the topic of data warehouse is defined as:

Definition 4 (Data Warehouse) *A Data Warehouse is a subject-oriented, integrated, time-variant (temporal), non volatile collection of summary and detailed data, used to support strategic decision-making process for enterprises.*

Figure 2.5 shows the architecture of a data warehouse system: in a data warehouse system, the information from the source databases needs to be extracted, transformed then loaded into the data warehouse. To do that a tool called ETL (Extraction Transformation Loading) is used. ETL involves extracting data from outside sources, transforming it to fit operational needs and loading it into the end target. It performs what is called in a more general way *data exchange*.

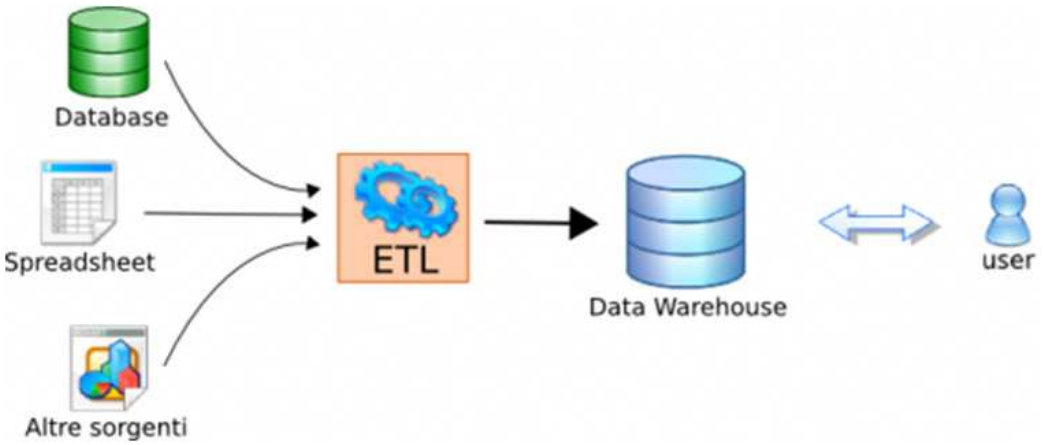

Figure 2.5: Data warehouse architecture

Data exchange has been a recurrent problem that has taken a new significance with the advent of semi-structured data and the resulting need to exchange data between heterogeneous schemata [Kolaitis, 2005, Afrati and Kolaitis, 2008]. The goal of data exchange is to take a given source instance and transform it to a target instance such that it satisfies the specifications of the schema mapping and also "reflects" the given source data as accurately as possible. In data exchange, data structured under the *source schema* have to be restructured and translated into an instance of a *target schema*. Data exchange is used in many tasks that require data to be transferred between existing, independently created applications. Data exchange has been described as the "oldest database problem" and the need for systems supporting data exchange has persisted over the years.

Data integration and data warehouse are two research areas strongly connected which present clear similarities, but also important differences: while in data integration, the Global Schema is a virtual view and the data remain in the sources (see Figure 2.3), in data warehouse the target instances are materialized instances (not a virtual view) (see Figure 2.1). However, both in data integration and data warehouse, schema matching is used to specify the relationships between the schemata involved.

Ontology Matching. An ontology defines concepts used for representing knowledge on the web, e.g., for annotating a picture, specifying a web service interface or expressing the relation between two persons. There are a number of languages for ontologies, both proprietary and standards-based. However, OWL (Web Ontology Language) represent the ontology W3C standard[1]. OWL is a language for making ontological statements, developed as a follow-on from RDF

[1] `http://www.w3.org/TR/owl-ref/`

(Resource Description Framework) and RDFS (RDF Schema)[2].

Thanks to the success of the Web, there was a fast increase in the number of available ontologies. However, these ontologies, even if they describe similar domains, use different terminologies and structures. It is thus necessary to find correspondences between concepts of heterogeneous ontologies.

Definition 5 (Ontology Matching) *Ontology matching, or ontology alignment, is the process of determining alignment (also called correspondences) between concepts of two ontologies.*

Alignments are used for importing data from one ontology to another or for translating queries. Ontology alignment is a foundational problem area for semantic interoperability. Virtually any application that involves multiple ontologies should establish semantic mappings among them, to ensure interoperability [Doan et al., 2004].

Unfortunately, manually specifying such correspondences is time-consuming, error-prone, and clearly not possible on the Web scale. Hence, the development of tools to assist in ontology mapping is crucial to the success of the Semantic Web [Doan et al., 2004].

The problems of schema and ontology matching are strictly connected, even if they present some substantial differences. Database schemata often do not provide explicit semantics for their data: semantics is usually specified explicitly at design-time, and frequently is not becoming a part of a database specification, therefore it is not available. On the contrary, ontologies are logical systems that themselves obey some formal semantics, for example, we can interpret ontology definitions as a set of logical axioms. Moreover, while schema matching is usually performed with the help of techniques trying to guess the meaning encoded in the schemata, ontology matching systems try to exploit knowledge explicitly encoded in the ontologies [Shvaiko and Euzenat, 2005, Giunchiglia et al., 2007]..

Moreover, ontologies may be applied in the context of data integration: the MOMIS system performs data integration by exploiting the semantics provided by a lexical thesaurus, and permits to export the annotated Global Schema as an OWL ontology (see Section 3.1.2). Moreover, an example of ontology-driven data integration is described in Chapter 3. *An ontology-driven approach to data integration* relies on the alignment of the concepts of a global ontology that describe the domain, with the concepts of the ontologies that describe the data in the local databases. Once the alignment between the global ontology and each of the local ontologies is established, users can potentially query hundreds of databases using a single query that hides the underlying heterogeneities. Using this approach, the query phase can be easily extended to a new database by aligning its ontology with the global one.

[2] `http://www.w3.org/RDF/`

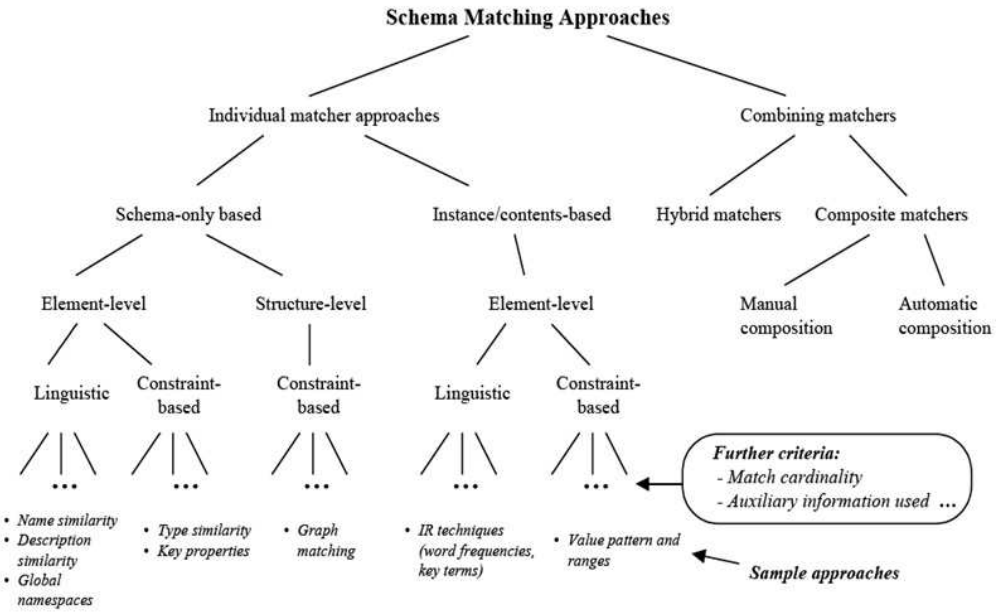

Figure 2.6: Classification of schema matching techniques.

2.1.2 Classification of Schema Matching Techniques

In this section, we briefly describe the main schema matching techniques in the literature and their classification.

Several surveys have been proposed [Shvaiko and Euzenat, 2005, Rahm and Bernstein, 2001, Euzenat and Shvaiko, 2007, Bergamaschi et al., 2011a]. In these surveys, the authors present a general classification of the schema matching approaches and consider various dimensions along which a classification can be elaborated. In this thesis, we consider the schema matching classification proposed in [Rahm and Bernstein, 2001] which distinguishes between *individual matchers* and *combined matchers*.

Individual matchers include: *schema-based matching* and *instance-based matching*, *element* and *structural-level* matchers, *linguistic* and *constraint-based* matchers. Moreover, also the cardinality and the use of external information (like thesauri) is taken into account. Figure 2.6 reports the schema matching technique classification proposed in [Rahm and Bernstein, 2001]. In the following, we briefly describe the main dimensions of this classification.

Individual vs. Combinational. An individual matcher uses a single algorithm to perform the match. Combinational matchers can be one of two types: (1) hybrid matchers using multiple criteria to perform the matching, and (2) composite matchers running independent match algorithms on two schemata and combining

the results.

Schema-based vs Instance. Schema-based matchers consider only schema information, not instance data. The available information includes the usual properties of schema elements, such as name, description, data type, relationship types (part-of, is-a, etc.), constraints, and schema structure. In general, a matcher will find multiple match candidates. For each candidate, it is customary to estimate the degree of similarity by a normalized numeric value in the range [0-1], in order to identify the best match candidates. Schema-based matching algorithms work well when naming conventions are standardized and there is a general consensus about how the data should be organized. Examples of schema-based matching system are OMEN (Ontology Mapping ENhancer) [Mitra et al., 2005], S-Match [Giunchiglia et al., 2005], Similarity Flooding [Melnik et al., 2002], COMA (COmbination of MAtching algorithms) [Do and Rahm, 2002], Cupid [Madhavan et al., 2001], H-Match [Castano et al., 2003], and the MOMIS data integration system [Beneventano et al., 2003a, Bergamaschi et al., 1999].

Pure instance-based matchers usually do not consider the schema information. They either use meta-data and statistics collected from data instances to annotate the schema or to directly find correlated schema elements, for example, using machine learning techniques. Instance-based matching requires that the instances of the schemata to be matched have common features. The main drawback of this approaches is that there are several situations where data instances are not available due to security reasons or restricted authorizations. Among the instance-based matchers, we can find the well known GLUE [Doan et al., 2004], LSD [Doan et al., 2001], and iMAP [Dhamankar et al., 2004].

Moreover, there are instance-based matchers like Clio [Hernández et al., 2001] or oMAP [Straccia and Troncy, 2005] that are able to exploit both schema and instance information.

Element vs. Structure level. An element-level matcher, for each element of the first schema, determines the matching element in the second input schema. Depending on the matcher type, the match comparison can be based on such properties as name, description, or data type of schema element.

On the contrary, a structure-level matcher compares combinations of elements that appear together in a schema, for example, classes or tables whose attribute sets only match approximately. Structure-level matching requires that the structures of the schemata to be matched have common features. In the ideal case, all components of the structures in the two schemata fully match. Alternatively, only some of the components may be required to match (i.e., a partial structural match). The great majority of the matching systems (S-Match, MOMIS, Cupid,

Clio etc.) exploit both structure and element levels.

Linguistic based. Linguistic matchers use the lexical information provided by the label of an element (also called word, token or name of an element) to find semantically similar schema elements. In linguistic matching, existing algorithms generally combine several methods. Name matching involves: putting the name into a canonical form by stemming and tokenization; comparing equality of names; comparing synonyms and hypernyms using generic and domain-specific thesauri; and matching sub-strings. A common solution used to compute similarity between element names is to use strings matching techniques. Other systems, like MOMIS and H-Match, exploit also the semantic information associated to the labels in order to discover the correspondences among schema elements. *The great majority of current linguistic matching solutions are based on the use of WordNet, a lexical database for English* (see Section 2.2). Linguistic approaches produce a linguistic matching solution more comprehensively and efficiently. However, linguistic matching may produce high similarity scores even though the nodes do not semantically correspond to each other (see Chapter 5), thus we need techniques that can adjust such incorrectness.

Constraint based. A constraint-based matcher uses schema constraints, such as data types and value ranges, uniqueness, required-ness, cardinalities, etc. It might also use intra-schema relationships such as referential integrity. For example, the MOMIS system, exploits the constraints coming from relational schemata (such as Primary and Foreign Key) to infer correspondences among the schema elements. Moreover, it uses the data type constraints in order to validate the discovered correspondences.

Matching Cardinality. Schema matchers differ in the cardinality of the mappings they compute. Some only produce $1:1$ mappings (also called *simple mapping*) between schema elements. Others produce n:1 or 1:n mappings (also called *complex mapping*). Let us consider, for instance, the two schemata shown in Figure 2.2: an example of simple mapping between the table "Customer" and "Client" is the expression "Cust.C_ID = Customer.CustomerID", while the mapping which associates both "Customer.FirstName and Customer.LastName" to the schema element "Client.Name" represents a 2:1 complex mapping. LSD, GLUE and iMAP, for example, exploit domain constraints on the schema. The MOMIS system permits to discover both simple and complex mappings.

Auxiliary information. Schema matchers differ in their use of auxiliary information sources such as dictionaries, thesauri, and input match-mismatch informa-

tion. Systems like MOMIS, S-Match, CUPID, and H-Match, use external thesauri like WordNet during the schema matching process.

As it will be described in the following section, this thesis focuses on schema-based matching techniques which exploit, in particular, the linguistic information associated to schema elements.

2.2 Lexical Annotation: a powerful technique for Schema Matching

Traditional schema-based matching techniques exploit the information provided by schema labels by using simple *name-based techniques*, for instance, "string comparison techniques" where the mapping between two elements is discovered by comparing only the strings of their labels. The main drawback of this approach is that it does not consider the *semantics* associated to schema elements: a schema may contain *synonyms* (when different words are used to name the same entities) and *homonyms* (when the same words is used to name different entities) [Euzenat and Shvaiko, 2007]. Synonyms and homonyms can mislead a linguistic-based matcher as homonyms are similar names that refer to different elements, and synomyns are different names that refer to the same concepts. A schema matcher can reduce the number of wrongly matched candidates by exploiting the semantics associated to schema elements. The semantics can be explicated by discovering the meanings of element labels.

Definition 6 .(Lexical Annotation) *Lexical annotation of a schema label is the explicit assignment of meanings to the label with respect to a thesaurus or a reference lexical database.*

Lexical annotation helps to bridge the ambiguity of the natural language and provides a computational representation of the meaning of a label.

Lexical annotation can be manually executed by a schema designer. However, manual annotation is a very time-consuming and boring task. On the contrary, automatic annotation may drastically reduce the human intervention. To perform automatic annotation, a method for automatic WSD is mandatory.

Definition 7 (WSD). *WSD is the ability of identifying the meaning of words in a context by a computational technique. A WSD algorithm can be summarized as follows: given a set of words a technique is applied which makes use of one or more sources of knowledge to associate the most appropriate sense/senses with words in context.*

While humans are able to easily recognize the meaning of a word on the basis of the context in which it used, for an automatic algorithm it is still a really difficult

task. Nowadays, WSD is considered an Artificial Intelligence (AI) complete problem, that is, a task whose solution is at least as hard as the most difficult problems in AI [Navigli, 2009].

Knowledge sources can vary from text corpora (i.e., plain text collections) (e.g., the British National Corpus[3]) either unlabeled or annotated with word senses, to more structured resources, such as *thesauri* (e.g., WordNet [Miller et al., 1990]), *upper domain ontologies* (e.g., SUMO [Niles and Pease, 2001], DOLCE [Gangemi et al., 2003] or Cyc [Lenat et al., 1990]), and more recently source of knowledge like Wikipedia [Wikipedia, 2004]. Without a reference lexical knowledge, it would be impossible to computationally identify the meaning of words.

Currently the great majority of lexical annotation approaches both in NLP [Brody et al., 2006, Navigli and Velardi, 2005, Mihalcea and Moldovan, 2000, Preiss, 2004] and schema matching [Banek et al., 2008, Bouquet et al., 2003, Shvaiko et al., 2010, Castano et al., 2006], use WordNet as external lexical resource. WordNet is an electronic lexical database for English based on psycholinguistic principles and maintained at Princeton University[4]. It is based on the notion of *synsets* or sets of synonyms. A synset denotes a concept or a sense of a group of terms. Its latest version, WordNet 3.0, contains about 155,000 words organized in over 117,000 synsets. WordNet also provides a *hypernym* (superconcept/subconcept) structure as well as other relationships such as *meronym* (part of relations). It also provides textual descriptions of the concepts (*gloss*) containing definitions and examples. WordNet has been adapted to other languages: for instance, it has been developed an European version called EuroWordNet[5], which contains the WordNet version in languages such as Italian, Spanish etc.

The success of WordNet in schema matching applications is due to a fundamental characteristic: WordNet provides a wide network of semantic relationships (synonyms, hyponyms/hypernyms, meronyms etc.) among meanings. Starting from the lexical annotation of schema labels, thus, it is possible to derive semantic relationships between schema elements on the basis of the relationships defined in the WordNet network between their meanings. These semantic relationships represent the core of a semantic matcher.

Let us consider the example in Figure 2.7. By using traditional string-based techniques, it is not possible to automatically discover that exist an equivalence semantic relationship between the table names “CLIENT” and “CUSTOMER”. On the contrary, by performing automatic WSD, we discover that both the labels are

[3] http://www.natcorp.ox.ac.uk/
[4] http://wordnet.princeton.edu/
[5] http://www.illc.uva.nl/EuroWordNet/

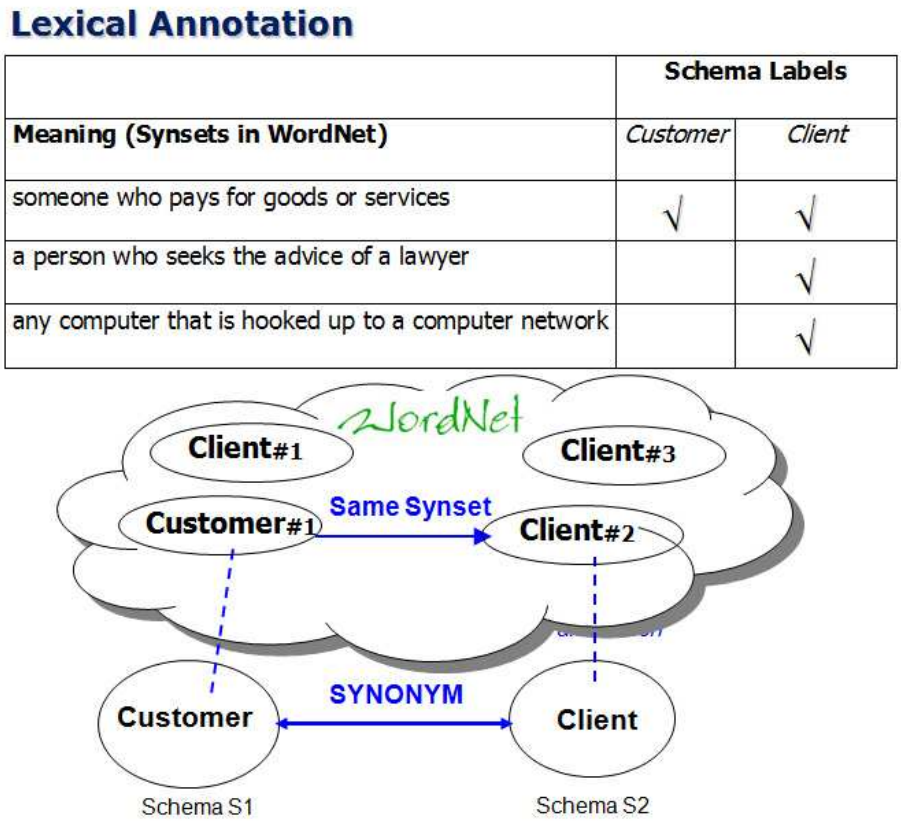

Figure 2.7: Lexical Annotation of the schema labels Customer and Client.

associated in WordNet to the synset "someone who pays for goods or services", and, as a consequence, that there exist an equivalence correspondence between this two concepts.

2.3 Automatic Annotation and Schema Matching: Evaluation Measures

For evaluating the effectiveness of WSD algorithm and schema matching techniques, it is important to understand if they perform well, i.e., if they can help to reduce the manual work required for the lexical annotation and matching tasks.

To show the effectiveness of our methods, it is necessary to demonstrate their application to some real world scenarios or conduct a study using a range of schema or ontology matching tasks. To different matching systems correspond to different evaluations in the literature, with diverse methodolo-

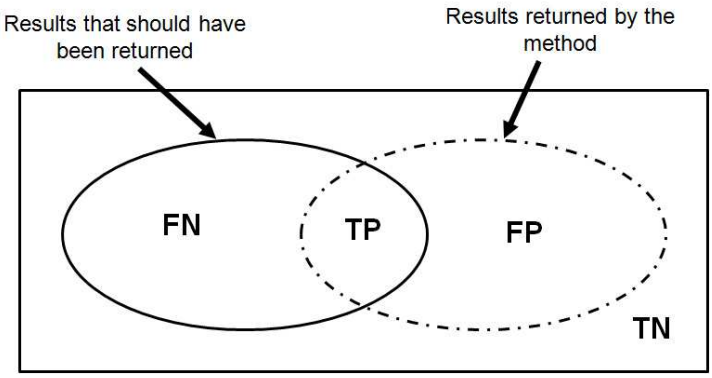

Figure 2.8: Graphic representation of TP, TN, FP, and FN.

gies, and data sets. It makes difficult to compare their effectiveness with respect to the state of the art. In this thesis, the proposed methods are evaluated by adopting the quality measures proposed in [Do et al., 2002], which have been used to asses the effectiveness of several relevant schema matchers [Li and Clifton, 2000, Melnik et al., 2002, Do and Rahm, 2002] and WSD algorithms [Navigli and Velardi, 2005, Navigli, 2009].

For each method, a *gold standard* representing the task performed manually by a designer has been created. Then, we compared the gold standard with the result obtained by using the automatic or semi-automatic methods developed. For each experimental phase, it is necessary to determine: the true positives, i.e. correct results (TP), as well as the false positives (FP), the false negatives (FN).

False negatives are results needed but not automatically identified, while false positives are results falsely proposed by the method. True positive are the correct result that have been correctly returned by the method, while true negatives, are false results, which have also been correctly discarded by the method. Intuitively, both false negatives and false positives reduce the quality of the results. Figure 2.8 shows a set representation of TP, TN, FP, and FN.

Obviously, both results and gold standard, vary on the basis of the task: for example, the result of a WSD algorithm will be different with respect to the result of the semantic correspondences discovery process. In this section, we refer in general to a result as the output of the automatic or semi-automatic method we propose. More details about the result and the gold standard will be given in each correspondent evaluation section, in the different chapters.

In the context of the schema matching and lexical annotation, the following quality measures are computed:

Definition 8 (Precision) *Precision is the number of the correct results returned by the automatic method divided by the total number of results that should have*

been returned. Based on the cardinalities of the TP, FP, and FN sets, Precision is defined as:

$$Precision = \frac{|TP|}{|TP|+|FP|}$$

Definition 9 (Recall) *Recall is the number of the correct results divided by the number of all the results returned by the automatic method. Based on the cardinalities of the TP, FP, and FN sets, Recall is defined as:*

$$Recall = \frac{|TP|}{|FN|+|TP|}$$

Definition 10 (F-Measure) *F-Measure is a weighted average of Precision and Recall. Its score reaches its best value at 1 and worst score at 0. F-Measure is defined as:*

$$F\text{-}Measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Precision and recall originate from the field of information retrieval but they have also been commonly used both in schema matching [Do et al., 2002] and WSD [Navigli, 2009]. In the ideal case, when no false negatives and false positives are returned, we have precision=recall=1. However, neither precision nor recall alone can accurately assess the quality of the results. In particular, recall can easily be maximized at the expense of a poor precision by returning all possible results. On the other side, a high precision can be achieved at the expense of a poor recall by returning only few (correct) results. Hence it is necessary to consider both measures or a combined measure like F-Measure [Do et al., 2002].

2.4 Automatic Annotation and Schema Matching: Open Problems

Automatic annotation and schema matching are considered really difficult tasks. As previously described, various tools and approaches have been proposed. However, several problems still open or have only been partially solved. In this section, we describe the main open problems in automatic annotation and schema matching. The first tree problems will be addressed in this thesis, while the remaining problems represent a future research line (see Chapter 7).

Automatic lexical annotation.
Automatic (or semi-automatic) lexical annotation is a difficult task for several

reasons: (1) human language is ambiguous, so that many words can be interpreted in multiple ways depending on the context in which they occur; (2) the task lends itself to different formalizations due to fundamental questions, like the approach to the representation of a word sense, the granularity of sense inventories, the domain-oriented versus unrestricted nature of texts, and the set of target words to disambiguate; (3) WSD heavily relies on knowledge that is used as lexical reference.

Moreover, even if WordNet is amply utilized as lexical knowledge base, it is too fine grained: it contains many meaning distinctions and there cases where, even a human annotator is not able to select only one right meaning.

In Chapter 4, we address this problem by proposing a method to semi-automatically annotate schema labels by exploiting WordNet associated with the WordNet Domains[6] resource.

Schema Label Normalization.
Another drawback of WordNet is that it does not cover different domains of knowledge with the same level of detail and many domain dependent terms (called *non-dictionary words*) may not be present in it. As a consequence, the result of automatic annotation is strongly affected by the presence of these non-dictionary words in the schemata.

Definition 11 (Non-dictionary Word) *A non-dictionary word is a term which does not have an entry in the lexical database of reference. In this thesis, we recognize the following main kinds of non-dictionary words: compound nouns (e.g., "company address"), abbreviations and acronyms (e.g., "qty"), and domain specific terms (e.g., the biomedical term "aromatase", which is an enzyme involved in the production of estrogen).*

For this reason, a method to expand abbreviations and to enrich WordNet with non-dictionary words is required.

Definition 12 Schema Label Normalization. *Schema label normalization (also called* linguistic normalization *in [Euzenat and Shvaiko, 2007]) is the reduction of the form of each label to some standardized form. With label normalization, we mean the processes of abbreviation expansion and CN interpretation.*

In Chapter 5, we address this problem by introducing a method for *schema label normalization*.

[6] `http://wndomains.fbk.eu/`

Uncertainty in Schema Matching.
An open problem in schema matching is due to the need of computing schema matching in a complete automatic way. As described before, schema matching may find application in different areas. On the basis of the final application, requirements and priorities may change and may be more or less restrictive. For example, in the data warehouse scenario, we need to produce a set of *high confidence* correspondences from the structure of the incoming data to the canonical structure of the target schema. In this case, the "reliability" of the semantic correspondences is fundamental while the time needed to perform the schema matching process becomes a secondary requirement.

As point out from Rahm and Bernstein in [Rahm and Bernstein, 2001], it is not possible to automatically determine all matches between two schemata, primarily because most schemata have some semantics that affects the matching criteria but is not formally expressed or often not even documented. The schema matcher should, therefore, only determine match candidates, which the designer can accept, reject or modify. Furthermore, the designer should be able to specify matches for elements for which the system was unable to find satisfactory match candidates.

However, while this is possible in application scenarios such as data warehouse or data integration of a limited number of sources, it is not feasible in a dynamic context that involves a large number of sources dynamically growing. In this cases, the matching needs to be automatically extracted and therefore approximate. Examples of these scenarios are Web source interconnection, very large database integration etc. Moreover, there are cases where the data are inherently uncertain [Dong et al., 2007]. This uncertain nature represents a challenge in regards to its handling and manipulation of schema matching.

In these scenarios, first of all, it is important that little or no human intervention be required, so we can scale up to integrate a large number of dynamic data sources (i.e. with new ones arriving all the time). Second, they do not need or expect a perfect match in order to provide a useful service.

Although an imperfect matching will degrade the match quality, the results may still be acceptable, especially if the imperfection affects only a few mappings. In the recent years, these observations have brought to the development of *pay-as-you-go* approaches, where the schema matching is performed *on-the-fly*. In these approaches the system starts with very few (or approximate) semantic mappings and these mappings are improved over time as deemed necessary [Sarma et al., 2008].

Despite these different requirements, we can still apply conventional schema matching techniques, but we need to enrich the traditional solutions with the notion of *uncertainty*. Recently, to cope with the uncertainty in schema matching, several methods based on *probabilistic approaches* have been pro-

posed [Nottelmann and Straccia, 2005, Mitra et al., 2005, Sarma et al., 2008]. In these approaches, the discovered mappings are associated with a probability value representing the confidence of the mapping itself. In Chapter 6, we address this problem by proposing an innovative method to discover probabilistic relationships among schema elements.

Non-informative schema labels

As described in Chapter 1, this thesis focus on schema-based matching techniques. However, there are cases where the information provided by the schema is poor or not informative about the content of a data source: for instance, it is a common practice for companies to label the columns of a table by using codes (e.g. "IDS_XF02") which are not informative about the semantics of the schema. We refer to these labels as *non-informative schema labels*. These labels cannot be automatically annotated and they are difficult to disambiguate even for a schema designer.

Moreover, schemata may contain *misleading labels*: for instance, the label "phone", which in WordNet has the meaning of "electronic equipment", is often used in schemata to refer to phone numbers.

In these cases, instance-based matching techniques represent the unique solution to discover semantic mappings. However, as previously described, one of the main drawback of the traditional instance-based techniques, such as Duplicate Detection [Bilke and Naumann, 2005], is that they are computationally expensive especially when we have to deal with a large set of data sources. To address this problem, it is possible to employ RELEVANT [Bergamaschi et al., 2007e, Bergamaschi et al., 2007b], a tool for calculating the "relevant values" among the string values of an attribute. The tool has been conceived for improving the users knowledge of the attributes of database tables: by means of clustering techniques, RELEVANT provides to the designer a synthetic representation of the values of the attribute.

By using RELEVANT, we can enrich the schema description by adding as metadata the relevant values of schema attributes[7]. In this way, it will be possible to exploit the lexical and semantic information provided by instance metadata in order to annotate "non-informative and misleading schema labels".

This problem has not been addressed in this thesis, but represents a future research line as described in Chapter 7.

Schema Matching in a specific domain

Another common problem in schema matching is to deal with semantic heterogeneity of schemata belonging to a specific application domain. Let us suppose,

[7]This is a partial solution as RELEVANT can be applied only on string values.

for instance, that we need to integrate two schemata in the Medical domain. These schemata may contain both generic terms such as “medicine” and “doctor”, and specific terms such as “aromatase” (which is an enzyme involved in the production of estrogen) which are not present in a general lexical resource as WordNet.

In this case, the domain of the schemata has a strong influence on the sense of the schema elements, and then on the result of the semantic schema mapping discovery process. As a consequence, we need of domain specific knowledge which is not provided by general lexical resources such as WordNet.

To the other end, domain-specific lexical resources are rarely available and the cost for their hand creation for every specific domains is prohibitive. Automatically constructed thesauri offer a potential solution. They are usually built by analyzing large document collections, employing statistical methods to identify concepts and semantic relationships. However, the complexity of natural language and the primitive state of language technology means that such thesauri are greatly inferior to manual ones in terms of accuracy and conciseness [Milne et al., 2006].

To the other end, the Semantic Web, currently provides a large set of semantic resource which incorporate specific terms and that have been created by domain-experts. The resulting information is freely available, electronically encoded and conveniently presented [Milne et al., 2006]. These semantic resources include freely available ontologies, and online encyclopedias as Wikipedia [Wikipedia, 2004], which could be integrated with generic lexical resources in order to solve the problem of schema matching in specific domains [Suchanek et al., 2008, Mihalcea and Csomai, 2007].

In this thesis, this problem has not been addressed; however, it represents an interesting future research line as described in Chapter 7.

Chapter 3

The MOMIS System

This chapter is focused on the MOMIS[1] system (Mediator EnvirOment for Multiple Information Sources), which represents the application context of the methods proposed in this thesis. MOMIS has been developed by the DBGroup of the University of Modena and Reggio Emilia. It is an Intelligent Data Integration framework designed for the integration of heterogeneous data sources that adopts a GAV approach (see Section 2.1.1). A general description of MOMIS is provided in this chapter. For more details about MOMIS, please refer to [Beneventano et al., 2003a, Bergamaschi et al., 1999]. An open-source version of MOMIS has been released on May 2010 by the DataRiver Spin-Off of the University of Modena and Reggio Emilia[2].

Moreover, in this chapter, we present an early effort to obtain an effective approach for Ontology-Based Data Integration, based on the combination of MOMIS and STASIS (SofTware for Ambient Semantic Interoperable Services) [Abels et al., 2008b]. The STASIS IST project[3] is a research and development project sponsored under the EC 6th Framework programme. In STASIS, a general framework to perform Ontology-Driven Semantic Mapping has been proposed [Beneventano et al., 2008a]. The MOMIS-STASIS approach has been published in [Beneventano et al., 2009b, Beneventano et al., 2009c].

The rest of the chapter is organized as follows: Section 3.1 describes the MOMIS system, its architecture, and its components; in Section 3.2, the MOMIS-STASIS approach and an application example are described.

---

[1] See `http://www.dbgroup.unimore.it` for references about the MOMIS project.

[2] `http://www.datariver.it`

[3] FP6-2005-IST-5-034980, `http://www.stasis-project.net`

3.1 The MOMIS Data Integration System

Definition 13 *(The MOMIS Data Integration System) Given a set N of data sources to be integrated, we can define a MOMIS Data Integration System IS = (GS, N, M) as constituted by:*

- *A Global Schema, which is a schema expressed in the ODL$_{I^3}$ language*
- *A set N of local data sources; each source has a schema also expressed in ODL$_{I^3}$*
- *A set M of GAV mapping assertions between the Global Schema and N, where each assertion associates to an element G in the Global Schema a query q_N over the schemata of the N local sources. More precisely, for each global class $G \in theGlobalSchema$ we define:*
  1. *a (possibly empty) set of classes, denoted by L(G), belonging to the local sources in N*
  2. *a conjunctive query q_G over the L(G) classes*

Intuitively, the Global Schema is the intensional representation of the information provided by the Data Integration System, whereas the mapping assertions specify how such an intensional representation relates to the local sources managed by the Integration System. The semantics of an Integration System is defined in [Cali et al., 2002, Beneventano and Lenzerini, 2005].

MOMIS performs information extraction and integration from both structured and semi-structured data sources. An object-oriented language, with an underlying Description Logic, called ODL$_{I^3}$, described in Section 3.1.1, is introduced as schema language. Information integration is then performed in a semi-automatic way, by exploiting the knowledge in a Common Thesaurus (defined by the framework) and the ODL$_{I^3}$ descriptions of the local source schemata with a combination of clustering techniques and Description Logics. This integration process gives rise to a virtual integrated view (the Global virtual Schema) of the underlying sources for which mapping rules and integrity constraints are specified to handle heterogeneity. Given a set of data sources related to a domain, it is possible to semi-automatically synthesize a Global Schema that conceptualizes a domain and thus might be thought as a basic domain ontology for the integrated sources.

3.1.1 The ODL$_{I^3}$ Language

The ODL$_{I^3}$ language used in the MOMIS system to represent data is an extension of the *Object Definition Language* (ODL), an object-oriented language

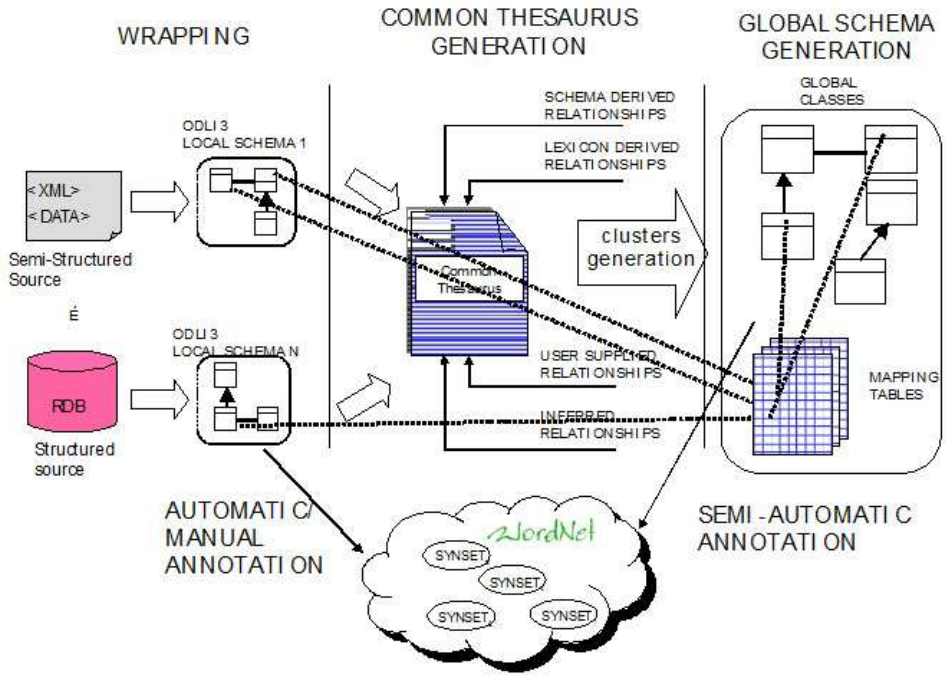

Figure 3.1: The MOMIS Global Schema generation process.

developed by ODMG[45]. ODL$_{I^3}$ is transparently translated into a Description Logic [Beneventano et al., 2003c, Bergamaschi et al., 2001]. ODL$_{I^3}$ allows different kinds of data sources and the view resulting from the integration process to be represented in a common data model. In ODL$_{I^3}$ all the the data sources are represented as a set of classes and properties (for a full description see A). Properties may be composed or simple and in the following we refer to them also as attributes. Some constructors and rules are present in the language to handle the heterogeneity:

Union constructor The union constructor is introduced to express alternative data structures in the definition of an ODL$_{I^3}$ class, thus capturing requirements of semi-structured data;

Optional constructor The optional constructor is introduced for properties to specify that an attribute is optional for a property (i.e., it could be null in the

[4] http://www.odmg.org/
[5] http://www.service-architecture.com/database/articles/odmg_3_0.html

instance);

Integrity constraint rules. This kind of rule is introduced in ODL$_{I^3}$ in order to express, in a declarative way, *if then* integrity constraint rules at both intra- and inter-source level.

Intensional/Lexical relationships. They are *terminological relationships* (also called in the following "lexical relationships" expressing intra- and inter-schema knowledge for the source schemata. Lexical relationships are defined between classes and attributes, and are specified by considering class/attribute labels, also called terms. Formally, a lexical relationship can be defined as:

Definition 14 (Lexical relationship) *Let T_1 and T_2 be two heterogeneous schemata, and the elements $t_i \in T_1$, $t_j \in T_2$. A lexical relationship is defined as the triple $< t_i, t_j, R >$ where R defines the type of the relationship between t_i and t_j. The types of lexical relationship are:*

- *SYN (Synonym-of): defined between two elements whose meanings are synonymous, formally*

  $$t_i \textit{ SYN } t_j \textit{ iff } \exists\ s_{\#w} \textit{ synonym of } s_{\#u}$$

  where $s_{\#w}$ is an annotation assigned to t_i and $s_{\#u}$ is an annotation assigned to t_j;

- *BT (Broader Term): defined between two elements where the meaning of the first is more general than the meaning of the second (the opposite of BT is NT, Narrower Term), formally*

  $$t_i \textit{ BT } t_j \textit{ iff } \exists\ s_{\#w} \textit{ hypernym of } s_{\#u}$$

  where $s_{\#w}$ is an annotation assigned to t_i and $s_{\#u}$ is an annotation assigned to t_j;

- *RT (Related Term): defined between two elements whose meanings are related in a meronymy hierarchy , formally*

  $$t_i \textit{ RT } t_j \textit{ iff } \exists\ s_{\#w} \textit{ is related to } s_{\#u}$$

where $s_{\#w}$ is an annotation assigned to t_i and $s_{\#u}$ is an annotation assigned to t_j.

Lexical relationships may be computed by exploiting the semantics of the schemata and by annotating the schema elements w.r.t. a lexical resources of reference. In MOMIS lexical annotation is performed w.r.t the WordNet lexical database. MOMIS generates lexical relationships by using the following WordNet constructors:

- SYNONYMY (similar relation) corresponds to a SYN relationship;
- HYPONYMY (sub-name relation) corresponds to an NT relationship;
- HYPERNYMY (super-name relation) corresponds to a BT relationship;
- HOLONYMY (whole-name relation) corresponds to an RT relationship;
- MERONYMY (part-name relation) corresponds to an RT relationship.
- CORRELATION (two terms share the same hypernym) corresponds to a RT relationship.

Extensional/Structural relationships. Lexical relationships between two classes C_1 and C_2 of the same schema may be "strengthened" by establishing that they are also *extensional* relationships. Structural relationships are also called intra-schema relationships. Formally, we can define a structural relationship as:

Definition 15 (Extensional Relationship) *Let T be a schema. An extensional relationship is a relationship defined as the triple $< t_i, t_j, R >$ where $t_i, t_j \in T$, and R specifies a type of structural relationship between t_i and t_j. The types of structural relationships are:*

- *SYN_{EXT}: t_i is equivalent to t_j iff $extension(t_i) = extension(t_j)$;*
- *BT_{EXT}: t_i BT_{EXT} t_j (or t_i subsumes t_j) iff $extension(t_i) \supseteq extension(t_j)$ (the opposite of BT_{EXT} is NT_{EXT});*
- *RT_{EXT}: t_i RT_{EXT} t_j (or t_i is related to t_j) iff $extension(t_i)$ is related to $extension(t_j)$ (the opposite of BT_{EXT} is NT_{EXT});*

For example, when analyzing XML data files, MOMIS generates BT_{EXT} and NT_{EXT} relationships from couples IDs and IDREFs (in an XML file an ID is an identifier for an element and a IDREF is a reference to an ID) and RT relationships from nested elements. Further extraction rules can be applied to other data models. For example, we extract intra-schema RT relationships from Foreign Keys in

relational source schemata. In the relational model, a Foreign Key is a set of attributes of a relation used to express a reference from a relation to another. When a Foreign Key is also a Primary Key, in both the original and referenced relation, MOMIS extracts BT_{EXT} and NT_{EXT} relationships, which are derived from inheritance relationships in object-oriented schemata [Beneventano et al., 2003b].

Mapping Rules. This kind of rule is introduced in ODL$_{I^3}$ in order to express relationships holding between the Global Schema and the local schemata.

Using ODL$_{I^3}$ for representing sources and ontologies is not a limitation: the interoperability of the ODL$_{I^3}$ descriptions is guaranteed by a software module able to translate the descriptions into the Web Ontology Language OWL [Orsini, 2004].

3.1.2 Global Schema Generation with MOMIS

The Global Schema generated by the MOMIS system is composed of a set of global classes that represent the information contained in the underlying sources and the mappings that establish the connections among the Global Class attributes and the Local Schema attribute [DBGroup, 2010]. The process of creation of the Global Schema and the mappings, shown in Figure 3.1, can be summarized in the following steps:

Local source schema extraction. The first phase of the integration process is the choice of the data sources and their translation into the ODL$_{I^3}$ format. The translation process is performed by the MOMIS wrappers, which logically converts the source data structure into the ODL$_{I^3}$ model. The wrapper architecture and interfaces are crucial, because wrappers are the focal point for managing the diversity of data sources.

Lexical annotation of Local Sources. This phase represent the core of the MOMIS system and the focus of this thesis. The goal of the annotation phase is to semantically annotate labels denoting schema elements according to a common lexical resource. As described in Section 2.2, the WordNet lexical database is usually referred to as lexical reference [Miller et al., 1990], but other lexical references might be used to cope with specific domains. The annotation step can be performed manually by the integration designer. The manual annotation is composed of two different steps: in the Base Form choice step, the WordNet morphological processor suggests a word form corresponding to the given term; in the Meaning choice step, the designer can choose to map an element to zero, one or

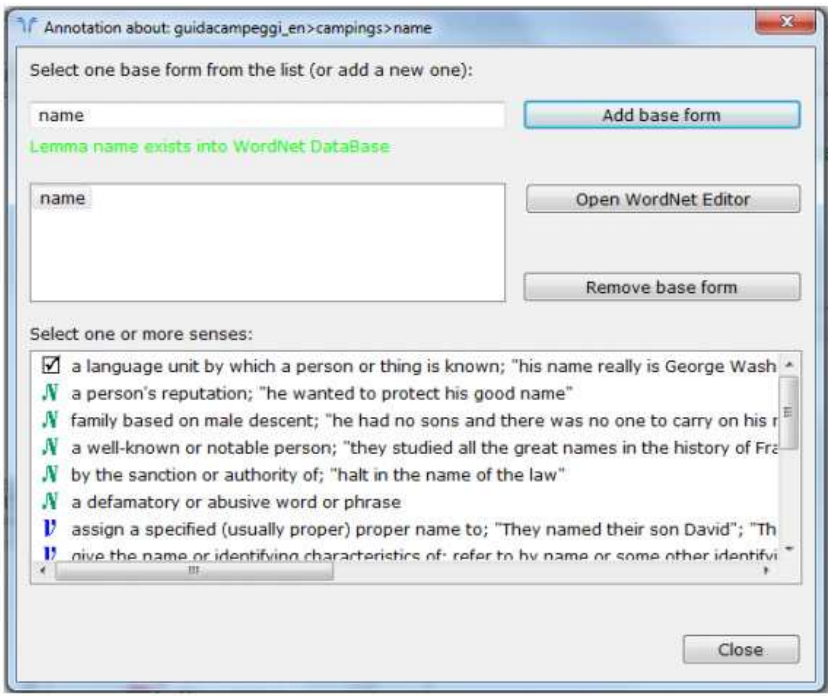

Figure 3.2: The MOMIS manual annotation process.

more senses. Figure 3.2, shows an example of manual annotation: the MOMIS GUI allows the designer to manually annotate the term “Name”.

WNEditor addresses three important issues: (1) inserting new synsets for a term: the WNEditor provides an approximate matching technique that computes syntactic and semantic similarity between the glosses associated to two synsets; (2) inserting new lemmas: an approximate string match algorithm performs the similarity search on the whole synset network; (3) inserting new relationships: WNEditor supports the designer in the definition of new relationships between synsets.

As manual annotation is not feasible when the number of schema to be integrated is large, we studied a method to perform semi-automatic lexical annotation, which is described in details in Chapter 4. Moreover, during this phase, the designer can insert new terms in WordNet, by using the *WNEditor* MOMIS component.

Common Thesaurus generation. Starting from the annotated local schemata, MOMIS constructs a Common Thesaurus describing intra and inter-schema ODL_{I^3} relationships in the form of SYN (synonyms), BT/NT (broader terms/narrower terms), and RT (meronymy/holonymy) relationships. The Com-

mon Thesaurus is structured very much like an *Associative Network*, where nodes (class or attribute names) are connected through bidirectional relationships. It is constructed through an incremental process in which the following relationships are added:

- schema-derived relationships: these are structural ODL$_{I^3}$ relationships which are holding at intra-schema level and they are automatically extracted by analyzing each schema separately. Some heuristic can be defined for specific kind of sources. For example, MOMIS extracts intra-schema RT relationships from foreign keys in relational source schemata. When a foreign key is also a primary key, in both the original and referenced relation, MOMIS extracts BT and NT relationships, which are derived from inheritance relationships in object-oriented schemata.
- lexicon-derived relationships: these are the lexical ODL$_{I^3}$relationships which are automatically extracted starting from the annotation phase is exploited to translate relationships holding at the lexical level into relationships to be added to the Common Thesaurus. These relationships are inferred from lexical knowledge (e.g. by querying WordNet for relationships between senses).
- designer-supplied relationships: new relationships can be supplied directly by the designer, to capture specific domain knowledge.
- inferred relationships: Description Logics (DL) techniques of ODB-Tools [Bergamaschi et al., 1997], are exploited to infer new relationships.

Global Schema Generation. The Global virtual Schema consists of a set of classes (called Global Classes), plus mappings to connect the global attributes of each Global Class and the local source attributes. The MOMIS methodology allows identifying similar ODL$_{I^3}$ classes (i.e. classes that describe the same or semantically related concept in different sources) and mappings to connect the global attributes of each global class to the local source attributes. To this end, affinity coefficients are evaluated for all possible pairs of ODL$_{I^3}$ classes, based on the relationships in the Common Thesaurus properly strengthened. Affinity coefficients determine the degree of matching of two classes based on their names (*Name Affinity coefficient*) and their attributes (*Structural Affinity coefficient*) and are fused into the *Global Affinity coefficient*, calculated by means of the linear combination of the two coefficients [Castano et al., 2001a, Castano et al., 2001b].

Definition 16 (Name Affinity coefficient). *Given two schema elements e_i and e_j, their Name Affinity coefficient, denoted $NA(e_i, e_j)$ is:*

$$NA(e_i, e_j) = \begin{cases} A(n(e_i), n(e_j)), & \textit{if } A(n(e_i), n(e_j)) \geq \alpha \\ 0, & \textit{otherwise} \end{cases}$$

where α is an affinity threshold determined by the designer.

The evaluation of the Structural Affinity coefficient is based on the definition of affinity classes for element properties. Let $P_{ij} = P(e_i) \cup P(e_j)$ be the set of properties of both e_i and e_j. Based on semantic correspondences, properties of P_{ij} are partitioned into affinity classes. An affinity class contains all the properties that have a weak or strong correspondence with a given property p_l, called *representative* of the class. Formally, denoting an affinity class by $[p_l]$ we have,

$$[p_l] = \{p_k \in P_{ij} | pk \longleftrightarrow *p_l \vee pk \longleftrightarrow p_l\}$$

Let ${P^{AC}}_{ij} = \{[p_l] | p_l \in P_{ij}\}$ be the set of affinity classes resulting from P_{ij}. For the evaluation of Structural Affinity, only the so-called well-formed affinity classes are of interest among those in ${P^{AC}}_{ij}$. An affinity class $[p_l]$ is said to be well-formed, denoted $[p_l]^{wf}$, if it contains at least one property of e_i and of e_j, respectively. A well-formed affinity class is said to be minimal if it contains exactly one property of e_i and one property of e_j. We are now ready to define the Structural Affinity coefficient as follows:

Definition 17 (Structural Affinity coefficient). *Given two schema elements e_i and e_j, their Structural Affinity coefficient, denoted $SA(e_i, e_j)$ is:*

$$SA(e_i, e_j) = \frac{2|\{[p_l] \in {P^{AC}}_{ij} | [p_l]^{wf}\}|}{|P_{ij}|}$$

where notation $|A|$ denotes the cardinality of the set A. The Structural Affinity coefficient measures the level of overlapping between the structure of the two schema elements based on well-formed affinity classes of their properties and returns a value in the range $[0; 1]$. The value 0 indicates that no well-formed affinity classes are defined for properties of e_i and e_j. The value 1 indicates that all affinity classes are well-formed and minimal (i.e., each property of e_i has a semantic correspondence with only one property of e_j and vice versa).

To assess the level of affinity of two schema elements in a comprehensive way, the Global Affinity coefficient is defined.

Definition 18 (Global Affinity coefficient). *Given two schema elements e_i and e_j, their Global Affinity coefficient, denoted $GA(e_i, e_j)$ is:*

$$GA(e_i, e_j) = \begin{cases} wn_A \cdot NA(e_i, e_j) + w_{SA} \cdot SA(e_i, e_j), & \textit{if } NA(e_i, e_j) \neq 0 \\ 0, & \textit{otherwise} \end{cases}$$

where weights w_{NA} and w_{SA}, with $w_{NA}, w_{SA} \in [0,1]$ and $w_{NA} + w_{SA} = 1$, are introduced to assess the relevance of each kind of affinity in the computation of $GA(e_i, e_j)$.

The Global Affinity coefficient of e_i and e_j is the linear combination of their Name and Structural Affinity coefficients.

Global affinity coefficients are then used by a hierarchical clustering algorithm, to cluster ODL_{I^3} classes according to their degree of affinity. For each cluster C, a Global Class G, with a set of Global Attributes $GA_1, \ldots GA_N$, and a Mapping Table MT, expressing mappings between local and global attributes, are defined. The Mapping Table is a table whose columns represent the local classes which belong to the Global Class and whose rows represent the global attributes. An element $MT[GA][LC]$ is a function which represents how local attributes of the Local Class LC are mapped into the global attribute GA:

$$MT[GA][LC] = f(LAS)$$

where LAS is a subset of the local attributes of LC.

Global Schema Lexical Annotation. To annotate a Global Schema means to assign a name and a set (eventually empty) of meanings to each global element (class or attribute). MOMIS automatically annotates each global element proposing the broadest meaning extracted from the annotations of the local sources, based on the relationships included in the Common Thesaurus. Names and meanings have then to be confirmed by the ontology designer. This annotation step is a significant result, since these metadata may be exploited by external users and applications, for example by exporting the annotated Global Schema as an OWL ontology.

After the Global Schema generation process, each Global Class (GC) is associated to a Mapping Table (MT). Starting from the Mapping Table of GC, the integration designer can implicitly define the mapping query q_G associated to the GC G by:

1. extending the MT with
   - Data Conversion Functions from local to global attributes
   - Join Conditions among pairs of local classes belonging to G
   - Resolution Functions for global attributes to solve data conflicts of local attribute values.
2. using and extending the *full outerjoin-merge* operator, proposed in [Naumann et al., 2004], to solve data conflicts of common local attribute values and merge common attributes into one [DBGroup, 2010].

3.1.3 Query Execution

The MOMIS Query Manager allows the user to pose a query expressed in OQL [Beneventano et al., 2003c, Orsini, 2009] over the ontology and to obtain a unified answer from all the data sources integrated in the Global Schema. When the MOMIS Query Manager receives a query, it rewrites the global query as an equivalent set of queries expressed on the local schemata (local queries); this query translation is carried out by considering the mapping between the Global Schema and the local schemata. The query translation is thus performed by means of query unfolding, i.e. by expanding a global query on a global class G of the Global Schema according to the definition of the mapping query q_G. The local queries are then executed on the sources, and MOMIS performs the fusion of the local answers into a consistent and concise unified answer, and present the global answer to the user. In order to assure full usability of the system even to users with low information technology skills, that are often the target of many information integration application, a graphical user interface to compose queries over the MOMIS Global Schema was developed in [Sala, 2010].

3.1.4 MOMIS Architecture

In this section, the architecture of the MOMIS system, shown in Figure 3.3 is briefly described. Further details can be found in [Orsini, 2009].

The main components of MOMIS are:

- **Wrappers:** Wrappers are software modules with the role of managing the interactions with the data sources. The MOMIS wrappers translate the source data structures into ODL_{I^3}. Their role is to deal with the diversity of the data sources thus allowing MOMIS to pay no attention to the language details of the different data sources. Wrappers are available for different kind of data sources, ranging from different database management systems to semi-structured data like XML, RDF, OWL formats. Wrappers logically guarantee two main operations:
  - *getschema()* translates the schema from the original format into ODL_{I^3}, dealing with the necessary data type conversions
  - *runquery()* executes a query on the local source. The MOMIS Query Manager translates a query on the Global Schema (a global query) into a set of local queries to be locally executed by means of wrappers.
- **Global Schema Builder:** The Global Schema Builder is the main module that interacts with the wrappers and the Service tools to generate the Global

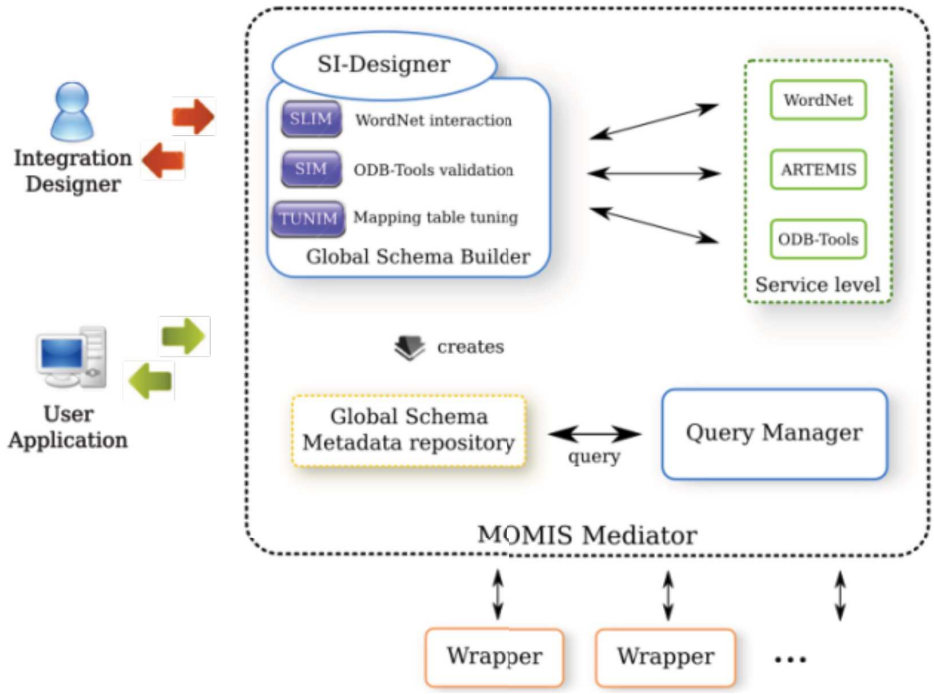

Figure 3.3: The MOMIS architecture

Schema. It is composed of different components that implements the different steps of the integration process described in Section 3.1.2.

- **Service tools:** the Service tools are important modules exploited by the Global Schema Builder and the Query Manager. These include the software module in charge of managing the WordNet lexical database used for the annotation phase, the Artemis tool implementing the clustering algorithm, and the ODB-Tools reasoner used to calculated inferred relationships.

- **Query manager:** the Query Manager is the component in charge of solving a user query: it generates the local queries queries for wrappers, starting from a global query formulated by the user on the global schema. The Query Manager provides a graphical Query Composer to graphically formulate queries on a Global Schema described in details in [Sala, 2010].

3.1.5 MOMIS and Web Services

In the NeP4B project[6] [Beneventano et al., 2008b] the traditional MOMIS architecture has been extended and coupled with XIRE (eXtended Infor-

[6] http://www.dbgroup.unimo.it/nep4b

mation Retrieval Engine), a semantic Web service retriever developed by the "DISCO" of the University of Milano Bicocca [Domenico et al., 2009, Bergamaschi and Maurino, 2009]. In the NeP4B approach, data sources and services are grouped into semantic peers. Each semantic peer generates a Peer Virtual View (PVV), i.e. a unified representation of the data and the services held by the sources belonging to the peer. A PVV is made up of the following components:

- a *Semantic Peer Data Ontology (SPDO)* of the data, i.e. a common representation of all the data sources belonging to the peer; the SPDO is built by means of MOMIS as described in [Sala, 2010];
- a *Global Light Service Ontology (GLSO)* that provides, by means of a set of concepts and attributes, a global view of all the concepts and attributes used for the descriptions of the Web services available in the peer;
- a set of *mappings* which connect GLSO elements to SPDO elements.

In particular, shown in Figure 3.4 MOMIS has been extended with new modules to create and query a PVV. These new components are:

- *PVV builder* which is in charge of generating the PVV that represents both the data sources and the services available in a peer. The PVV builder is a component of the MOMIS Schema Builder that provides the mappings between the SPDO and the GLSO (i.e. the ontology representing the web services in a peer) elements.
- *SPDO-GLSO Mappera*, which extract the relevant keywords from a query on the SPDO. The relevant keywords are those identifying schema elements and searched values. This component then evaluates the mappings computed by the PVV builder, to extract, for each keyword, the correspondent terms in the GLSO, if they exist.
- *Xire Connector* which is the component in charge of managing the interactions between MOMIS and the XIRE system.

For further details about Data and Services integration please refer to [Sala, 2010].

3.2 The MOMIS-STASIS Approach

STASIS aims to enable SMEs (Small and Medium Enterprises) to fully participate in the Economy, by offering semantic services and applications based on open registries and repository networks. The goal of the STASIS project is to create a

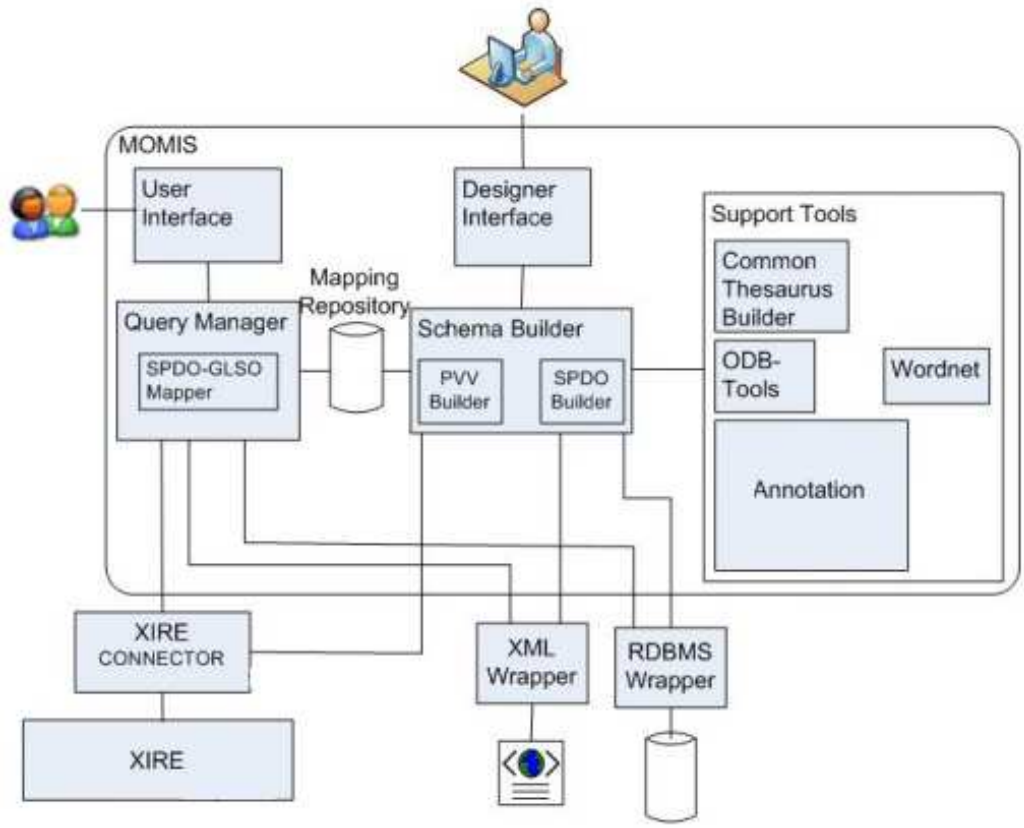

Figure 3.4: The data and services aggregated search prototype.

comprehensive application suite which allows enterprises to simplify the mapping process between data schemata, by providing a GUI, allowing users to identify semantic elements in an easy way [Abels et al., 2008a, Abels et al., 2008b]. In the STASIS project, OWL is used as language to include in the framework generic external ontologies.

In this section, we describe an approach to combine the MOMIS and STASIS frameworks in order to obtain an effective approach for Ontology-Based Data Integration. An ontology-based approach to data integration relies on the alignment of the concepts of a global ontology that describe the domain, with the concepts of the ontologies that describe the data in the local databases. Once the alignment between the global ontology and each of the local ontologies is established, users can potentially query hundreds of databases using a single query that hides the underlying heterogeneities. The proposed approach addresses the following points:

1. enabling the MOMIS system to perform annotation with respect to a *generic* OWL ontology, not only using only the WordNet lexical database;

2. enabling the MOMIS system to exploit a *multiple ontology* approach with respect to the actual *single ontology* approach;

3. developing a new method to compute semantic mappings among source schemata in the MOMIS system.

3.2.1 Ontology-Based Data Integration

This section describes an approach to use the Ontology-Driven Semantic Mapping Framework provided by STASIS, during the MOMIS Global Schema generation process. In the following, we will refer to this new approach as the MOMIS-STASIS approach.

The MOMIS-STASIS approach is shown in Figure 3.5. It can be divided into two macro-steps: (1) STASIS: Semantic Link Generation (shown in Figure 3.5-a) and (2) MOMIS: Global Schema Generation (shown in Figure 3.5-b).

STASIS: Semantic Link Generation

As stated in [Abels et al., 2008a, Abels et al., 2008b] the key aspect of the STASIS framework, which distinguishes it from most existing semantic mapping approaches, is to provide an easy to use GUI, allowing users to identify *semantic entities* in an easy way, where with semantic entities we mean the set of classes and properties of the a data sources. Once this identification has been performed STASIS lets users to map their semantic entities to the entities of their business partners where possible assisted by STASIS. This allows users to create mappings in a more natural way by considering the meaning of elements rather than their syntactical structure. Moreover, all mappings that have been created with STASIS, as well as all semantic entities, are managed in a distributed registry and repository network. This gives STASIS another significant advantage over traditional mapping creation tools as STASIS may reuse all mappings. In this way, STASIS may make some intelligent mapping suggestions by reusing mapping information from earlier semantic links.

Besides the semantic links explicitly provided by the user, an Ontology-Driven Semantic Mapping approach, for the STASIS framework, has been proposed [Beneventano et al., 2008a]. The mappings between semantic entities used in different schemata can be achieved based on annotations linking the semantic entities with some concepts belonging to a part of an ontology. In [Beneventano and Montanari, 2008], this framework has been further elaborated and it has been applied to the context of products and service catalogues.

An overview of the process for Ontology-Driven Semantic Mapping Discovery is given in Figure 3.5-a. It can be summed up into 3 steps (each step number is

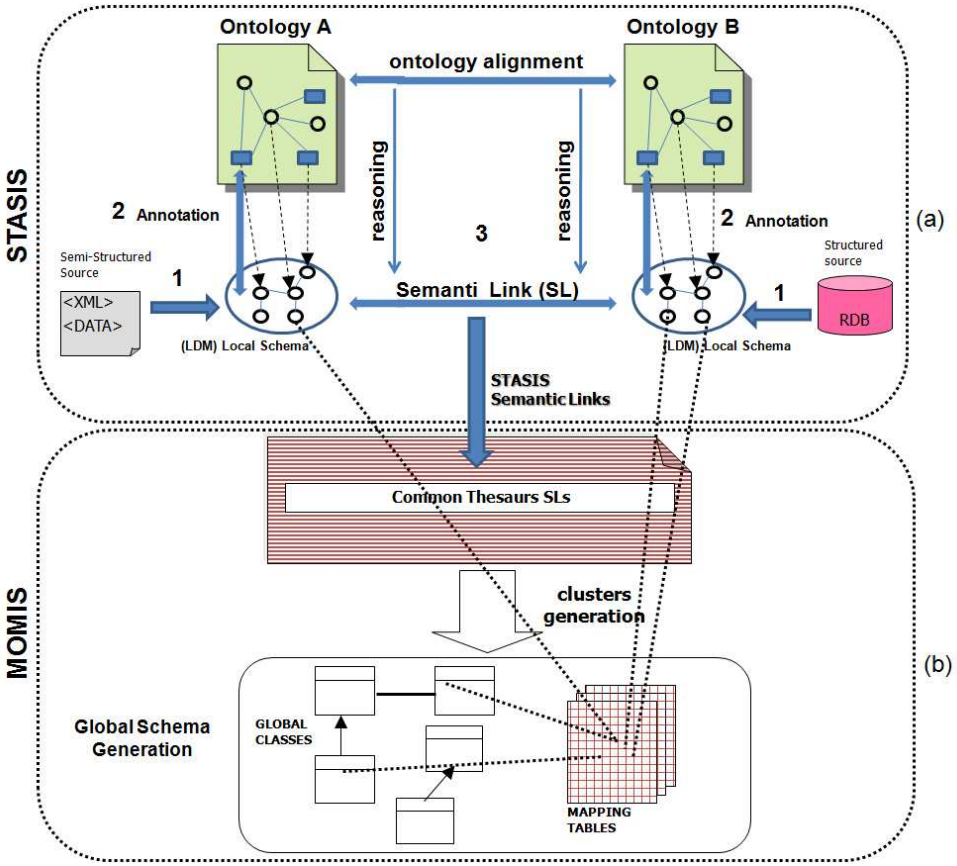

Figure 3.5: The MOMIS-STASIS approach.

correspondingly represented in figure): (1) obtaining a neutral schema representation, (2) annotating local sources, and (3) discovering semantic mappings.

Step 1. Obtaining a neutral schema representation. As sketched in Figure 3.5-a, the STASIS framework works on a neutral representation, which is abstracted from the specific syntax and the data model of a particular schema definition by using wrappers; therefore, all the structured and semi-structured local sources first need to be expressed in a neutral format. The neutral representation is obtained by describing the local schemata through a unified data model called *Logical Data Model* (LDM). This model contains common aspects of most semantic data models: it allows the representation of *classes* (or concepts), i.e. unary predicates over individuals, *relationships* (or object properties), i.e. binary predicates relating individuals, and *attributes* (or data-type properties) i.e. binary predicates relating individuals with values such as integers and strings; classes are organized in the familiar *is-a* hierarchy. *Classes*, *relationships* and *attributes* are

called *semantic entities*.

Step 2. Annotating local sources. The proposed mapping process identifies mappings between semantic entities through a "reasoning" with respect to aligned ontologies. For this purpose the semantic entities need to be annotated with respect to one or more ontologies. More formally, an *annotation element* is a 4-tuple $< ID, SE, R, concept >$ where ID is a unique identifier of the given annotation element; SE is a semantic entity of the schema; $concept$ is a concept of the ontology; R specifies the semantic relationship which may hold between SE and $concept$. The following semantic relationships between semantic entities and the concepts of the ontology are used: equivalence (AR_EQUIV); more general (AR_SUP); less general (AR_SUB); disjointness (AR_DISJ).

Within the STASIS framework only simple automatic annotation techniques, e.g. the "name-based technique" are implemented, where the annotation between a semantic entity and a ontology concept is discovered by comparing only the strings of their names. The main drawback of this automatic technique is due to the existence of *synonyms* and *homonyms*. For these reason the designer has to manually refine the annotations in order to capture the semantics associated to each entities. In Chapter 4 a semi-automatic lexical annotation method to overcome this limitation is described.

Step 3. Discovering semantic mappings. Based on the annotations made with respect to the ontologies and on the logic relationships between these aligned ontologies, reasoning can identify correspondences among the semantic entities and support the mapping process. Given two schemata S1 and S2, and assuming that OntologyA and OntologyB are the reference ontologies which have been used to annotate the content of S1 and S2 respectively, given a mapping between OntologyA and OntologyB which provides a correspondence between concepts and relationships in the two ontologies, a semantic mapping between the annotated schemata S1 and S2 is derived. The following semantic mappings between entities of two source schemata (called *semantic link*- SL) can be discovered: equivalence (EQUIV); more general (SUP); less general (SUB); disjointness (DISJ); this definition is based on the general framework proposed in [Giunchiglia et al., 2007].

Formally, a SL is a 4-tuple $< ID, semantic\_entity1, R, semantic\_entity2 >$, where ID is the unique identifier of a given mapping element; $semantic\_entity1$ is the entity of the first local schema; R specifies the semantic relationship which may hold between $semantic\_entity1$ and $semantic\_entity2$; $semantic\_entity2$ is an entity of the second local schema.

An application example of the Ontology Driven Semantic Mapping approach is described in Section 3.2.2; other examples can be found

in [Beneventano and Montanari, 2008].

MOMIS: Global Schema Generation

As described before (see Section 3.1.2) MOMIS performs information extraction and integration from both structured and semi-structured data sources.

In the MOMIS-STASIS approach, the semantic links among source schemata are the semantic links defined with the STASIS framework; in other words, we consider as input of the Global Schema generation process the *Common Thesaurus SLs* generated by the STASIS framework. An overview of this Global Schema generation process is given in Figure 3.5-b.

The rest of the integration process is not modified: exploiting the Common Thesaurus SLs and the local sources schemata, the approach generates a Global Schema consisting of a set of global classes, plus a MT for each global class, which contains the mappings to connect the global attributes of each global class with the local sources' attributes.

3.2.2 Example

As a simple example let us consider two relational local sources $L1$ and $L2$, where each schema contains a relation describing purchase orders:

```
L1: PURCHASE_ORDER(ORDERID, BILLING_ADDRESS, DELIVERY_ADDRESS,
 DATE)
L2: ORDER(NUMBER, CUSTOMER_LOCATION, YEAR, MONTH, DAY)
```

STASIS: Semantic Link Generation

Step 1. Obtaining a neutral schema representation
During this step the local sources L1 and L2 are translated in a neutral representation and are represented in the LDM data model; for a complete and formal description of such representation see [Beneventano et al., 2008a], where a similar example is discussed. As said before, for the purpose of this paper, we consider that the local schema L1 contains a class PURCHASE_ORDER with the attributes ORDERID, BILLING_ADDRESS, DELIVERY_ADDRESS, DATE. In this way L1.PURCHASE_ORDER, L1.PURCHASE_ORDER.BILLING_ADDRESS, L1.PURCHASE_ORDER.DELIVERY_ADDRESS etc. are semantic entities. In the same way, the local schema L2 contains a class ORDER with attributes NUMBER, CUSTOMER_LOCATION, YEAR, MONTH, DAY.

Step 2. Local Source Annotation
For the sake of simplicity, we consider the annotation of schemata and the

derivation of mappings with respect to a single common ontology ("Ontology-based schema mapping with a single common ontology" scenario considered in [Beneventano et al., 2008a]). Let us give some examples of annotations of the

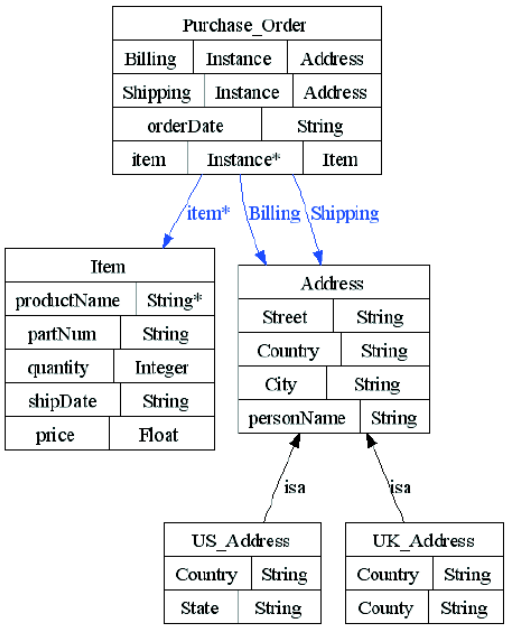

Figure 3.6: The Purchase Order ontology.

above schemata with respect to the Purchase Order Ontology shown in Figure 3.6. In the examples the identifier ID is omitted and a concept C of the ontology is denoted by O:C. In a *simple annotation* the concept O:C is a primitive concept or a primitive role of the ontology (e.g. the class O:ADDRESS or the property O:BILLING). In a *complex annotation* the concept O:C is obtained by using the OWL language constructs (e.g. O:ADDRESS and BILLING-1.Purchase_Orderwhere BILLING-1denotes the inverse of the property O:BILLING). The following annotations are examples of simple annotations:

```
(L1.PURCHASE_ORDER.BILLING_ADDRESS, AR_EQUIV, O:ADDRESS)
```

and

```
(L1.PURCHASE_ORDER.BILLING_ADDRESS, AR_EQUIV, O:BILLING).
```

These annotations are automatically discovered by using string-comparison techniques. However, as this technique does not consider the semantics associated to each entities, the following annotation

```
(L2.ORDER.CUSTOMER_LOCATION, AR_EQUIV, O:ADDRESS)
```

is not discovered: the entities CUSTOMER_LOCATION and the concept ADDRESS have complete different names but, in this context, they have the same meaning. To overcome this problem, the annotations may be obtained by performing manual or automatic WSD (see Section 3.1.2).

An example of complex annotation is

```
(L1.PURCHASE_ORDER.DELIVERY_ADDRESS, AR_EQUIV,
 O:Address and Shipping-1.Purchase_Order)
```

which can be considered as a refinement by the designer of the above simple annotations to state that the address in the PURCHASE_ORDER table is the "address of the Shipping in a Purchase Order".

Other examples of complex annotations are:

```
(L1.PURCHASE_ORDER.BILLING_ADDRESS, AR_EQUIV,
 O:Address and Billing-1.Purchase_Order)
```

where is explicitly declared by the designer to state that the address in the PURCHASE_ORDER table is the "address of the Billing in a Purchase_Order".

```
(L2.ORDER.CUSTOMER_LOCATION, AR_EQUIV,
 O:Address and Shipping-1.Purchase_Order)
```

where is explicitly declared by the designer to state that the address in the ORDER table is the "address of the Shipping in a Purchase_Order".

Moreover, the designer supplies also the annotations with respect to the ontology for the semantic entities L1.PURCHASE_ORDER.ORDERID, L1.PURCHASE_ORDER.DATE and L2.ORDER.NUMBER, L2.ORDER.YEAR, L2.ORDER.MONTH, L2.ORDER.DAY.

Step 3. Semantic mapping discovery
Starting from the previous annotations, for example, the following semantic link is derived:

```
(L2.ORDER.CUSTOMER_LOCATION, EQUIV,
 L1.PURCHASE_ORDER.DELIVERY_ADDRESS)
```

while no semantic link among CUSTOMER_LOCATION and BILLING_ADDRESS is generated.

MOMIS: Global Schema Generation
Given the set of semantic links described above and collected in the Common Thesaurus, the Global Schema is automatically generated and the classes, describing the same or semantically related concepts in different sources, are

Global attributes *ORDER*	Local attributes *ORDER*	Local attributes *PURCHASE_ORDER*
NUMBER	NUMBER	ORDER_ID
DATE	YEAR,MONTH,DAY	DATE
CUSTOMER_LOCATION	CUSTOMER_LOCATION	DELIVERY_ADDRESS
BILLING_ADDRESS	*NULL*	BILLING_ADDRESS

Table 3.1: Mapping Table example

identified and clusterized in the same global class. Moreover, the MT shown in Table 3.1 is automatically created by the MOMIS-STASIS approach.

The global class ORDER is mapped into the local class ORDER of the L1 source and to the local class PURCHASE_ORDER of the L2 source. The NUMBER, DATE and CUSTOMER_ADDRESS global attributes are mapped to both the sources, the BILLING_ADDRESS global attribute is mapped only to the L2 source.

One of the main advantage of the proposed approach is an accurate annotation of the schemata that produces more reliable relationships among semantic entities. The relationships among semantic entities are then exploited in order to obtain a more effective integration process. On the other hand, this more accurate annotation has the disadvantage that is currently performed manually by the integration designer. This drawback will be addressed in the following chapters.

Chapter 4

Word Sense Disambiguation for Semi-Automatic Lexical Annotation

This chapter is focused on the study and development of a semi-automatic method for performing lexical annotation of structured and semi-structured data sources. The proposed method is based on an evolution of some WSD algorithms proposed in the NLP research area to disambiguate texts, adapted to the case of structured and semi-structured data sources. Lexical annotation is performed in the MOMIS system, but the method may be applied in general in the context of schema and ontology matching.

This work was conducted in the context of the MUR FIRB Network Peer for Business project[1], and it has been carried on in collaboration with the Ph.D. and fellow researcher Laura Po. The work described in this chapter has been published in [Bergamaschi et al., 2007c, Bergamaschi et al., 2007d, Bergamaschi et al., 2008].

The rest of the chapter is organized as follows: in Section 4.1, we propose the CWSD (Combined Word Sense Disambiguation) method and its components; Section 4.3 is devoted to the CWSD evaluation ; finally, a comparison of the proposed method with related work is presented in Section 4.4.

4.1 Problem Definition

WSD is the task to computationally determine which sense of a word is activated by its use in a particular context [Navigli, 2009]. WSD can be viewed as a classification task: word senses are the classes, and an automatic classification method is used to assign each occurrence of a word to one or more classes based on the evidence from the context and from external knowledge sources.

[1] http://www.dbgroup.unimo.it/nep4b/it/index.htm

WSD in schema matching has to deal with several problems. First of all, WSD algorithms are traditionally applied on plain text, and as a consequence, often they cannot be directly applied to structured and semi-structured data sources. For example, WSD of schema labels cannot take advantage from the analysis of the syntactic structure of sentences in texts. To the other end, in our context, we can exploit the semantics deriving from the structure of a schema. Moreover, while texts contain words belonging to different parts of speech (i.e. nouns, adjectives, verbs, and adverbs), the great majority of words in schemata belong to the noun syntactic category. This represent an advantage, with respect to the textual context, since it has been demonstrated during one of the most important WSD competition, SemEval[2], that WSD algorithms, on average, obtain better performances in disambiguating nouns.

Moreover, most of the WSD algorithms proposed in the literature assign to a term only one meaning. This approach is simple but suffers from a main limitation: there are cases where more than one WordNet meaning may be associated to a term. The limits become even more obvious when we deal with structured or semi-structured data sources. On these sources, there is not a wide context as in a textual source; in addition, there are less terms that concur to the definition of a concept (i.e. only classes, attributes and relationships). Therefore, it is difficult also for a domain expert designer to select only one meaning for a term as correct for each element.

This observation is not new in the literature: Resnik and Yarowsky ratify that there are common cases where several fine-grained meanings may be correct [Resnik and Yarowsky, 2000]. Let us consider the example shown in Figure 4.2: the noun "bank" in WordNet has ten possible meanings (i.e. synsets), but these meanings are not all mutually exclusive, in fact they may be grouped in different categories and sub-categories. In this case, to disambiguate the noun "bank" with the meaning of "repository" it may be correct to choose one of the first four synsets. In the following, we present the CWSD method which address all the problems previously identified.

4.2 The CWSD method

In this section, we present CWSD, a method for the automatic annotation of structured and semi-structured data sources.

CWSD has been integrated in the MOMIS system, to overcome the heavy user involvement in manual lexical annotation of data source terms (see Section 3.1.2).

CWSD exploits as external lexical resource, the well known lexical database

[2] http://semeval2.fbk.eu/semeval2.php

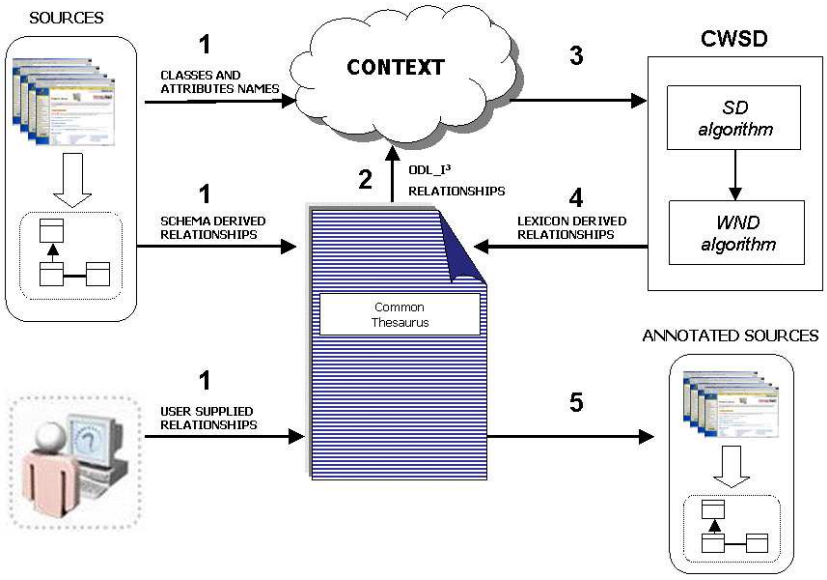

Figure 4.1: Automatic annotation of local data sources with CWSD

WordNet (see Section 2.2). CWSD is based on a combination of two different WSD algorithms. Combined methods are an effective way for improving WSD performance. The idea of combining the results of different WSD methods is not new in the literature [Rigau et al., 1997, Mihalcea and Moldovan, 2000].

In particular, CWSD combines a structural disambiguation algorithm, that exploits the structural relationships extracted from the data source schemata, with a domain algorithm based on the WordNet Domains resource[3]. WordNet Domains has been proven a useful resource for WSD: it has been used in different WSD combined algorithms [Gliozzo et al., 2005, Novischi, 2004].

By using these two algorithms, CWSD tries to couple the traditional WSD approaches from the NLP area with the structural knowledge provided by the database and the semantic web communities [Rahm and Bernstein, 2001, Noy, 2004].

CWSD represents a new approach that, differently from the traditional WSD methods, allow to associate more than one meaning to a term. When a term has several possible meanings, CWSD overcomes the usual disambiguation approaches and associates to the term the union of the meanings related to it.

CWSD is composed of two algorithms: SD (Structural Disambiguation) and WND (WordNet domain Disambiguation).

SD tries to disambiguate schema labels by using semantic relationships in-

[3] http://wndomains.fbk.eu/

I	*Bank*	- REPOSITORY
	I.1	Financial Bank
		I.1a - the institution
		I.1b - the building
	I.2	General Supply/Reserve
II	*Bank*	- GEOGRAPHICAL
	II.1	Shoreline
	II.2	Ridge/Embankment
III	*Bank*	- ARRAY/GROUP/ROW

Figure 4.2: The grouping of the WordNet synsets of "Bank".

ferred from the structure of the data sources; WND tries to disambiguate schema labels by using the domain information supplied by the resource WordNet Domains.

In order to disambiguate the meaning of an ambiguous word, each WSD algorithm receives as input the *context* of the word. Many WSD algorithms in NLP represent the context as a *bag-of-word*, i.e. a set of words that have to be disambiguated [Pahikkala et al., 2005], and sometime they insert in the context the information of the word positions in the text. Other approaches [Banerjee and Pedersen, 2002] consider a *window-of-context* around each target word, and submit all the words in this window as input to the disambiguation algorithm.

In CWSD, the context is composed by: the set of labels (classes and attributes names) to be disambiguated, and the set of structural relationships among these labels included in the Common Thesaurus of MOMIS as shown in Figure 4.1. As described in Section 3.1.2, the Common Thesaurus is a set of ODL_{I^3} relationships describing inter- and intra-schema knowledge among a set of data source schemata in term of SYN (Synonym-of), BT (Broader Term), NT (Narrower Term), and RT (Related Term).

The choice of the context represents a critical issue in WSD and we need to take into account two main factors: (1) the higher is the number of the considered labels, the greater is the probability to introduce noise in the process; (2) the lower is the number of considered labels, the smaller is the probability to find relationships among the considered labels. The default context for a data integration system is given by the data sources to be integrated and the structural ODL_{I^3} relationships. In the following subsections, we describe in details the two algorithms.

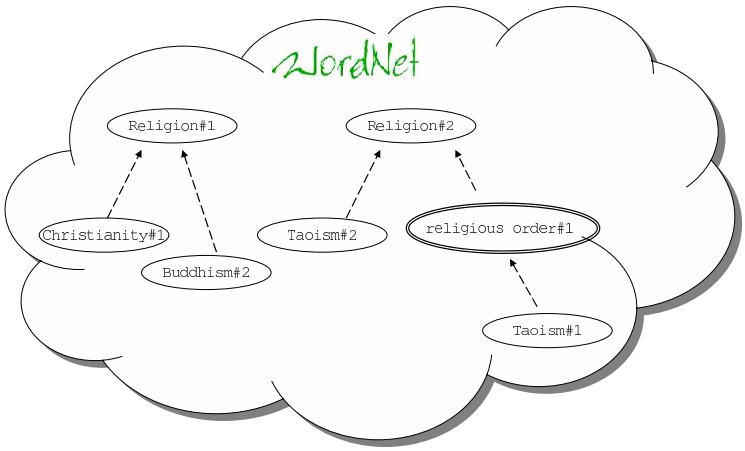

Figure 4.3: Hyponym relationships extracted by SD.

4.2.1 The Structural Disambiguation Algorithm

The SD algorithm exploits the structural ODL_{I^3} relationships (i.e., extensional SYN, BT/NT, and RT, see Section 3.1.2) of a data source to infer the meanings of schema labels.

SD tries to find a corresponding lexical relationship when an extensional relationship hold between two terms. In practice, if we have a direct/chain of relationships between two terms, we try to find the semantically related meanings in WordNet and to annotate the terms with these meanings. A chain of relationships is obtained by navigating through the WordNet lexical database relationships.

Figure 4.4 shows an example of the application of the SD algorithm to a hierarchical data source, i.e. a portion of the first three level of the "society" subtree in the Google directory. First of all, all the ISA relationships in the schemata are extracted from the sources and inserted in the Common Thesaurus as NT relationships, then, SD finds the corresponding hyponym/hypernym relationships in WordNet. The annotations generated by using SD enrich the Common Thesaurus of new ODL_{I^3} relationships (all the lexicon-derived relationships shown in figure). Using the ODB-Tools MOMIS component [Bergamaschi et al., 1997] the Common Thesaurus infers new relationships.

Figure 4.4 shows some hyponym relationships found in WordNet, and the correspondent chosen synsets. In particular, for the labels "Religion" and "Taoism", SD chooses two synsets, because two different hyponym relationships exist between these labels.

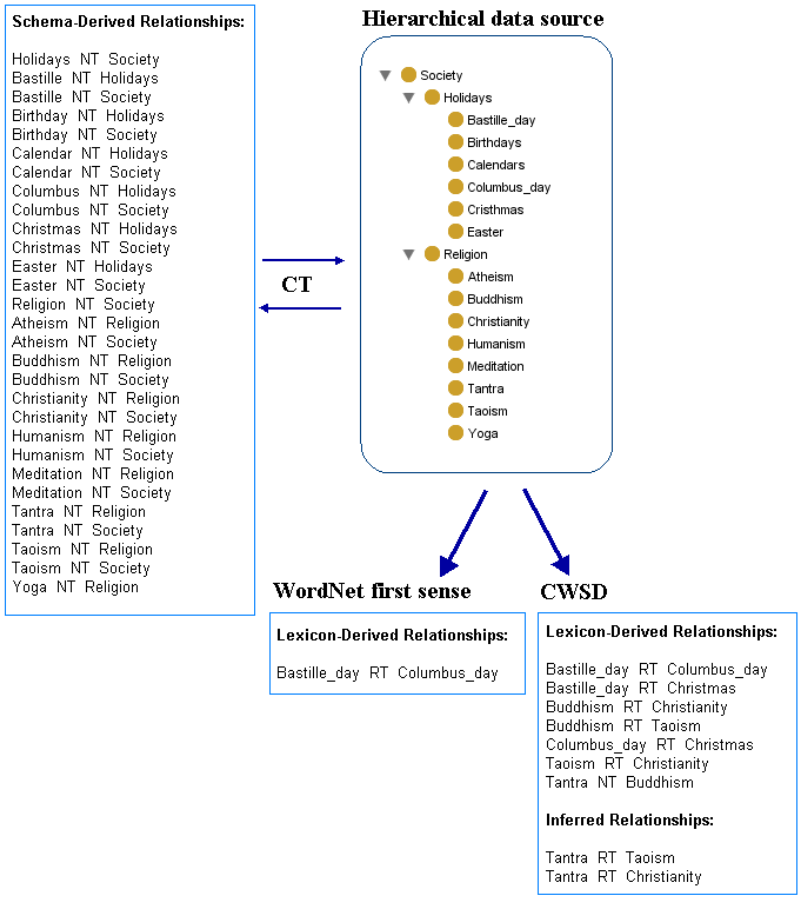

Figure 4.4: Enrichment of the CT with relationships extracted by CWSD.

4.2.2 The WordNet Domains Algorithm

WordNet Domains [Gliozzo et al., 2004, Magnini et al., 2002] can be considered an extended version of WordNet in which each synset has been associated with one or more domain labels. For example, the term "crane" has synsets in the domains of *Zoology* and *Construction*. The information brought by domains is complementary with the one already present in WordNet. A domain may include synsets from different syntactic categories and meanings from different WordNet sub-hierarchies. Besides, domains may group meanings of the same terms, into a thematic cluster, which has the important side effect of reducing the level of ambiguity of polysemic words. WordNet Domains organizes about two hundred domains in a hierarchy, where each level is made up of domains with the same degree of specificity, as described in [Magnini et al., 2002]. In WordNet, there are synsets that do not belong to any specific domain, but they appear in almost all of

Algorithmus 1: Structural Disambiguation algorithm

Input: *WordNet lexical Database and its extensions if any*
$T = [t_i \;\; i = 1..Ncont]$ *the set of the terms to be disambiguated;*
$R = [r_k \;\; k = 1..Nrel]$ *the set of the structural relationships linking two different terms* t_i *and* t_j*;*
Variables:
$S_i = [s_{iy} \;\; y = 1..Nsyn_i]$ *the set of all possible synsets related to the term* t_i*;*
$F_{il} = [f_{yw} \;\; y = 1..Nsyn_i, w = 1..Nf_il]$ *the set of synsets linking by a chain of hypernym relationships of length* l *to one of the synsets* $\in S_i$*;*
$ANNOT_i = [syn_{iz} \;\; z = 1..Ncsyn_i]$ *the set of synsets chosen by the algorithm to disambiguate the term* t_i*;*
for all $r_{ij} \in R$ that link two terms $t_i, t_j \in T$ **do**
  if r_ij is a BT relationship **then**
    determine S_i and S_j;
    initialize $l = 1$ and $annotation = false$;

    repeat
      determine F_{il};
      if F_{il} not empty **then**
        for all s_{jz} in S_j **do**
          if exist a $f_{yw} = s_{jz}$ where $s_{jz} \in S_j$ and $f_{yw} \in F_{il}$ **then**
            insert the synset s_{iy} in $ANNOT_i$;
            insert the synset s_{jz} in $ANNOT_j$;
            set $annotation$=true;
          end if
        end for
      end if
      set $l = l + 1$;
    until $(annotation = true)$ or (no more hypernyms)
  end if
  if r_ij is a SYN relationship **then**
    if $t_i \neq t_j$ **then**
      set $ANNOT_i$ = $ANNOT_j$ = $S_i \cap S_j$;
    end if
  end if
end for
Output: *different set* $ANNOT_i$ *for each term* t_i *that have a structural relation in* R*.*

them. For this reason, a *factotum* domain has been created. This domain, basically, includes the following types of synsets: generic synsets (e.g., "$Man_{\#1}$" - an adult male person), stop sense synsets (e.g., colors and numbers) etc. By exploiting the lexical resource WordNet Domains, it is possible to overcome one of the main limitation of WordNet: domains may provide a useful coarse-grained level of sense distinctions. The availability of WordNet Domains[4] allow us to undertake a domain-oriented analysis of the structural or semi-structural data and to implement an effective WSD algorithm based on domain information.

The WND algorithm takes inspiration from the domain-based approach proposed in [Magnini, 2000]. First, it examines all possible synsets associated with a term and extract the domains connected to these synsets. Then, it computes the list of the *more frequent domains* in the schema context. The choice of the number of more frequent domains to consider is still an open problem: they can be manually selected by the designer, otherwise, by default, the algorithm considers

[4] See `http://wndomains.itc.it`

Algorithmus 2: WordNet Domain disambiguation algorithm

Input: *WordNet lexical Database and its extensions if any*
$T = [ti \;\; i = 1..Ncont]$ *the set of the terms to be disambiguated;*
$NMaxDOM$ *the maximum number of domains we want to use in the algorithm.*
Variables:
$S_i = [s_{iy} \;\; y = 1..Nsyn_i]$ *the set of all possible synsets related to the term* t_i*;*
$D_j = [d_k \;\; k = 1..Ndom_j]$ *the set of the possible domains related to the synset* s_j*;*
$DOM = [dom_x \;\; x = 1..Ndom]$ *an ordered set of the domains related to the set of the terms* T*;*
$FreqDOM = [f_x \;\; x = 1..Ndom]$ *the corresponding set of the frequency of the domains related to the set of the terms* T*;*
$ANNOT_i = [syn_{iz} \;\; z = 1..Ncsyn_i]$ *the set of synsets chosen by the algorithm to disambiguate the term* t_i*;*
for all t_i in T **do**
  for all s_{ij} in S_i **do**
    for all d_{jk} in D_j **do**
      if $d_{jk} \in DOM$ **then**
        increase the $FreqDOM_k$;
      else
        insert the domain d_jk in DOM and set $FreqDOM_k = 1$;
      end if
    end for
  end for
end for
for all t_i in T **do**
  if t_i is a monosemic term **then**
    $ANNOT_i = syn_{ij}$;
  else
    set $annotation = false$;
    for $x = 1$ to $NMaxDOM$ **do**
      for all s_{ij} in S_i **do**
        if d_x is contained in D_j **then**
          insert the synset syn_{ij} in the $ANNOT_i$;
          set $annotation = true$;
        end if
      end for
      if $annotation = true$ **then**
        BREAK the cycle FOR;
      end if
    end for
  end if
end for
for all t_i in T **do**
  if $ANNOT_i$ is empty **then**
    for all s_{ij} in S_i **do**
      if $factotum$ is contained in D_j **then**
        insert the synset syn_{ij} in the $ANNOT_i$;
      end if
    end for
  end if
end for
Output: *a list* DOM *of the more frequent domains, and a set* $ANNOT_i$ *of sysnset that disambiguate each terms in* T.

only the first three most frequent domains.

Finally, it compares the domain list with the domains associated with each term: it chooses all the synsets associated with the more frequent domains.

As described before, in WordNet Domains there is a particular domain called factotum which is the domain associated to synsets that do not belong to any specific domain and, as described in [Magnini et al., 2002], in most cases it is

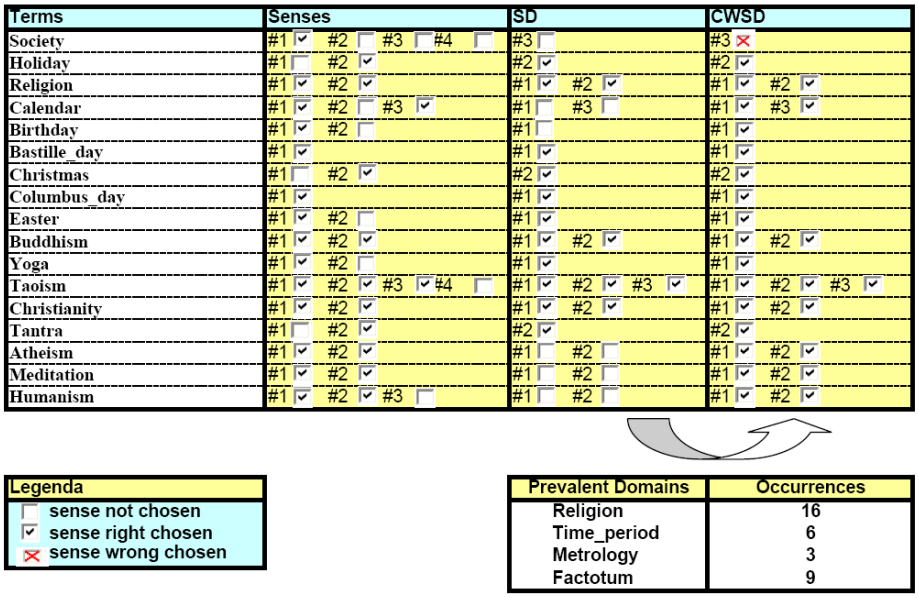

Figure 4.5: Evaluation of the CWSD method.

the more frequent domain in a context. Unlike [Magnini, 2000], CWSD uses the factotum domain only when no domain in the *more frequent domain* list is related to the meanings of a term. WND results depend on the context and on the chosen *configuration*. The configuration is the maximum number of domains we select for the disambiguation. The choice of the configuration is delegated to the designer.

In Figure 4.5, the final result of the application of CWSD to hierarchical data sources is shown. In particular, the figure compares the result obtained by using CWSD with the result obtained by using only SD. If we disambiguate by using only the SD algorithm, we obtain the correct senses for only some terms. By using CWSD, we improve the results in two directions: (1) the disambiguation of the terms is more accurate: polysemy leads to have more than one synset associated to a terms, thanks to CWSD we can assign to these terms more than one meaning; (2) moreover, CWSD enriches the Common Thesaurus with new relationships: this is particularly important for the integration task (as shown in Figure 4.4). The only term annotated incorrectly is “Society”: “Society” is associated, by WND, with the factotum domain, but the correct meaning is associated to the “anthropology” domain which is not present in the more frequent domain list.

WSD algorithm	Recall	Precision
SD	8.00%	97.00%
WND	66.62%	69.97%
CWSD	74.18%	74.18%

Table 4.1: Comparing the different WSD algorithms on the Google and Yahoo directories.

4.3 Experimental Evaluation

CWSD has been experimented over real data sources. In particular, the first three levels of two subtrees of the Yahoo and Google directories have been considered ("society and culture" and "society", respectively), for a total of 327 categories (i.e. labels or terms) for Yahoo and 408 for Google.

Table 4.1 compares the disambiguation of the subtrees of the Google and Yahoo directories obtained by using SD, WND, and CWSD.

The annotation results have been evaluated in terms of recall (the number of correct annotations obtained by the method divided by the total number of annotations, i.e. one for each category, as defined in a gold standard) and precision (the number of correct annotations retrieved divided by the total number of annotations retrieved) (see Section 2.3). In the table, the recall and precision values have been obtained by considering an element as properly annotated if the manually selected annotation (i.e. the gold standard) was included in the set of annotations returned by the WSD algorithm (SD, WND, or their combination in CWSD).

The application of SD over the web directories exploits the 792 ISA relationships and allows to obtain 60 annotations, where 58 of them are correct annotations. In this way, we obtain a high precision value but a very low recall. On the basis of our experience, this was caused by the incompleteness of hypernym/hyponym relationships in the WordNet hierarchy.

The results remark that a combined method outperforms the performance of the single algorithms of which it is composed[5].

4.4 Related Work

Works related to the issues discussed in this chapter are in the area of WSD in NLP, lexical annotation in schema/ontology matching and finally the use of WordNet in

[5]In this evaluation, we do not discuss about the configuration chosen, because in general this is delegated to the designer; however, the results have been obtained by considering as context all the terms of the classes. The results of WND and CWSD have been obtained, by selecting the best configuration for the number of more frequent domains.

schema/ontology matching.

4.4.1 WSD in the NLP area

In the literature, several WSD approaches have been proposed [Navigli and Velardi, 2005, Gliozzo et al., 2005, Mihalcea and Moldovan, 2000, Lesk, 1986].

During the WSD process an algorithm can exploits different external knowledge sources. In [Navigli, 2009], a brief overview about the classification of external sources is proposed. The author divides the different sources into two macro categories: *unstructured resources* and *structured resources*.

Unstructured resources include *corpus*, that are collections of texts used for learning language models that can be sense annotated or raw, and *collocation resources*, which register the tendency for words to occur regularly with others.

Structured data sources include *thesauri* (e.g., WordNet and the Rogets International Thesaurus [Roget, 1852]), *machine-readable dictionaries* (e.g., the Oxford Dictionary of English [Soanes and Stevenson, 2003]), *ontologies* (e.g., the SUMO upper ontology [Pease et al., 2002]). WordNet often has been defined as a dictionary, or thesaurus and even as an ontology because it has characteristics common to all these tree groups of sources. For this reason, in the literature, it has often referred to a *computational lexicon*.

WSD algorithms can be classified into the following macro-categories: *supervised* and *unsupervised* approaches and *knowledge-based* approaches.

Supervised algorithms are typically employed in setting where a set of manually hand-labeled instances (called *training set*) is available. Thus, they can be trained using training sets and then applied to classify a set of unlabeled examples (called *test set*). Usually, these approaches use machine-learning algorithms such as Bayesian classifiers [Escudero et al., 2000] decision trees [Quinlan, 1986], decision lists [Rivest, 1987, Yarowsky, 1994], support vector machines (SVMs) [Boser et al., 1992], etc.

Unsupervised algorithms exploit unlabeled and raw corpora or knowledge base, and do not need of any hand-labeled corpus to provide a sense choice for a word in context. They are based on the idea that the same sense of a word will have similar neighboring words [Navigli, 2009]. Co-occurrence graphs [Widdows and Dorow, 2002], and context and word clustering [Pantel and Lin, 2002] are some examples of unsupervised WSD approaches. Moreover, there are a set of unsupervised WSD algorithms that exploit knowledge bases such as WordNet to discover the meaning of a term [Lesk, 1986, Mihalcea and Moldovan, 2000, Pedersen et al., 2004]. Supervised approaches suffer of three main drawbacks:

1. the manual creation of knowledge resources is an expensive and time consuming effort, which must be repeated every time the disambiguation scenario changes (e.g., in the presence of new domains, different languages, and even sense inventories). This is a fundamental problem which pervades the field of supervised WSD, and is called the *knowledge acquisition bottleneck* [Navigli, 2009, Gale et al., 1992];

2. there is a trade-off between how much training data (pre-tagged corpora) is used and the performance of the method;

3. human intervention is required not only to manually annotate the right meaning of a term, but also to select and filter the training data from large corpora.

Unsupervised algorithms have the potential to overcome the previous supervised drawbacks. To the other end, supervised approaches have generally obtained better results than unsupervised methods [Navigli, 2009]. Combination methods have been shown to be an effective way of improving unsupervised WSD performance. In [Brody et al., 2006] an evaluation study on different combination of unsupervised WSD algorithms is presented, and it is shown that combination systems outperform the behavior of the individual algorithms of which they are composed

CWSD is an unsupervised and combined WSD disambiguation algorithm based on WordNet, thus, it does not need of any training data.

4.4.2 Lexical Annotation in Schema Matching

Several works about automatic lexical annotation have been proposed in the literature [Navigli and Velardi, 2005, Navigli, 2009] but only a few of them have been applied in the context of schema/ontology matching [Mandreoli et al., 2005, Bergamaschi et al., 2007a].

In [Bergamaschi et al., 2007a], the authors developed a software tool, MELIS, for enabling an incremental process of automatic annotation of local schemas, which exploits knowledge provided by the initial annotation. On the contrary, CWSD does not need of any initial annotation to disambiguate the schema labels.

H-MATCH [Castano et al., 2006] makes use of linguistic and contextual features of OWL ontologies to compute the affinity value between two concepts. CUPID [Madhavan et al., 2001] implements an algorithm comprising linguistic and structural schema matching techniques, and computing similarity coefficients with the assistance of domain specific thesauri.

Falcon-AO [Jian et al., 2005], a system for matching OWL ontologies, is made of two components, one to perform linguistic matching and the other to perform structure matching.

Some methods rely only on algorithms (*intrinsic methods*), while others make use of external resources such as dictionaries (*extrinsic methods*). Intrinsic methods produce a linguistic normalization of entities in order to represent ontology entities as sets of words that can be compared by string-based techniques. Extrinsic methods exploit external resources to find similarities between terms. However, they open new possible matches between entities because they recognize that two terms can denote the same concept. Unfortunately, since they recognize that the same term may denote several concepts at once, these techniques provide many possible matches.

Unlike these methods, the proposed approach is based on a first step of lexical annotation of ontology/schema elements. It is only after this phase that the similarity between elements is computed, thus overcoming the limitation of extrinsic methods that cannot recognize the meaning of the elements.

To the best of my knowledge, the work presented in [Banek et al., 2008] is the only one that share our approach by introducing WSD techniques in a Data Integration process. In that paper, WSD is presented as the first step in an ontology integration process. The paper presents an approach to automatically disambiguate the meaning of OWL ontology classes by providing lexical annotation with respect to WordNet. The approach associates WordNet synsets with an ontology class and defines an affinity coefficient.

One of the main limitations of this approach (and of our CWSD method) is that it does not make use of normalization techniques to process compound nouns, and this is reflected in a low coverage of the method. Indeed, the disambiguation techniques are not able to annotate most of the ontology elements labeled with non-dictionary words. In Chapter 5, we address this problem by proposing an automatic schema label normalization method that expands abbreviations and acronyms, and enrich WordNet with new compound nouns.

4.4.3 The use of WordNet in Schema Matching

Semantic taxonomies and thesauri such as WordNet, are a key source of knowledge for NLP applications, and provide structured information about semantic relations between concepts [Lin and Sandkuhl, 2008]. I opted to use WordNet because, as previously described, it is the most commonly used English lexical thesaurus for the task of WSD [Navigli, 2009]. However, the CWSD method can be easily adapted to the use of other thesauri which provide a network of semantic relationships among meanings like WordNet does.

Potentially, all matchers that exploit WordNet or some other thesaurus to discover semantic relationships can integrate CWSD and thus refine the semantic relationships (e.g., some of these matchers are CtxMatch [Bouquet et al., 2003], S-Match [Shvaiko et al., 2010], and H-Match [Castano et al., 2006]).

For instance, CtxMatch uses a semantic matching approach that is a sequential composition of two techniques. At the element level it uses WordNet to find initial matching among classes (CtxMatch2 [Bouquet et al., 2006] improves on CtxMatch by handling ontology properties). At the structured level, it exploits description logic reasoners to compute the final alignments. CtxMatch makes an essential use of linguistic resources to identify the meanings of an element, although it does not make any disambiguation on the set of all possible meanings of a element.

Chapter 5

Schema Label Normalization

In this chapter, a method to perform schema label normalization is presented. This work has been realized in collaboration with the Ph.D. student Maciej Gawinecki [1] and the Ph.D. and fellow researcher Laura Po. The normalization method has been implemented in a stand-alone tool called NORMS (NORMalizer of Schemata) [Sorrentino et al., 2011] and has been evaluated in the context of the MOMIS system. However, as described in the following, it may be applied in general in the context of schema mapping discovery, ontology merging, data integration systems, and web interface integration. Moreover, it might be effective for reverse engineering tasks, e.g., when an ER schema needs to be extracted from a legacy database.

This work was partially supported by the "Searching for a needle in mountains of data!" project funded by the Fondazione Cassa di Risparmio di Modena within the Bando di Ricerca Internazionale 2008[2] and it was conducted in the context of the MIUR FIRB NeP4B (Networked Peers for Business) project[3]. The work presented in this chapter has been published in [Sorrentino and Bergamaschi, 2009, Sorrentino et al., 2009, Sorrentino et al., 2010, Sorrentino et al., 2011, Beneventano et al., 2009a, Bergamaschi et al., 2011b].

The rest of this chapter is organized as follows: in Section 5.1, we define the problem of label normalization in the context of schema matching; in Section 5.2.1, a brief overview of the method is given; in Sections 5.2.2, 5.2.3, and 5.2.4, the subsequent phases of the method are described: schema label preprocessing, abbreviation expansion, and CN annotation. In Section 5.3, the method effectiveness is demonstrated with extensive experiments on real-world data sets. Finally, a comparison of our method with related work is presented in

[1] http://www.ibspan.waw.pl/~gawinec/

[2] http://www.dbgroup.unimo.it/keymantic

[3] http://www.dbgroup.unimo.it/nep4b

Section 5.5.

5.1 Problem Definition

As described in Chapter 4 lexical annotation is an effective process to discover lexical relationships among heterogenous data sources. The great majority of lexical annotation approaches use WordNet as external lexical resource to perform WSD of schema labels.

The strength of a thesaurus, like WordNet, is the presence of a wide network of semantic relationships among word meanings, thus providing a corresponding inferred semantic network of lexical relationships among the labels of different schemata. Its weakness, is that it does not cover different domains of knowledge with the same detail and that many domain-dependent words, or *non-dictionary words*, may not be present in it. Non-dictionary words include CNs (e.g., "company address"), abbreviations (e.g., "QTY") and acronyms (e.g., WSD-Word Sense Disambiguation). In the following, we will refer to both abbreviations and acronyms with the term *abbreviations*. The result of automatic lexical annotation techniques is strongly affected by the presence of such non-dictionary words in schemata. For this reason, a method to expand abbreviations and to semantically "interpret" CNs is required. *Schema label normalization* helps in the identification of similarities between labels coming from different data sources, thus improving schema mapping accuracy.

A manual process of schema label normalization is laborious, time consuming and itself prone to errors. Furthermore, when the number of schemata is very large and dynamically growing a manual process is not feasible. In this chapter, we describe a semi-automatic method for the normalization of schema labels that is able to expand abbreviations and acronyms, and to enrich WordNet with new CNs. The proposed approach uses only schema-level information and can thus be used in scenarios where data instances are not available [Rahm and Bernstein, 2001].

For the sake of simplicity, the discussion is limited to the context of schema matching between two data sources, but the method can be generalized to N schemata. Schema label normalization (also called *linguistic normalization* in [Euzenat and Shvaiko, 2007]) is the reduction of the form of each label to some standardized form. With label normalization, we mean the processes of abbreviation expansion and CN interpretation.

Definition 19 (Compound Noun). *A compound noun (CN) is a word composed of two or more words, called CN constituents. It is used to denote a concept, and can be interpreted based on the meanings of its constituents.*

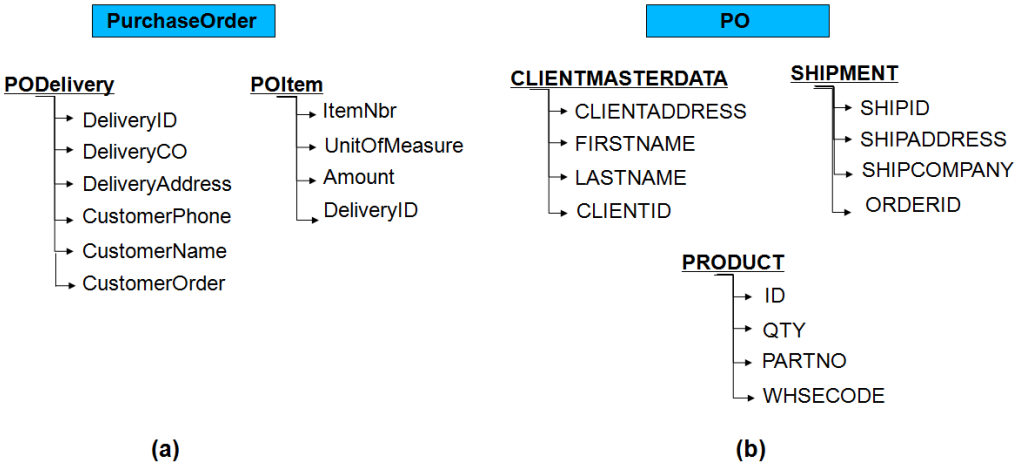

Figure 5.1: Graph representation of two schemata.

Definition 20 (Abbreviation). *An abbreviation/acronym is a shortened form of a word or phrase, that consists of one or more letters taken from the word or phrase.*[4]

Definition 21 (Compound Noun Interpretation). *The interpretation of a CN is the task of determining the semantic relationship that is held among the constituents of a CN.*

Definition 22 (Abbreviation Expansion). *Abbreviation expansion is the task of finding a relevant expansion (long form) for a given abbreviation (short form).*

To give an intuition of the problem, let us consider the example in Figure 6.1 which shows two schemata to be integrated containing many labels in form of non-dictionary CNs (e.g., "CustomerName"), acronyms (e.g., "PO") and abbreviations (e.g., "QTY"). These labels do not have an entry in the lexical dictionary, thus they need to be manually or automatically processed in order to be annotated with respect to WordNet. Schema label normalization improves the schema matching process by reducing the number of discovered *false positive/false negative relationships*.

Definition 23 (False Positive Relationship). *Let* $< s_i, t_j, R >$ *be a lexical relationship. This is a false positive relationship if the concept denoted by the label* s_i *is not related by* R *to the concept denoted by the label* t_j.

[4]The long form that is extracted through abbreviation expansion may not be an entry in WordNet. This issue remains an open problem. For the moment, we limit the method to examine long forms which have an entry in WordNet (e.g., the long form "Number" for the abbreviation "Nbr") or that correspond to CNs (e.g., the long form "Purchase Order" for the abbreviation "PO").

For example, let us consider the two schema labels "CustomerName" and "CLIENTADDRESS", to be found in the schemata "PurchaseOrder" and "PO" respectively (Figure 6.1). If we annotate separately the terms "Customer" and "Name", and "CLIENT" and "ADDRESS", then we will discover a SYN relationship between them, because the terms "Customer" and "CLIENT" share the same WordNet meaning. In this way, a false positive relationship is discovered because these two CNs represent "semantically distant" schema elements.

Other approaches in the literature [Su and Gulla, 2004, Li, 2004] propose to split CNs into separate words and then compare the meaning of the individual constituents in order to compute a similarity score. In these works, the largest the number of common meanings between two CNs, the highest their similarity. Let us consider three schema elements, shown in (Figure 6.1): "CustomerOrderID" in the "PurchaseOrder" schema, and "CLIENTID" and "ORDERID" in the "PO" schema. By using the previously described approach, we discover two SYN relationships between these CNs: one between "CustomerOrderID" and "CLIENTID" as they share the same meaning for the terms "CLIENT" and "Customer", and the term "ID"; and one between "CustomerOrderID" and "ORDERID", as they share the same meaning for the terms "ORDER" and "Order", and for the term "ID". As in both cases the CNs share the meaning of two constituents, these relationships will be assigned the same similarity value. However, the relationship between "CustomerOrderID" and "CLIENTID" is a false positive relationship.

Definition 24 (False Negative Relationship). *Let $< s_i, t_j, R >$ be a lexical relationship. R is a false negative relationship if the concept denoted by the label s_i is related by R to the concept denoted by the label t_j but the schema matching process does not return this relationship.*

Let us consider two corresponding schema labels: "amount" in the "PurchaseOrder" source and "QTY" (abbreviation for "quantity") in the "PO" source (Figure 6.1). Without abbreviation expansion, we would not discover that there exists a SYN relationship between the elements "amount" and "QTY".

5.2 The Schema Label Normalization Method

5.2.1 Overview

As shown in Figure 5.2, the schema label normalization method consists of three steps: (1) schema label preprocessing, (2) abbreviation expansion and (3) CN annotation.

In this section, we briefly analyze the different phases and describe a simple example of the application of the normalization method on the schema element

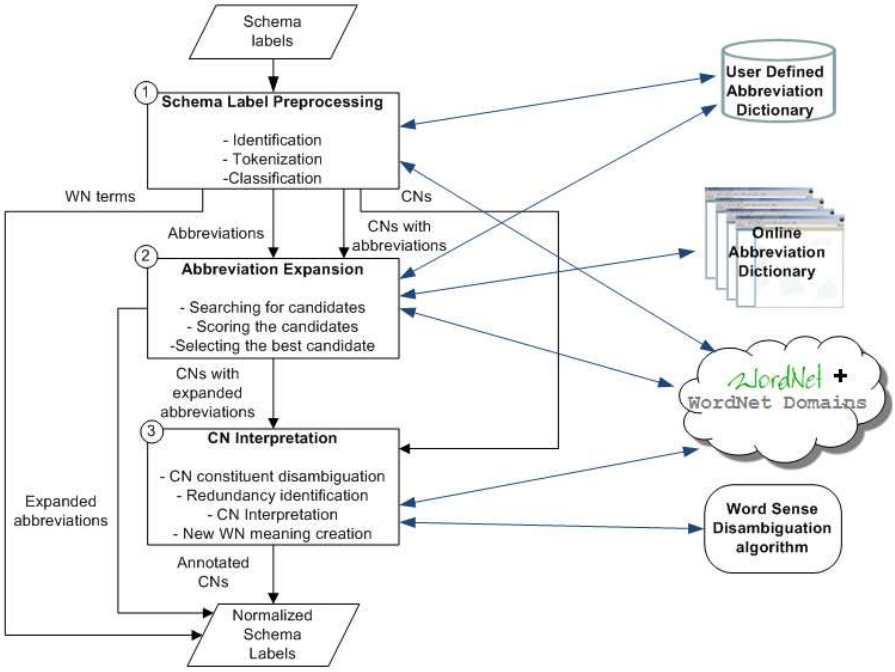

Figure 5.2: Overview of the schema label normalization method.

"DeliveryCO" belonging to the "PurchaseOrder" schema in Figure 6.1.

Schema Label Preprocessing
The input of schema label preprocessing is the set of schema element labels. During this phase, the labels to be normalized are automatically selected (for details see Section 5.2.2). The output of this module are the tokenized labels classified into four groups (as shown in Figure 5.2): *WordNet terms* (i.e. labels having an entry in WordNet which do not need normalization, e.g., "FIRSTNAME") *abbreviations* (e.g., "QTY"), *CNs* (e.g., "PurchaseOrder"), and *CNs with abbreviations* (e.g., "DeliveryCO"). For instance, the schema label "DeliveryCO" is selected for normalization, thus it is tokenized in two single words "Delivery" and "CO" and finally, it is classified as a CNs that contains the abbreviation "CO".

Abbreviation Expansion
The input of abbreviation expansion are the schema labels classified as abbreviations or CNs with abbreviations (as shown in Figure 5.2). During this phase, each abbreviation is expanded with the most relevant long form by using the information provided by the schemata and abbreviation dictionaries (for details see Section 5.2.3). For instance, the abbreviation "CO" of the CN "DeliveryCO", is expanded as "Company".

CN Annotation
The input of CN annotation are the schema labels classified as CNs and CNs with expanded abbreviations (as shown in Figure 5.2). During this phase, the constituents of a CN are annotated with respect to WordNet by applying a WSD (Word Sense Disambiguation) algorithm; then, starting from these annotations a semantic relationship between the constituents (for details see Section 5.2.4) is discovered. For instance, for the CN "Delivery Company", the semantic relationship "MAKE" is selected. In the end, a new WordNet meaning for the CN is created and inserted in WordNet.

In conclusion, the output of the normalization method applied on the schema label "DeliveryCO" is the normalized label "Delivery Company" with its interpretation ("Company MAKE Delivery"). In the following sections, each step will be described in details.

5.2.2 Schema Label Preprocessing

To perform schema label normalization, schema labels need to be preprocessed. Schema label preprocessing is divided into three main sub-steps (as shown in Figure 5.2): (1) identification, (2) tokenization, (3) classification.

Step 1. Identification
The goal is to identify those schema labels that do not have an entry in WordNet and thus need to be normalized. CNs (e.g., "company name") and abbreviations (e.g., "GDP", standing for "Gross Domestic Product") having an entry in WordNet need no normalization. Moreover, a set of exceptions (called *schema standard abbreviations* in Section 5.2.3) that, although they have an entry in WordNet are mostly used as abbreviations in the context of schemata (e.g., "id", which is a state in the Rocky Mountains in WordNet, is often used as a short form of "identifier" in schemata) have been identified. All such abbreviations are gathered in a "user-defined dictionary" and automatically identified for normalization.

Definition 25 . *A label needs to be normalized if it occurs on the list of schema standard abbreviations or if it does not have an entry in WordNet.*

Step 2. Tokenization
This step tokenizes the previously identified labels by using one of the approaches described in [Feild et al., 2006]: the *simple* (ST) approach is based on camel case and punctuation; the *greedy* approach hands also multi-word labels without

clearly defined word boundaries (e.g., "WHSECODE"). The latter uses simple tokenization to split the label around explicit word boundaries into single words and then for each non-dictionary word iteratively looks for the biggest prefixing/suffixing dictionary word or schema standard abbreviation. Two alternative variants of greedy tokenization are considered: GT/WordNet, which makes use of WordNet to identify dictionary words, and GT/Ispell, that makes use of the English word list in Ispell[5].

Step 3. Classification
This step classifies tokenized labels into four groups: dictionary words that exist in WordNet, abbreviations that need expansion, CNs that need interpretation, and CNs containing abbreviations that need both expansion and interpretation. The same heuristic rules as those used during the identification step are applied here. However, abbreviations might be expanded as WordNet terms (e.g., "CO" as "Company") or CNs (e.g., "CO" as "Company Order"). Because of this, the classification step is performed again after abbreviation expansion in order to identify non-dictionary CNs.

For instance, let us assume we are preprocessing the "DeliveryCO" label (shown in Figure 6.1). This label is neither a dictionary word nor a schema standard abbreviation and therefore it needs to be normalized. The tokenization, based on camel case, splits it into: "Delivery" and "CO" words. The classification identifies "Delivery CO" as a CN with the abbreviation "CO".

5.2.3 Abbreviation Expansion

The problem of abbreviation expansion cannot be reduced to a simple substitution, as many abbreviations are ambiguous, i.e. the same abbreviation may refer to different concepts (e.g., "CC" may mean "Credit Card", "Country Club" or "Carbon Copy"). Moreover, a schema can contain both *standard* (e.g., "Nbr" (Number)) and *ad hoc* abbreviations (e.g. "Del_a" (Delivery address)).

In order to identify potential expansions, the method exploits four *expansion resources*: Local Context (LC), Complementary Schemata (CS), Online abbreviation Dictionary (OD), and Local abbreviation Dictionary (LD). For example, for the "CO" abbreviation contained in "DeliveryCO", the local context is its schema "PurchaseOrder" and the schema "PO" is the complementary schema. LC and CS are particularly relevant for expanding ad hoc abbreviations. It is common practice to abbreviate class name in its attribute name (for instance, "SHIPMENT" table in Figure 6.1 contains "SHIPADDRESS", "SHIPDATE" and "SHIPCOM-

[5]Ispell is a popular tool for the correction of spelling errors: `http://wordlist.sourceforge.net/`.

PANY" attributes, where "SHIP" is an abbreviation for "SHIPMENT"). LD is initially bootstrapped with standard schema abbreviations from schema design guidelines for the OTA standard[6]. Moreover, the designer can enrich this user-defined dictionary by inserting new standard abbreviations. As OD, the method uses "Abbreviations.com".

To handle different types of abbreviations the algorithm uses the four aforementioned resources. The abbreviation expansion algorithm can be divided into three main steps (as shown in Figure 5.2): (1) searching candidate long forms; (2) scoring the candidate long forms; (3) selecting the most appropriate long form.

The algorithm looks for possible expansions in LC and CS using the abbreviation patterns proposed in [Hill et al., 2008]. Moreover, the algorithm tries to find an entry for a given abbreviation into OD and LD. For instance, for the abbreviation "CO" the following expansions are identified: from OD {"Company", "Colorado", and "Check Out"}, from LC no results, from CS schemata {"Company"}. Next, the algorithm merges lists of long form candidates into a single one: {"Company", "Colorado", "Check Out"}.

For each identified expansion exp_i, the algorithm computes a combined score $sc(exp_i) \in [0, 1]$ and suggests the top-score expansion. As shown in Figure 5.2, for the abbreviation "CO" the tool suggests the expansion "Company". The score $sc(exp_i)$ is computed by combining scores from the single resources:

$$sc(exp_i) = \alpha_{SD} \cdot sc_{SD}(exp_i) + \alpha_{CS} \cdot sc_{CS}(exp_i) + \\ \alpha_{LC} \cdot sc_{LC}(exp_i) + \alpha_{OD} \cdot sc_{OD}(exp_i)$$

where sc_{SD}, sc_{CS} and sc_{LC} are equal to 1 if exp_i is found in the given resource or 0 otherwise; and sc_{OD} is computed as described in [Sorrentino et al., 2010]. $\alpha_{SD}+\alpha_{CS}+\alpha_{LC}+\alpha_{OD} = 1$ are the weights of resource relevance. The algorithm uses as default weights $\alpha_{SD} = 0.4$, $\alpha_{LC} = 0.3$, $\alpha_{CS} = 0.2$ and $\alpha_{OD} = 0.1$. These weights were selected after an evaluation of the abbreviation expansion phase on several real schemata. The weights can also be manually set up by the designer.

For more details about the abbreviation expansion phase, please refer to [Gawinecki, 2011].

5.2.4 CN Annotation

In the NLP (Natural Language Processing) literature different CN classifications have been proposed [Plag, 2003, Levi, 1978]. In this work, we use the classification introduced in [Plag, 2003], where CNs are classified in four distinct categories: *endocentric*, *exocentric*, *copulative*, and *appositional*.

[6]OpenTravel Alliance Xml schema for travel industry. Available online at `http://www.opentravel.org/`.

Definition 26 (Endocentric CN) *Endocentric CNs consist of a head (i.e. the categorical part that contains the basic meaning of the whole CN) and modifiers, which restrict the meaning of the head. An endocentric CN exhibits a* modifier-head structure*, where the head noun occurs always after the modifiers. Endocentric CNs are often not included in dictionaries, but they can be interpreted by using the knowledge about their constituents. Based on this property, endocentric CNs can be also defined as* transparent *[Barker and Szpakowicz, 1998].*

On the contrary, exocentric CNs do not have a head and are usually represented by a single word. Their meaning cannot be inferred from the meaning of its constituents (e.g., "pickpocket", "loudmouth") and their semantics is deviant: for example, a "white-collar" is neither a kind of collar nor a white thing, but a particular socioeconomic status.

Copulative compounds are CNs which have two semantic heads (e.g., "bitter-sweet", "sleepwalk"). The constituents of this kind of CNs are characterized by the fact that none of the two constituents seems in any sense more important than the other.

Finally, appositional compounds refer to CNs that have two (contrary) attributes (e.g., "actor director", "maid servant").

In this work, we only consider endocentric CNs. Our restriction is motivated by the following observations: (1) the vast majority of CNs in schemata fall in endocentric category; (2) endocentric CNs are the most common type of CNs in English; (3) exocentric and copulative CNs, which are represented by a unique word, are often present in a dictionary; (4) appositional CNs are not very common in English and less likely used as elements of a schema. Moreover, we performed a set of tests in order to verify that endocentric CNs are also the main category of CNs in the context of structured and semi-structured data sources. Our tests showed that, on average, endocentric CNs account for 78% of the total number of CNs appearing in a given source[7].

The constituents of endocentric compounds are noun-noun or adjective-noun, where the adjective derives from a noun (e.g., "Asian food", where the adjective "Asian" derives from the noun "Asia"). The method considers endocentric CNs composed of only two constituents, because CNs consisting of more than two words can be constructed recursively by *bracketing* them into pairs of words and

[7]These tests have been performed on several real data sources, containing several CNs, in different domains and formats (relational and XML): the first three levels of a subtree of the Yahoo and Google directories ("society and culture" and "society", respectively); three schemata of an application scenario (ICT-A partner search) of the NeP4B project, available at `www.dbgroup.unimo.it/nep4b/NeP4BScenarioICTA.xml`; the test schemata number 6 (RDB vs. Star datawarehouse schema) and the test schemata number 5 available at (CIDX and Excel) used in [Madhavan et al., 2001], available at `http://dit.unitn.it/~accord/Experimentaldesign.html`

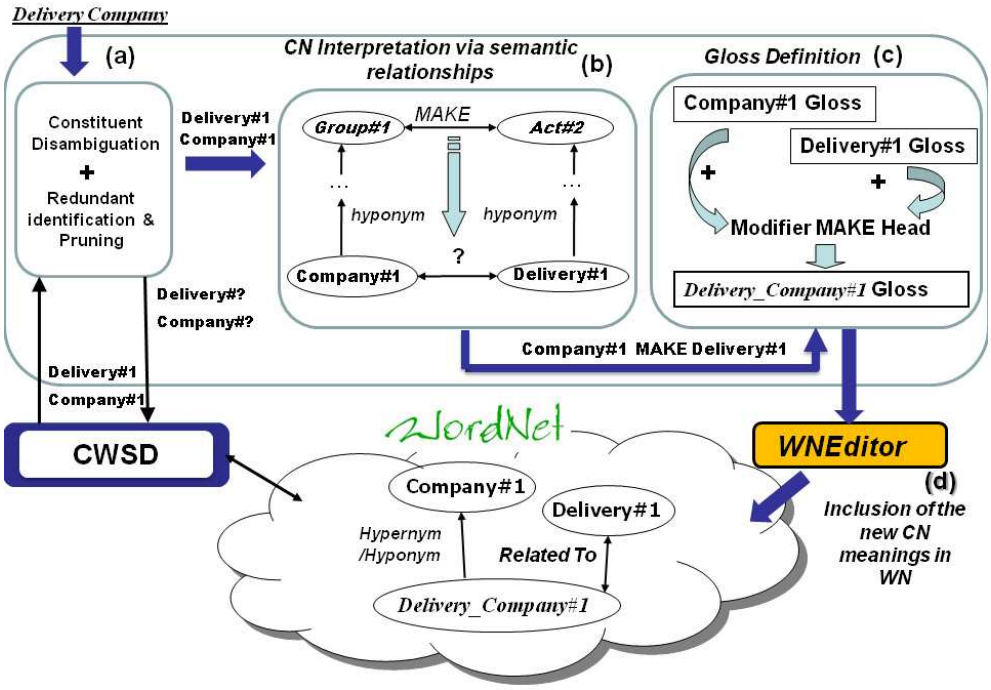

Figure 5.3: The CN annotation process.

then interpreting each pair. In the following, we will refer to endocentric CNs simply as CNs.

Our method can be summed up into four main steps (as shown in Figure 5.2): (1) CN constituent disambiguation; (2) redundant constituent identification; (3) CN interpretation via semantic relationships; (4) creation of a new WordNet meaning for a CN.

Step 1. CN constituent disambiguation. In this step, the WordNet synset of each constituent is chosen in two moves:

1. *Part of speech tagging*: this step performs the part of speech analysis of CN constituents, in order to identify the syntactic category of its head and modifier. It uses the Stanford part of speech tagger [Toutanova and Manning, 2000][8]. If the CN does not fall under the endocentric syntactic structure (noun-noun or adjective-noun where the adjective derives from a noun), then it is ignored. For example, the constituents of the CN "Delivery Company" both belong to the noun syntactic category;

[8]The Stanford part of speech tagger is freely available at `http://nlp.stanford.edu/software/tagger.shtml#Download`

2. *Disambiguating head and modifier*: this step is part of the general lexical disambiguation problem. By applying the CWSD method (see Chapter 4) each word is automatically mapped onto its corresponding WordNet synset.

We agree with [Kim and Baldwin, 2005] that WSD can significantly improve the accuracy of CN interpretation.

For example, as shown in Figure 5.3a, for the schema elements "DeliveryCO", previously expanded as "Delivery Company", we obtain the two constituents annotated with the correspondent WordNet meanings (i.e. "$Company_{\#1}$" and "$Delivery_{\#1}$"[9]).

Step 2. Redundant constituent identification and pruning. During this step, the method controls whether a CN constituent is a *redundant word*.

Definition 27 (Redundant Word). *A redundant word is a word that does not contribute new information, as its semantics contribution can be derived from the schema or from the lexical resource.*

The typical situation in a schema is when the name of a class is a part of its attribute name, see for instance the "SHIPADDRESS" attribute of the "SHIPMENT" class (Figure 6.1). The "SHIPADDRESS" attribute is expanded in the abbreviation expansion phase as "SHIPMENT ADDRESS". As a result, the constituent class name is not considered, because the relationship that hold among a class and its attributes can be derived from the schema. Moreover, a redundant word exists when one of the constituents is a hypernym/hyponym of the other, e.g., the CN "mammal animal" where the meaning associated by CWSD to the head "animal" is a hyponym of the meaning associated to the modifier "mammal". The information that "a mammal is a kind of animal" is redundant because it can be directly derived from the WordNet hierarchy.

Step 3. CN interpretation via semantic relationships. This step concerns selecting from a set of predefined relationships the one that best captures the semantic relation between the meanings of a head and a modifier.

In the literature, several sets of semantic relationships have been proposed. Levi defines a set of nine possible semantic relationships to interpret CNs [Levi, 1978] (shown in Table 5.1) In contrast, Finin claims an unlimited number of semantic relationships [Finin, 1980]. In [Plag, 2003] the problem of identifying a set of relationships is sidestepped: the semantics of a

[9] #1 is a standard notation used in the WordNet literature to indicate the first WordNet meaning associated to a word; a similar way, the second WordNet meaning for a work will be indicated as #2.

Relationship	**Definition**	**Example**
MAKE 1	H *MAKE* M	honey bee
MAKE 2	M *MAKE* H	daisy chains
CAUSE 1	H *CAUSE* M	flu virus
CAUSE 2	M *CAUSE* H	snow blindness
HAVE 1	H *HAVE* M	college town
HAVE 2	M *HAVE* H	company assets
USE	H *USE* M	water wheel
BE	H *BE* M	chocolate bar
IN	H *IN* M	mountain lodge
FOR	H *FOR* M	headache pills
FROM	H *FROM* M	bacon grease
ABOUT	H *ABOUT* M	adventure story

Table 5.1: The Levi's set of semantic relationships (M= modifier, H= head).

CN is then simply the assertion of an unspecified relationship between its constituents. Other sets of semantic relationships to interpret CNs are proposed in [Kim and Baldwin, 2005, Moldovan et al., 2004, Rosario and Hearst, 2001].

The choice of the set of semantic relationships for CN interpretation has frequently been discussed in the NLP literature [Nastase et al., 2006]: one criticism is that the variety of relationships is so great that listing them is impossible; moreover, when the semantic set is too wide often it is difficult to say which relationship should be applied to a certain CN, and there are many cases where many relationships seem appropriate [Ó Séaghdha, 2008].

The proposed method uses the Levi's semantic relationship set, whose nine types of relationship are a common subset to several approaches of CN interpretation. A more detailed explanation of the reasons for this decision will be provided in the following.

Following [Fan et al., 2003], the method is based on an assumption: the semantic relationship between the head and modifier of a CN is derived from the one that hold between their top level WordNet nouns in the WordNet noun hierarchy.

Top levels of a lexical resource include concepts that make important ontological distinctions, and although they contain relatively few concepts, these concepts are important for the task of CN interpretation and cover all different conceptual and lexical domains present in the lexical resource. In particular, the WordNet nouns hierarchy has been proven to be very useful in the CN interpretation task [Nastase et al., 2006]. The top level concepts of the WordNet hierarchy are the 25 *unique beginners* (e.g., act, animal, artifact etc.) for WordNet English nouns defined by Miller in [Miller et al., 1990] (see Figure 5.4). In particular, in WordNet a unique beginner is a noun synset (i.e. a synset belonging to the

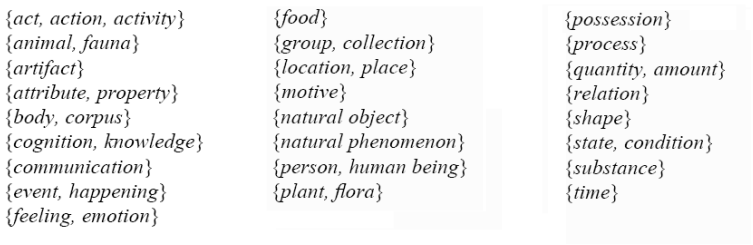

Figure 5.4: The 25 unique beginners for the WordNet noun hierarchy.

noun syntactic category, thus belonging to the WordNet noun hierarchy) with no hypernymy synsets. These unique beginners are related to other synsets through hyponym relationships (e.g., in Figure 5.3b the unique beginner "$Group_{\#1}$" is related through a chain of hyponym relationships to the synset "$Company_{\#1}$"), and they cover distinct conceptual and lexical domains [Miller et al., 1990]. As these unique beginners cover all noun synsets in WordNet, by annotating all the possible combinations of unique beginners we can infer the semantic relationship for all the possible pairs of noun-noun (and adjective-noun where the adjective derives from a noun) in WordNet.

The decision to use Levi's set should now appear clear: an excessive granularity of the set of semantic relationships is not suitable to interpret the relevant pair of WordNet unique beginners: on this hierarchy level, it is difficult to express fine differences among the relationships, and a very detailed and fine interpretation of CNs is not required in the context of semi-automatic data integration. Moreover, as shown in Table 5.1, each Levi's relationship is associated with a definition (i.e. the paraphrase of the relationship) which can profitably be exploited during the process of WordNet meaning creation (see below Step 4).

For each possible pair of unique beginners, the relationship from Levi's set that best describes the meaning of the pair has been associated. For example, for the unique beginner pair "group and act" the Levi's relationship MAKE (e.g., "group MAKE act") is chosen, which can be expressed as "a group that performs an act". In this way, as shown in Figure 5.3b, the label "Delivery Company" can be interpreted with the MAKE relationship, because "Company" is a hyponym of "group" and "Delivery" is a hyponym of "act". Our method required an initial human intervention aimed at associating the Levi's relationship to each pair of unique beginners: as WordNet has 25 unique beginners, we associated a semantic relationship from Levi's set to 625 pairs of unique beginners[10]. This human intervention may be considered acceptable, when compared with the effort required by traditional approaches based on a pre-tagged cor-

[10]The 625 pairs of unique beginners were annotated by two Ph.D student; while a third annotator intervened in cases of disagreement.

pora [Moldovan et al., 2004, Su Nam Kim, 2008], as will be discussed in Section 5.5. Moreover, the method is independent from the domain under consideration and can be applied to any thesaurus providing a wide network of hyponym/hypernym relationships between meanings.

Step 4. Creation of a new WordNet meaning for a CN. During this step, the method automatically creates a new WordNet meaning for a CN starting from the meanings of its constituents and using the discovered relationship. In this step, we have following two sub-steps:

1. *Gloss definition*: a WordNet *gloss* is the definition and explanation in natural language of the meaning of a term[11]. Starting from the relationship associated to a CN and exploiting the glosses of the CN constituents, the method creates the gloss to be associated to a CN. To create a new gloss for the CN, we need to express in natural language the meanings of a semantic relationship. As previously described, CNs are interpreted according to Levi's relationships, which can be used directly in the gloss. As shown in Figure 5.3c, the glosses of the constituents "Company" and "Delivery", are joined by means of Levi's relationship MAKE. The new gloss for the CN "Delivery Company", thus, becomes "An institution created to conduct business MAKE the act of delivering or distributing something".

2. *Inclusion of a new CN meaning in WordNet*: the insertion of a new CN meaning into the WordNet hierarchy implies the definition of its relationships with the other WordNet meanings. As the concept denoted by a CN is a subset of the concept denoted by the head, it is possible to assume that a CN inherits most of its semantics from its head [Plag, 2003]. Starting from this consideration, it is possible to infer that the CN is related, in the WordNet hierarchy, to its head by a hyponym relationship. Moreover, the CN semantics related to its modifier is represented by inserting a generic relationship RT (*Related term*), corresponding to the WordNet relationships *member meronym*, *part meronym*, *substance meronym*. As RT is a bidirectional relationship, it also includes the inverse relationships *part holonym*, *part holonym* and *substance holonym*. During this step, the method automatically controls if other new CNs with the same head have been previously inserted in WordNet. For example, if we need to insert in WordNet a new meaning for the CN "student name" and we have previously inserted the CN "person name", we control if there exists a *hyponym/hypernym* relationship between the modifiers "person" and "student". In this case, we

[11] In WordNet, "gloss" is the standard term. Each synset, is associated with one and only one gloss which can optionally include some example sentences.

insert the new meaning for the CN "student name" as a hyponym of the already inserted CN "person name". However, the insertion of these two relationships is not sufficient; it is also necessary to discover the relationships of the new inserted meaning with respect to the other WordNet meanings. To this end, the method uses the WNEditor tool to create/manage the new meaning and to set relationships between it and the existing WordNet meanings [Beneventano et al., 2003b]. The WNEditor automatically retrieves a list of candidate WordNet meanings sharing similarities with the new meaning. The designer is then asked to explicitly declare the type of relationship (e.g., hyponymy or meronymy[12]) to be established between the new meaning and the others, if any. Figure 5.3d illustrates this step with and example.

5.3 Experimental Evaluation

Our evaluation goals were as follows: (1) measuring and explaining the performance of our method, (2) checking whether our method improves the *lexical annotation* process and finally (3) estimating the effect of schema label normalization on the *lexical relationship discovery* process. In particular, to achieve the last two goals, the output of the schema normalization method was given as input of the MOMIS system. We tested the effectiveness of our method in several real integration scenarios.

Data Sets. To evaluate our method, we used the following five data sets: (1) GeneX, (2) Mondial, (3) Amalgam (an integration benchmark for bibliographic data [Miller et al., 2001]), (4) TCP-H, and (5) PurchaseOrder (which contains Paragon schema and the OpenTrans e-business standard schema). Each data set consists of two schemata that need to be integrated. These data sets[13] have been used in several schema matching experiments [Aumueller et al., 2005, Chiticariu et al., 2008]. Figure 5.5 summarizes the features of the schemata. We chose these data sets for the following reasons: they are particularly suitable to evaluate schema normalization as they contain several non-dictionary words; they represent different application domains; finally, they contain both relational (RDB) and XML schemata (with different XML formats: XML schema, DTD, XDR).

Experimental methodology. To assess the quality of our method, gold standards were created for each normalization phase as well as for the lexical anno-

[12] For a complete list of the WordNet relationships see `http://wordnet.princeton.edu/man/wngloss.7WN.html`.

[13] All the data sets are publicly available at `http://queens.db.toronto.edu/project/clio/index.php#testschemas` and `http://dbs.uni-leipzig.de/Research/coma_index.html`

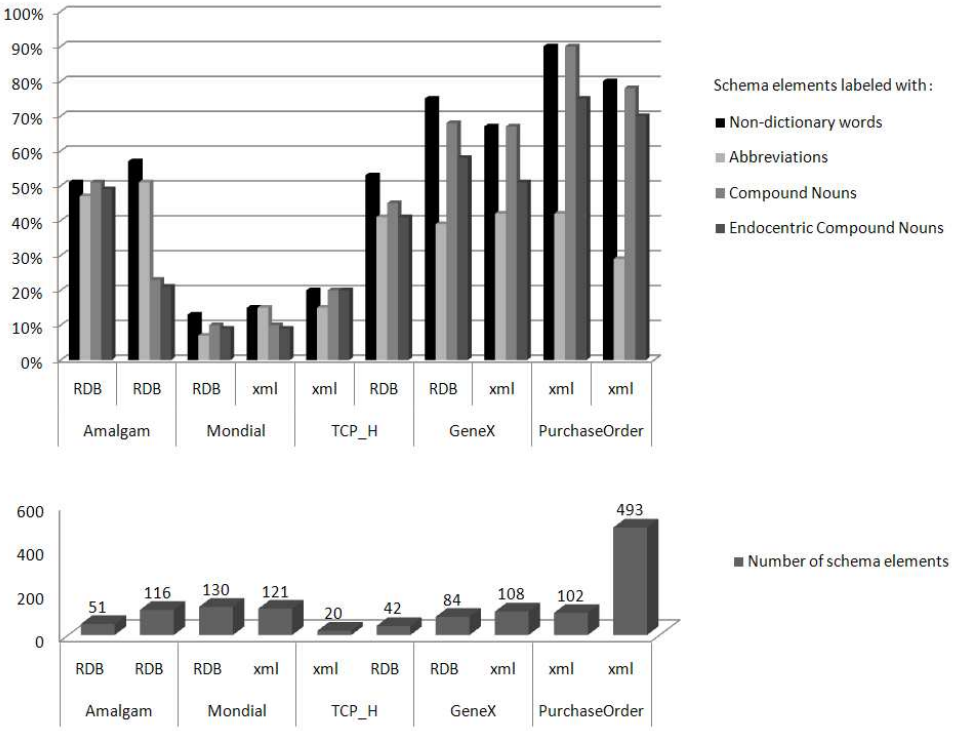

Figure 5.5: Feature summary of the data sets.

tation and the lexical relationship discovery process. The gold standards were manually generated by a human expert. The results obtained in each experiment were compared with respect to the corresponding gold standard.

External resources. The experiments were carried out by using the lexical database WordNet 2.0, its extension WordNet Domains 3.2 and the Abbreviations.com dictionary as external sources.

5.3.1 Evaluating Normalization

The normalization method consists of different phases. Since the errors of each phase can accumulate in subsequent phases, we evaluated the performance of each phase first separately and then as a whole.

Schema Label Preprocessing Evaluation

In order to perform a complete evaluation of this phase, we evaluated tokenization separately and then identification and classification together, as they are based on the same heuristics (see Section 5.2.2). We evaluated tokenization only for

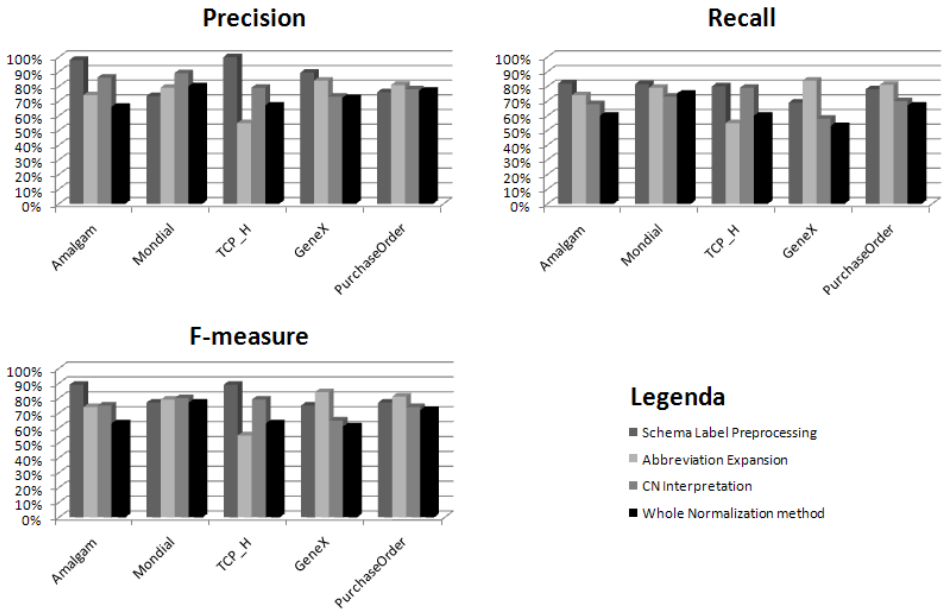

Figure 5.6: Performance of schema normalization.

labels identified for normalization in the gold standard. The F-Measure of the tokenization approach was affected by the nature of the schema labels, and reached an average value of 86%. Identification achieved 96% recall (averaged over all evaluated schemata), meaning that 4% of the labels with abbreviations were not recognized. Amalgam and TCP-H schemata contain such difficult abbreviations, e.g., "RID" standing for "Record Identifier", which is also a synonym of the verb "free" in WordNet. The same reason caused a more marked drop in recall (on average 78%) for the classification of manually tokenized labels, especially for GeneX (54%). Finally, a number of errors were caused by the presence of stop words (e.g., "to") in schema labels that do not have an entry in WordNet.

Abbreviation Expansion Evaluation

We evaluated automatic abbreviation expansion starting from the manually preprocessed labels (gold standard). We used the default relevance weights for expansion resources described in Section 5.2.3 ($\alpha_{UD} = 0.4$, $\alpha_{LC} = 0.3$, $\alpha_{CS} = 0.2$, $\alpha_{OD} = 0.1$). During the evaluation, an expanded abbreviation was considered a TP (i.e. *correctly expanded*) if the automatic expansion was the same as the one returned by the gold standard; if not, it was considered a FP expansion. FN expansions were all the expansions rendered by the gold standard but not returned by the algorithm. The results of the algorithm are presented in Figure 5.6. The

algorithm provided correct expansions on average for 74% of the abbreviations.

For more detail about the evaluation of the abbreviation expansion algorithm please refer to [Gawinecki, 2011]

CN Annotation Evaluation

In this phase the gold standard is represented by the manual interpretation of all CNs contained in the data sets. During the evaluation, a CN was considered a TP (i.e. *correctly interpreted*) if the automatically selected Levi's relationship was the same as the one returned by the gold standard. If not, it was considered a FP interpretation. FN interpretations were obtained for all the interpretations contained in the gold standard but not returned by our method.

As shown in Figure 5.6, the CN interpretation method obtained good results on both precision (on average 81%) and recall (on average 70%), and consequently on F-Measure (on average 75%). In all data sets, the recall value was affected by the presence in the schemata of non-endocentric CNs (such as "ManualPublished", "isMember" or "InProceedings") that our method is not able to interpret. Moreover, the GeneX, PurchaseOrder, and Mondial data sets also contain schema elements labeled with digits (e.g., "sea 2" or "treatment list sequence 1"). As digits are dictionary words in WordNet, these CNs were automatically considered endocentric and interpreted incorrectly by our method. These incorrect interpretations mainly stem from the fact that the problem of the presence of digits in schema labels needs to be treated in a different way.

The poorest performance was obtained for the GeneX data set. There are two main reasons for this: first, GeneX contains several complex CNs composed by three or four constituents (e.g., "GEML", expanded as "gene expression markup language", or "AM_FACTORVALUE", expanded as "array measurement factor value") which are difficult to interpret even for a human expert; second, in this source the number of non-endocentric CNs is greater than in the other data sets (20% of the total number of CNs in GeneX). On the other hand, for the PurchaseOrder data set we obtained good results on both precision and recall, although the set contains several complex CNs. The pruning step (see Section 5.2.4 - Step 2) significantly helps in reducing the complexity of CNs (e.g., the attribute label "ORDERCHANGE_ITEM_LIST" of the class "ORDERCHANGE" in the Paragon schema is reduced to the CN "ITEM_LIST").

Schema Label Normalization Evaluation

The input of the whole schema normalization method is the set of the original schema labels and the output is the set of normalized schema labels. The method has been evaluated with the GT/Ispell tokenization method that achieved the best

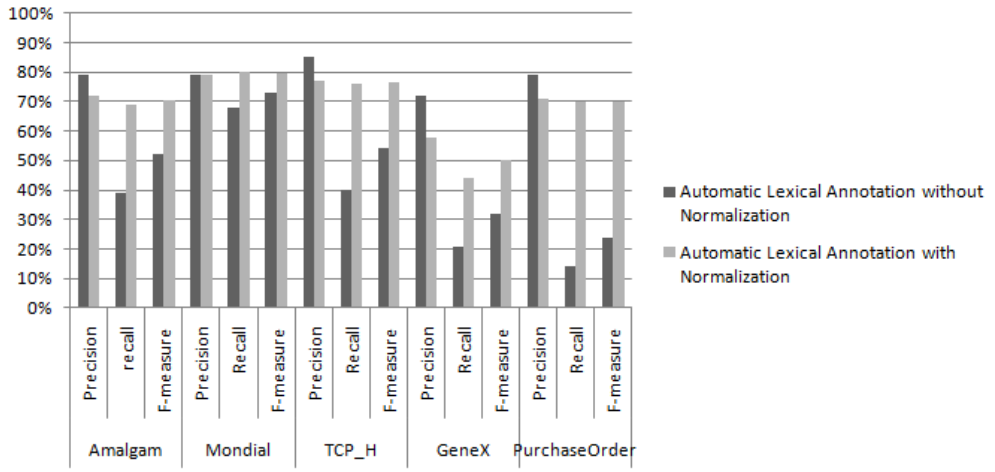

Figure 5.7: Lexical annotation evaluation.

results for the considered schemata. Figure 5.6 shows the result of the whole method. We obtained good results for both precision and recall (the average precision is 63% and the average recall is 72%).

5.3.2 Lexical Annotation Evaluation

The evaluation of the lexical annotation process was carried out by comparing the annotations returned by CWSD (starting from automatically normalized schemata) with respect to the gold standard. The gold standard was created by manually annotating each schema element with respect to WordNet starting from manually normalized schemata. During the evaluation, a schema element annotation was considered a TP (i.e. *correctly annotated*) if the WordNet meaning selected by CWSD was the same as the one returned by the gold standard; otherwise, it was considered an FP annotation. We obtained FN annotations when the schema elements were incorrectly annotated or not annotated at all.

Figure 5.7 shows the result of lexical annotation performed by CWSD *with and without* our normalization method. In this experiment the poorest performance was obtained once again on the GeneX data set. However, the results show that, by using our normalization method, we are able to significantly improve the F-Measure for each data set. In particular, the improvement is more evident when schemata contain several non-dictionary words (e.g., the Amalgam and PurchaseOrder data sets). Without schema normalization CWSD obtains a low recall value for each data set, because many CNs and abbreviations are present in the schemata. The application of our method increases recall while preserving a good precision.

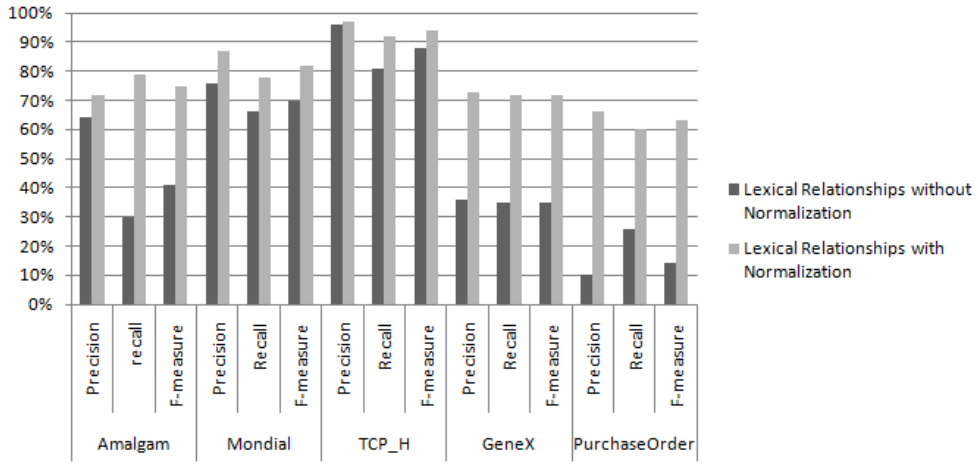

Figure 5.8: Lexical relationship discovery evaluation.

5.3.3 Lexical Relationship Discovery Evaluation

To create the gold standard for the lexical relationship discovery process, we manually mapped the schema elements with the appropriate lexical relationships. During the evaluation, a lexical relationship was considered a TP (i.e. a *correct lexical relationship*) if it was present in the set of manually determined lexical relationships (gold standard). If not, it was considered a FP relationship. FN relationships included all the relationships that were not returned by the automatic lexical relationship discovery process. During this evaluation, we decided to consider only "synonymy" (SYN) and "hypernymy/hyponymy"(BT/NT) relationships and not the "related term" (RT) relationships. This decision is supported by two main observations: RT relationships have minor relevance with respect to BT and SYN relationships; moreover, when the number of schema elements to be mapped becomes very large, the creation of the gold standard including RT relationships becomes difficult and error-prone even for a human designer.

Figure 5.8 shows the result of the lexical relationship discovery process with and without normalization. In the first case, the lexical relationship discovery process was performed without abbreviation expansion and by considering the constituents of a CN as single words with an associated WordNet meaning. Without schema label normalization we discovered few lexical relationships; the low value of precision was due to the presence of many false positive relationships. Moreover, recall was particularly low because many lexical relationships between schema elements labeled with abbreviation were not discovered. Hence, in general, the lexical relationship discovery process without normalization establishes the incorrect lexical relationships between the schema labels that share some words. Instead, with our method we are able to significantly improve recall and

precision (and F-Measure).

Another observation to be drawn from the graph is that, surprisingly, the lexical relationship discovery process outperforms the lexical annotation process. There are different reasons for this: several incorrectly normalized (and consequently incorrectly annotated) schema labels are not related to any element in the other schema to be integrated. For instance, in the TCP_H data sets, the labels "mfgr" and "ph" (abbreviations for "manufacturer group" and "phone") are normalized incorrectly and they are not connected to any element in the complementary schemata. The same holds true for the labels "language" and "update code" in Amalgam. Moreover, there are some lucky cases where, even if the schema elements are normalized and annotated incorrectly, a correct lexical relationship is discovered in GeneX, for instance, between the incorrectly normalized and annotated schema elements "Schema1.array.image_an_params" and "Schema2.ARRAYMEASUREMENT.IMAGE_AN_PARAMS" a correct SYN lexical relationships is discovered. Consequently, some errors in the normalization method did not affect the performance of the lexical relationship discovery process.

5.4 The NORMS Tool

As described in the previous section, the output of the schema label normalization method has been successfully used as input of the lexical annotation phase of the MOMIS Data Integration system. However, the method is a powerful tool to automatic annotation in a real world scenario which improves the performance of schema matching applications based on the semantics of schemata. For this reason, we decided to implement the method in a stand-alone tool called NORMS (NORMalizer of Schemata). The main innovative features of NORMS are: (1) it implements the label normalization functionalities previously described (i.e. it allows to expand abbreviations and to enrich WordNet with CNs in an automatic way); (2) it provides a GUI that supports the designer during the normalization process and allows him/her to enhance the automatic results by correcting potential errors; (3) it provides, as additional feature, the possibility to automatically annotate the schema elements with respect to WordNet. Moreover, thanks to its "portable" output, it can be used to improve the performance of matching systems and several other applications that utilize the semantic and/or lexical information associated to schema labels such as data exchange, web interface integration, ontology alignment, data warehousing, web service integration etc.

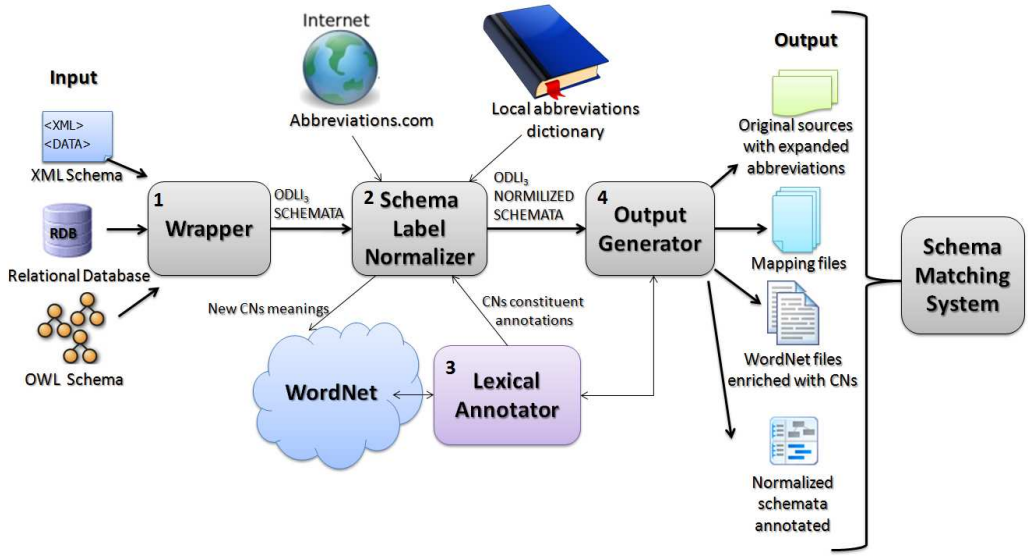

Figure 5.9: The NORMS architecture.

5.4.1 NORMS Overview

NORMS is composed of four main modules (see Figure 5.9): (1) Wrapper, (2) Schema Label Normalizer, (3) Lexical Annotator, (4) Output Generator. A modular architecture enables reuse and composition of single functionalities. In the following, we briefly describe the input and output of each module and the interaction among them.

Wrapper extracts the schemata to be normalized by the NORMS tool by using the wrappers implemented in the MOMIS system: each schema is logically converted into the internal object language ODL_{I^3}(see Section 3.1.1). Using the GUI the designer can upload one or more schemata from a number of sources: relational database (SQL Server, MySQL, ODBC and Oracle), XML and OWL.

Schema Label Normalizer processes schemata from the wrapper by performing the three main phases of the schema label normalization process: (1) Label Preprocessing, (2) Abbreviation Expansion and (3) Creation of New CN Meanings.

Lexical Annotator uses the CWSD (Combined Word Sense Disambiguation) algorithm [Bergamaschi et al., 2008] to annotate each label with corresponding WordNet synsets. As described before, this module is used to perform CN annotation but can be also used to automatically annotate the normalized schemata. This additional feature is important as some schema matching applications can take advantage from annotated schemata. The GUI allows the designer to correct the automatic annotations.

Output Generator exports all the results of NORMS in *portable* formats, so they can be directly used as input in schema matching systems or several other applications that utilize the semantic and/or lexical information associated to schema labels. Those applications include: data exchange, web interface integration, ontology alignment, data warehousing, web service integration etc. The module generates the following files:

1. *Original sources with expanded abbreviations*: a file for each source that contains a modified version of the original source where each abbreviation has been replaced with the previously selected expansion. The format of this file will be the same of the original source except for relational database sources. In this case the module is able to automatically generate the appropriate SQL scripts to modify the tables and the columns with their corresponding expanded abbreviations. For example, to expand the abbreviation "CO" in the label "DeliveryCO" contained in a SQL Server relational database, the module will generate the following script:

```
EXEC sp_rename
@objname = 'PODelivery.DeliveryCO',
@newname = 'DeliveryCompany',
@objtype = 'COLUMN'
```

2. *Mapping files*: a textual file for each source that contains the mapping between the old schema elements labeled with abbreviations and the new expanded labels where each label is univocally determined by its complete path within the source in "dot notation". For instance, for the label "DeliveryCO" the file will contain the following mapping: *PurchaseOrder.DeliveryCO = Purchase_Order.Delivery_Company*. These files are returned in order to provide the mapping with the original schemata because in some cases it is preferable to maintain the original schema and indirectly use the abbreviation expansion information.

3. *WordNet files enriched with CNs*: two WordNet files, *index.noun* and *data.noun*[14], which contain all the synsets and words of the noun syntactic categories in WordNet. These files are modified by the module by adding the new CN meanings created before (see Section 5.2.4). To use the enriched version of WordNet, thus, it is sufficient to substitute the original one in WordNet with these files.

[14] http://wordnet.princeton.edu/man/wndb.5WN.html

4. *Normalized annotated schemata*: a file containing the ODL_{I^3} normalized schema annotated by the Lexical Annotator module with respect to WordNet.

By using the GUI the designer can choose to export all these files or only some of them. The output has been conceived in order to satisfy the following requirements: to split the information in separate files in order to allow the designer to export and reuse only partial information; to provide the information in a easily to reuse way independently from the application where the output will be used.

We have implemented NORMS as an Adobe AIR[15] desktop application by using Adobe Flex 4 (for the GUI) and Java (for the business logic). We chose Flex technology because it makes creation of complex and well designed GUI an easy task and it can be easily deployed as a front end of Web application for normalization (our future work). Finally, Java and Flex applications are platform-independent.

Figure 5.10 shows a screenshot of the GUI of NORMS. It is divided in three main parts: an *action menu* (on the top) providing the list of actions for the normalization process, a *schema panel* with an intuitive tree representation of the wrapped schemata (on the left), and a set of *tabs*, one for each normalization phase (on the right). The whole normalization process requires a low level of human interaction, and the main operations are provided through these tabs that allow a designer to run and control the process and the results.

5.4.2 Performance and Human Effort Evaluation

As one of the major requirement for an automatic or semi-automatic tool is to reduce the amount of manual work, the initial evaluation of our normalization method presented in Section 5.3 has been extended by evaluating the amount of human effort that is possible to save by using NORMS. The evaluation starts from the default configuration where the designer does not perform any initialization task (i.e. no specific abbreviations manually inserted in LD and default abbreviation weights). In general, human effort may be divided in *pre-processing* (i.e. system parameter configuration) and *post-processing* effort (i.e. in our case, the correction of the automatic normalization results). Starting from the default configuration, we may assume the pre-processing effort equal to zero.

To approximate how much post-processing effort is possible to save by using NORMS, the *Accuracy* measure used in [Melnik et al., 2002] has been adopted to

[15] `http://www.adobe.com/products/air/`

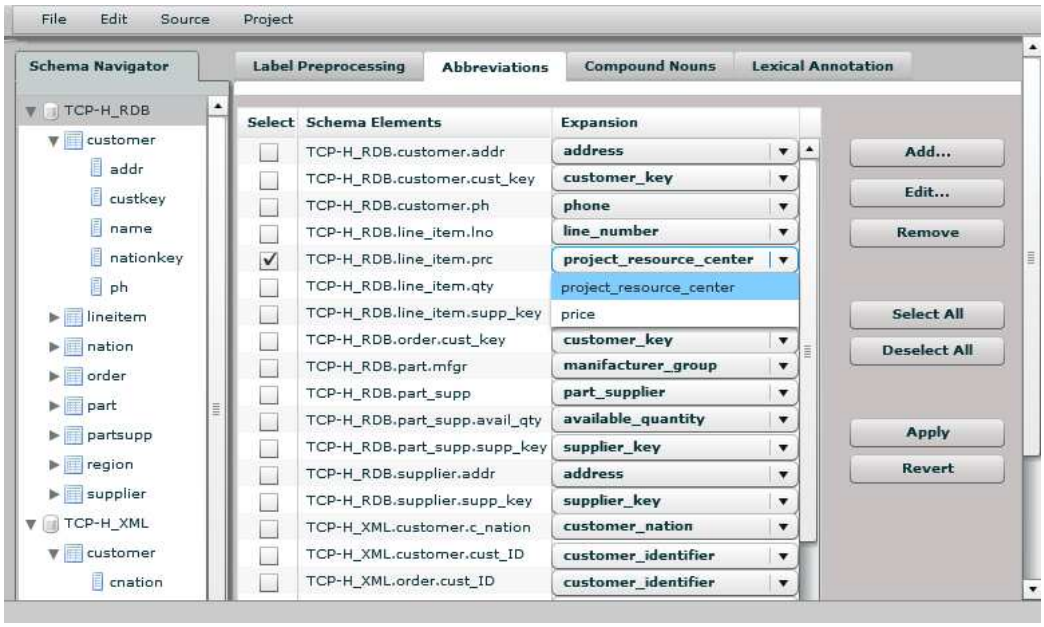

Figure 5.10: A NORMS screenshot.

estimate post-matching effort. It can be defined as

$$Accuracy = 1 - \frac{|a| + |c|}{|a| + |b|} = Recall * (2 - \frac{1}{Precision})$$

where, in our case, a is the number of false negatives (i.e. unnormalized labels requiring normalization), b is the number of true positives (i.e. correctly normalized labels) and c is the number of false positives (i.e. incorrectly normalized labels). In this way, we obtained an average Accuracy, over the five data sets, equal to 45%. However, according with [Melnik et al., 2002], Accuracy is very pessimistic if compared to F-measure (71%) and NORMS behave in general better than this measure predicts: first of all, Accuracy does not address semi-automatic normalization, in which the user interacts with the GUI by adjusting the single results step by step; moreover, it assumes equal effort to remove false positives and to add false negative normalizations, although in our case, the effort to remove or to correct a false positive normalization (e.g., the selection a wrong expansion for a given abbreviation) is considerably less than the effort to add a false negative (e.g., to manually add a new CN in WordNet).

Finally, we will demonstrate how NORMS improves the performance of schema matching, by running the MOMIS data integration system on both original and normalized schemata.

5.5 Related Work

Work related to the issues discussed in this paper are in the area of linguistic normalization, normalization for schema matching and, finally, the use of WordNet in schema matching.

5.5.1 Linguistic Normalization

In recent years, the problem of linguistic normalization has received much attention in different areas such as: machine translation, information extraction and information retrieval. However, the problems of abbreviation expansion and CN interpretation were originally conceived as fundamental tasks in the field of NLP.

In the NLP area, many works dealing with the abbreviation expansion task, assume that expansions often occur together in close proximity, i.e. in explicit position patterns, as for instance *"long form (short form)"* [Yeates et al., 2000, Chang et al., 2002]. However, these assumptions are not satisfied in structured and semi-structured data sources, and consequently, the corresponding approaches cannot be successfully applied in our context.

[Wong et al., 2006] proposes an ISSAC (Integrated Scoring for Spelling error correction, Abbreviation expansion and Case restoration) method that makes use of several resources (online abbreviation dictionaries, generic and domain specific corpora) to find correct forms of terms (including expansions). External text corpora may provide expansions for ad hoc abbreviations, while external dictionaries are generally suitable for expanding standard abbreviations as they provide content that has been verified (e.g., by dictionary editors). Similarly to ISSAC, our abbreviation expansion algorithm makes use of more resources to compute a list of candidate long forms. However, ISSAC does not assign different relevance to expansions coming from different sources. So far, we have not compared our algorithm to ISSAC as it is applied on textual sources and it is not directly applicable to structured and semi-structured data sources. Adoption and evaluation of this method in our context is part of our future work. For further details about the abbreviation expansion related work please refer to [Gawinecki, 2011].

As regards the task of CN interpretation, many works in the literature involve costly pre-tagged corpora and heavy manual intervention to collect training data. In [Moldovan et al., 2004] the authors extracted several CNs (3966 couples of noun-noun CNs) from a corpus and manually annotated them. 80% of these CNs were used as training data to automatically annotate the remaining 20% of the CNs. In [Su Nam Kim, 2008], in order to collect training data, 2169 pairs of noun-noun CNs were extracted from the Wall Street Journal and manually annotated by using a set of 20 semantic relationships. Half of this set was used as training data to automatically annotate the remaining 50% of the CNs. Our

method required to manually annotate a smaller number of pairs of noun-noun CNs (625 pairs of noun-noun unique beginners, see Section 5.2.4).

There are three other main problems with corpus-based methods: (1) they provide some underlying assumption in terms of domain or range of interpretations; this leads to problems in scalability and portability to novel domains; (2) there is a trade-off between how much training data (pre-tagged corpora) is used and the performance of the method; (3) human intervention is required not only to manually annotated the right relationship for CNs, but also to select and filter the training data from large corpora.

Following [Fan et al., 2003], we claim that the cost of acquiring knowledge from manually tagged corpora for different domains may overshadow the benefit of interpreting the CNs. Our CN interpretation method is domain-independent as it does not require to prepare training data on the basis of the domain under consideration.

5.5.2 Normalization Techniques for Schema Matching

As previously observed, the presence of non-dictionary words in schema element labels (including CNs and abbreviations) may affect the quality of *schema element matching* and requires additional techniques to be dealt with [Do, 2006].

Surprisingly, current schema matching systems either do not consider the problem of abbreviation expansion at all or solve it in a non-scalable way by including a *user-defined abbreviation dictionary* or by using only simple *string comparison techniques*.

For instance, both the well known CUPID [Madhavan et al., 2001] and COMA [Aumueller et al., 2005] schema matching systems overcome the problem of abbreviations by relying on the availability of a complete user-defined dictionary or a tool for abbreviation expansion.

Dealing with short forms by using a user-defined dictionary lends to problems in terms of scalability: (a) the dictionary cannot handle ad hoc abbreviations; (b) same abbreviations can have different expansions depending on the domain, which means that an intervention of a schema/domain expert is still required; and (c) the dictionary evolves over time and it is necessary to maintain the table of abbreviations.

Some works have tried to address the limitations of the user-defined dictionary approach. For instance, the Similarity Flooding [Melnik et al., 2002] algorithm can detect matches between elements labeled with simple ad hoc abbreviations and the corresponding long forms. They do not expand abbreviations but use simple string comparison techniques; more precisely, they compare common prefixes and suffixes of literals and are thus able to detect a match between, for instance, elements such as “Dep” and “Department”. However, syntactical methods are not

able to bring to the surface the semantics of abbreviations. In contrast with our method, they cannot detect a match between synonyms like "QTY" (short form of "quantity") and "amount".

The problem of ad hoc abbreviations has been further addressed by Ratinov and Gudes [Ratinov and Gudes, 2004] by employing an external text corpus as the source for potential abbreviation expansions. The authors focus on the extraction of possible expansions for a given abbreviation but they do not provide any support to select the most relevant one. Text corpora can be relevant sources to expand ad hoc abbreviations, but they suffer from some limitations: they do not provide explicit information useful for selecting a relevant expansion for an ambiguous abbreviation (i.e. abbreviation that can have more than one possible expansion). Therefore, in some cases the list of the suggested expansion candidates for a given short form is very long (several hundreds in the discussed approach) and it is not ranked. On the contrary, our method is able to assign a weight to each candidate long form and to automatically select the top-scoring one.

The problem of ambiguous abbreviations occurring in the user-defined dictionary has been addressed in [Chai et al., 2008] with predefined domain-dependent transformation rules, e.g., "SSN" → "Social Security Number" (for the schema that belongs to accounting domain) and "SSN" → "System Study Number" (the military domain). Again, in contrast with respect to our method, the need to manually define a priori rules requires intensive manual effort and thus limits the scalability as well as the user-defined dictionary.

Similarly to the abbreviation expansion problem, few papers address the problem of CN annotation in schema matching area. In [Su and Gulla, 2004] a preliminary CNs comparison for ontology mapping is proposed. This approach suffers from two main problems: first, it starts from the assumption that the ontology entities are accompanied by comments that contain words expressing the relationship between the constituents of a CN; second, it is based on a set of manually created rules. Xu and Embley in [Embley et al., 2001] perform attribute matching by using WordNet as an external resource. They recognize the problem of the presence of non-dictionary words among attribute labels, but in contrast with our method, abbreviation expansion and CN annotation are manually executed.

The S-Match [Giunchiglia et al., 2005, Shvaiko et al., 2010] and Ctx-Match [Bouquet et al., 2003] algorithms discover semantic matching by analyzing the meaning codified in the entities and the structures of ontologies and by using WordNet as an external semantic source. CNs that are not present in WordNet are split into single words and their meaning is represented as the intersection of the single constituent meanings. In contrast with our method, they neither make distinctions between head and modifier nor enrich WordNet with the new CNs. H-Match [Castano et al., 2006], similarly to MOMIS, creates a Common Thesaurus of semantic relationships among the schema elements.

This tool deal with the problem of CNs by inserting, in the common thesaurus, a hypernym/hyponym relationship between the CN and its head, and a generic RT relationship between the CN and its constituents. However, in contrast with our approach, H-match does not perform constituent disambiguation, constituent redundant identification, CN interpretation and does not enrich WordNet with new the CNs.

Other schema and ontology matching tools neither interpret nor normalize CNs but they treat the constituents of a CN in isolation [Su and Gulla, 2004, Li, 2004, Le et al., 2004, Euzenat et al., 2004]. This oversimplification leads to the discovery of false positive relationships, thus negatively affecting the matching results (as shown in the example in Section 5.1).

Another way to overcome the problem of limited amount of useful schema information and meaningless schema labels (as in the case of presence of several non-dictionary words) is the use of instance-level schema matching techniques [Rahm and Bernstein, 2001], which exploit the information associated with data instances. For example, the value of instances can be used to select the right expansion for an ambiguous abbreviation (for instance, if the column name is "tel" and the possible expansions are "telephone" or "Technology Enhanced Learning" we can decide on what expansion must be chosen based on the instances values). However, instance analysis is computationally a heavy task since it involves a great number of elements. A metadata structure derived from an analysis of the attribute extension could be of great help in overcoming such limitation. In [Bergamaschi et al., 2007b, Bergamaschi et al., 2007e, Orsini, 2009], a technique for providing metadata related to attribute values is described. Such metadata represent a synthesized and meaningful information emerging from the data. These metadata are called *relevant values*, as they provide the users with a synthetic description of the values of the attribute which refer to by representing with a reduced number of values its domain.

Chapter 6

Uncertainty in Lexical Annotation

In this chapter, we present an automatic method aimed at discovering probabilistic lexical relationships in the environment of data integration "on the fly". The method is based on a probabilistic lexical annotation technique, which automatically associates one or more meanings with schema elements with respect to WordNet. This method has been implemented in a tool called ALA (Automatic Lexical Annotator) integrated within the MOMIS system. However, as described in the following, it may be applied in general in the context of schema mapping discovery, ontology merging, data integration systems, and web interface integration.

This work was partially funded by the "Searching for a needle in mountains of data!" project funded by the Fondazione Cassa di Risparmio di Modena within the Bando di Ricerca Internazionale 2008[1] and it was conducted in the context of the NeP4B (Networked Peers for Business) project[2]. The work presented in this chapter has been published in [Po et al., 2009, Po and Sorrentino, 2011, Bergamaschi et al., 2009a, Bergamaschi et al., 2009b, Bergamaschi et al., 2011b] and it has been realized in collaboration with the Ph.D. and fellow research Laura Po.

The rest of this chapter is organized as follows: Section 6.1 gives a definition of the problem addressed in this paper. In Section 6.2, the architecture of the probabilistic relationship discovery method is described. Subsequently, Section 6.3.1 describes the new contributions with respect to the label normalization method presented in Chapter 5; Section 6.4 and Section 6.5 describe, respectively, the PWSD (Probabilistic Word Sense Disambiguation) algorithm and the generation of probabilistic relationships. Section 6.6 sketches out the evaluation of PWSD and the relationship discovery process in a real scenario, comparing the results with other WSD approaches. Finally, in Section 6.8, related work is discussed.

[1] http://www.dbgroup.unimo.it/keymantic

[2] http://www.dbgroup.unimo.it/nep4b

6.1 Problem Definition

Flexible systems capable of identifying mappings in an automatic and dynamic way are increasingly in demand: schemata exhibit a significant evolution over time, due to changing market conditions and evolving user sophistication and needs. The manual identification of such mappings is time-consuming and tedious, and clearly impossible in the context of dynamic schema matching. This chapter is focused on schema matching in the context of dynamic data integration [Bergamaschi et al., 1999]. Dynamic data integration systems are systems where semantic mappings among schemata of different sources have to be identified *on the fly* with minimal human intervention or with no intervention at all (i.e. in a semi-automatic or automatic way).

However, in performing automatic schema matching, several problems arise: (1) there is no uniform conceptualization of the schemata and the semantics of the information in the various sources involved; (2) schema elements can be ambiguous in their semantics and recognizing the meaning of schema elements can be difficult for designers themselves; (3) automatic schema matching is intrinsically uncertain and the more the information to be matched, the more difficult it becomes to determine an exact match [Dalvi and Suciu, 2007]. For example, applications such as Google Base, the large number of sources present in the "deep web" and the tools used for processing biological data all require flexibility and the handling of uncertainty [Louie et al., 2007].

In this chapter, we propose a method for the automatic discovery of probabilistic lexical relationships, which represents the first step for a fully *dynamic* data integration system. Our method focuses on the extraction of the underlying lexical knowledge from the schemata.

Schema elements of a data source are automatically annotated according to the lexical reference database WordNet [Miller et al., 1990] by using WSD techniques.

In our method, lexical annotation is performed by PWSD, an automatic algorithm that combines several WSD algorithms by using the Dempster-Shafer's rule of combination [Shafer, 1976]. For each schema element, the PWSD combines the output of several WSD algorithms and produces a probabilistic distribution on meanings. The main advantage of PWSD is its flexibility: it is possible to add or remove algorithms very easily; what is required is just that the output of the new WSD algorithm is a probabilistic distribution on meanings. By using the Dempster-Shafer's rule of combination, PWSD is able to model the uncertainty of the WSD algorithms (i.e their ignorance).

The probabilistic annotations, generated through PWSD, are used to derive probabilistic lexical relationships among local sources that are subsequently collected in the PCT (Probabilistic Common Thesaurus). The probabilistic relation-

ships represent the basic information that can be used to derive probabilistic mappings. In contrast with other approaches, we do not choose the "best" discovered relationships only but we compute all possible relationships among schemata and assign a probability value to each of them.

We formally define the semantics of our method for automatic generation of probabilistic relationships.

As previously described (see Section 2.2), lexical annotation is defined as the connection of a schema element with its meanings defined in a lexical resource. However from now on, the chapter will also make reference to the annotation of the label of a schema element[3].

Figure 6.1 shows an example of two schemata to be integrated. Let us consider, for instance, the element "address" contained in the schema (b). In WordNet the noun "address" has eight different meanings, including very similar ones such as "written directions for finding some location; written on letters or packages that are to be delivered to that location" or "a sign in front of a house or business carrying the conventional form by which its location is described". In the literature, many WSD approaches that generate a single (forced) annotation for each word can be found. However, in such cases choosing a given individual annotation would be difficult even for a human annotator: generating probabilistic annotations may be more useful and possibly the only solution that avoids losing semantic information. The concept of lexical annotation is enriched by adding the notion of uncertainty.

Uncertainty is an intrinsic feature of automatic and semi-automatic processes and provides a quantitative indication of the quality of the result.

In our method, uncertainty is qualified as probabilities, where the probabilities are values in the interval [0-1].

Definition 28 (Probabilistic Annotation) *Let T be a schema and t be an element (class or attribute) $\in T$. We define $S_t = \{s_{\#1}, .., s_{\#n}\}$ as the set of all possible meanings of t with respect to a lexical resource (such as WordNet). The probabilistic annotation of t is the triple $< T, t, A_t >$, where $A_t = \{a_1, ..., a_k\}$ is the set of annotations associated with t. In particular, a_i is defined as the couple $(s_{\#i}, P(s_{\#i}))$, where $s_{\#i} \in S_t$ is a selected meaning for the element t and $P(s_{\#i})$ is the probability value, in the interval [0-1], assigned to this annotation (this probability indicates how well the meaning $s_{\#i}$ represents the element t).*

[3]Manual annotation is independent of the label, as the designer can select the meanings to be associated with a schema element of other words (e.g., the designer can annotate the schema element "home" with a meanings of the word "house"). Automatic annotation, instead, is strongly linked to the label. This is because the automatic annotation will directly process the label of a schema element (or its normalization, if needed).

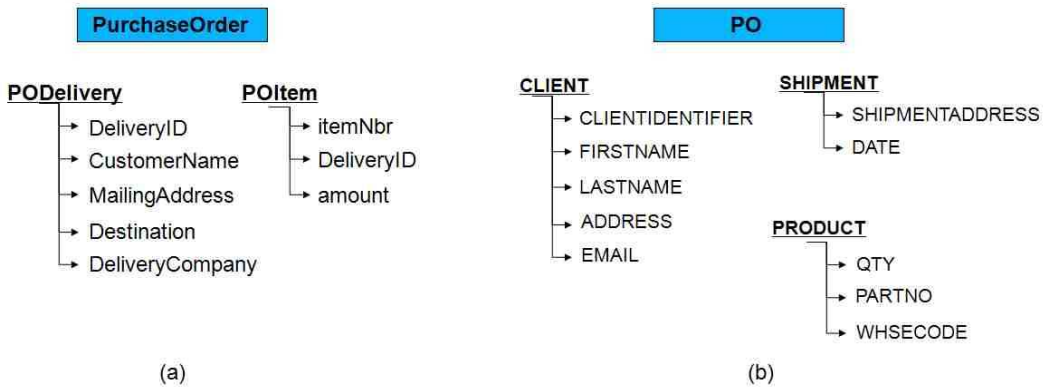

Figure 6.1: Graph representation of two schemata.

The schemata shown in Figure 6.1 contain many elements labeled with non-dictionary CNs (e.g., "CustomerName") and abbreviations (e.g., "PO" and "QTY").

As described in Chapter 3, starting from the lexical annotation of schema labels it is possible to derive lexical relationships among elements on the basis of the semantic relationships defined in WordNet among their meanings.

In our method, the uncertainty introduced during lexical annotation is propagated to the lexical relationship discovery process: for each relationship a value representing the probability of that relationship is computed.

Definition 29 (Probabilistic Lexical Relationship) *Let T_1 and T_2 be two heterogeneous schemata, and the elements $t_i \in T_1$, $t_j \in T_2$. A probabilistic lexical relationship is defined as the couple $(< t_i, t_j, R >, P)$ where $< t_i, t_j, R >$ is a lexical relationship between t_i and t_j of the type R, and P is the probability value, in the interval [0-1], associated to this relationship.*

Probabilistic lexical relationships are inter-schema relationships collected in the PCT. Moreover, the PCT contains the intra-schema relationships called structural relationships (see Section 3.1.1).

As structural relationships derive directly from the structure of schemata, they are not affected by uncertainty. In order to insert them in the PCT, we define each structural relationship as an "ordinary relationship" that is described by a probability value equal to 1.

Probabilistic lexical annotation helps to improve the schema matching accuracy by handling and modeling the uncertainty during the lexical relationship discovery process. Let us reconsider the previous example about the annotation of the schema element "ADDRESS". If we annotate this element without performing

probabilistic annotation, we can choose the WordNet meaning "the place where a person or organization can be found or communicated with", and we discover a BT lexical relationships with the term "MailingAddress" in the schema (a), or we can choose the meaning "written directions for finding some location; written on letters or packages that are to be delivered to that location" and we discover a SYN lexical relationship with the term "destination" in the schema (a). In both cases, by selecting one meaning only we may miss the right relationships between this element and the other schema elements, thus obtaining false negative relationships.

By using our method, we do not exclude a priori any of these lexical relationships, but we handle their uncertainty by associating to each of them a probability value depending on the previous probabilistic annotations.

6.2 Architecture

This section describes the overall architecture of our probabilistic relationship discovery method. I introduce our specific contributions in the context of this architecture, and describe the individual phases of the probabilistic relationship discovery method.

As shown in Figure 6.2, the process can be seen to comprise three main phases:

Source Schema Extraction (Wrapping) enables our method to manage structured and semi-structured data sources. The method exploit the MOMIS specialized software (wrappers) for logically converting the format of the data source schema into the internal object language ODL_{I^3} (Figure 6.2-a, see Section 3.1.2). The output of source schema extraction is the set of schemata to be integrated into the ODL_{I^3} format.

Lexical Knowledge Extraction represents the core of our method. The traditional MOMIS lexical knowledge extraction phase is extended by introducing the notion of uncertainty. In particular, this phase includes two steps (Figure 6.2-B): modificare la figura facendo vedere che la normalizzazione arriva da NORMS.

1. *Schema Label Normalization*: this step is performed by the NORMS tools, which receives as the schemata to be integrated. The approach presented in Chapter 5 is extended by introducing the notion of uncertainty during the disambiguation of CNs. The output of the schema normalizer are the normalized schemata (i.e. the schemata where the abbreviations have been expanded and the CNs have been annotated). For more detail about schema label normalization see Chapter 5.

2. *Probabilistic Word Sense Disambiguation*: during this phase the PWSD algorithm performs automatic probabilistic lexical annotation of schema ele-

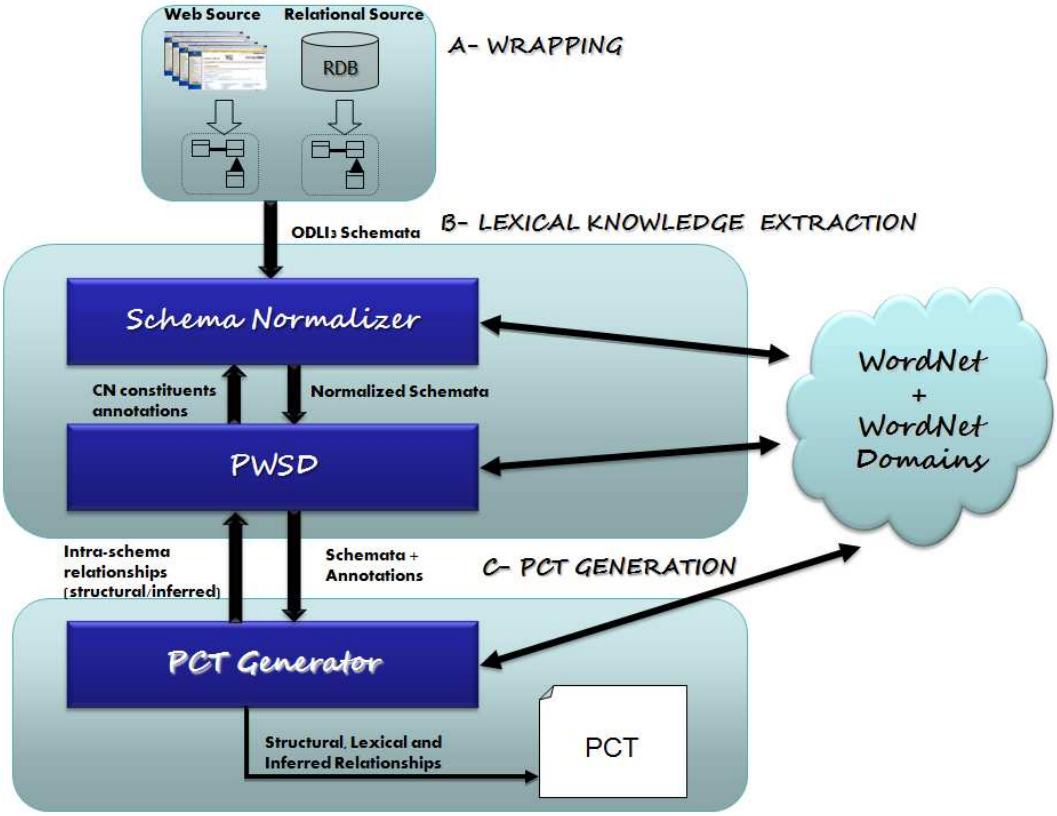

Figure 6.2: PWSD overview.

ments, by combining a set of WSD algorithms. PWSD associates a probability value to each annotation; this value shows the uncertainty of the annotation process. PWSD receives the normalized schemata as input, and provides the annotated schemata as output. In the annotated schemata, each schema element is associated with one or more probabilistic annotations. PWSD performs lexical annotation with respect to WordNet.

PCT Generation extends the lexical relationship discovery process of the MOMIS system [Beneventano et al., 2003b] by including the treatment of uncertainty. PCT collects a set of intra and inter-schema relationships (Figure 6.2-C). Starting from the annotated schemata, probabilistic lexical relationships are derived among schema elements on the basis of WordNet semantic relationships. Probabilistic lexical relationships are collected in the PCT, together with the ordinary structural relationships. Moreover, the PCT contains inferred probabilistic relationships detected by means of subsumption computation, performed by using the description logics techniques of ODBTools [Bergamaschi et al., 1997].

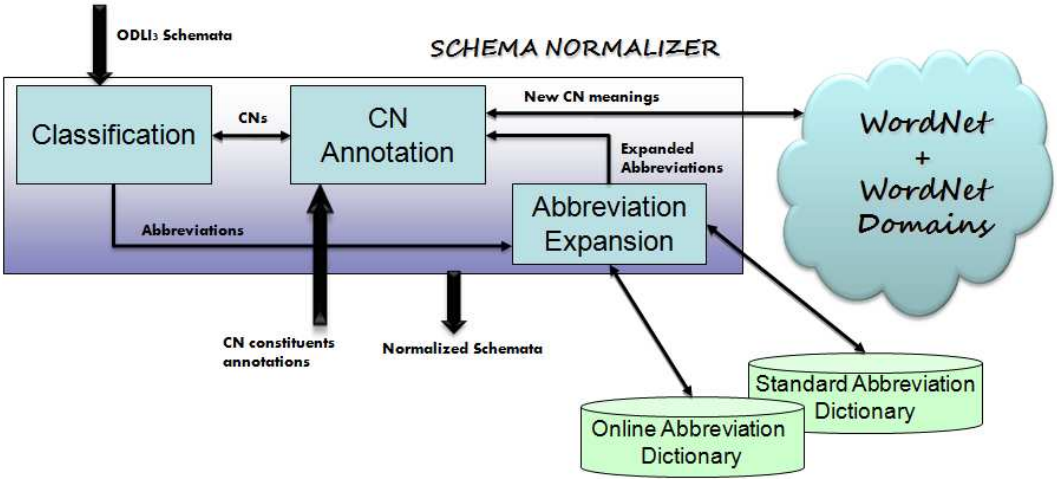

Figure 6.3: PWSD and schema label normalization interaction.

6.3 Schema Label Normalization

In this section, we describe how the schema normalization approach, presented in Chapter 5, has been modified in order to be integrated in our method. At this end, we modified only the CN annotation phase in order to introduce the notion of uncertainty during the constituent disambiguation step (see Section 5.2.4).

As regards the abbreviation expansion phase (see Section 5.2.3), it has not be modified and the method still forces the selection of the best possible long form, as our method does not support probabilistic label selection for an element. The selected expansion is subsequently annotated by using PWSD. Allowing probabilistic label selection will require the handling of uncertainty already in this phase, and up until the start of the relationship discovery process. In this way, we will significantly increase the complexity of our method: more probabilistic labels will be associated to an element, and more probabilistic annotations will be associated to each of these probabilistic labels.

6.3.1 Probabilistic CN Annotation

The CN annotation method proposed in Section 5.2.4, has been modified in order to be integrated with PWSD.

As seen in Chapter 5, the CN annotation algorithm includes four main steps: (1) disambiguation of CN constituents, (2) identification of redundant constituents, (3) CN interpretation via semantic relationships and (4) creation of new WordNet meanings for a CN.

The phase that has been modified is the CN constituent disambiguation phase.

In Section 5.2.4, the CN annotation algorithm does not produce probabilistic annotations as it uses a non-probabilistic WSD algorithm. In this method, the algorithm produces a set of probabilistic annotations, as the CN constituents are annotated by using PWSD.

During the phase of disambiguation of the head and the modifier, PWSD returns a set of probabilistic annotations for each CN constituent. For example, for the term "Delivery Company", the annotations returned by PWSD are "$(Company_{\#1}, 0.50)$", "$(Company_{\#3}, 0.12)$" for "Company" and "$(Delivery_{\#1}, 0.34)$" for "Delivery". In this case, to obtain all possible meanings for "Delivery Company", we combine all the annotations for the term "Company" with all the annotations for the term "Delivery". We obtain the following combined annotations: "$(Company_{\#1}Delivery_{\#1}, 0.17)$", and "$(Company_{\#3}Delivery_{\#1}, 0.04)$" where the probability values of the combined annotation is the product of the probability value of the individual annotations. To compute the probability of new CN meanings, the method assumes that the single probabilities are independent. This assumption does not usually hold. However, factoring out dependencies in WSD context is extremely difficult, as they are usually hidden[4] [Preiss, 2004].

When there is a high number of returned annotations, the size of the possible combined annotation set may be quite large, and incorrect CNs can be generated. To ensure that clearly incorrect CN annotations are not generated, a threshold is applied. In this case, by excluding the combined annotations whose probability is below the threshold of 0.1, only the annotation "$(Company_{\#1}Delivery_{\#1}, 0.17)$" are returned.

For the next steps, i.e. redundant constituent identification, CN interpretation via semantic relationships and the creation of a new WordNet meaning for a CN, please refer to Chapter 5.

6.4 Probabilistic Lexical Annotation

Lexical Annotation is an enabling technology that can give an important contribution to the identification of relationships among schema elements in data integration and ontology matching scenarios. However, to prove effective in the context of dynamic data integration, lexical annotation has to be computed by automatic techniques (see Section 2.4.

[4] In supervised or semi-supervised WSD algorithms, the conditional probability is estimated from training data, using relative frequencies [Navigli, 2009]. However, PWSD is an unsupervised WSD algorithm. Therefore, it does not use any training data nor a pre-annotated corpus. It is not able to make the dependence between the annotations of the two labels explicit.

Ensemble WSD methods, to perform lexical annotation, are becoming increasingly popular as they overcome the weaknesses of individual approaches. Different combination strategies can be applied, such as majority voting, probability mixture, rank-based combination, maximum entropy combination or probabilistic combination. In [Resnik and Yarowsky, 2000], Resnik and Yarowsky suggest that a measure based on cross-entropy or perplexity would allow for the case where a number of very fine-grained meanings are essentially correct. These measures are based on a WSD algorithm producing a probability distribution on meanings. This is the main reason for our choice of a probabilistic WSD algorithm, as it can assign a high probability to all close meanings rather than choosing one through a forced-choice method.

However, the output of the algorithm cannot only be used directly by applications that deal with probability, it can also be converted into an individual meaning assignment (choosing the best annotation, i.e. the annotation with the highest probability value computed) and used by applications that deal with individual annotations.

PWSD needs to satisfy the following requirements:

- The algorithm needs to produce a probability distribution (so, for example, majority voting is not suitable).
- The algorithm needs to cope with probability distributions over meanings and distributions over subsets of meanings (such as those produced by a WSD algorithm which assigns probabilities to a set of meanings and not to individual meanings).
- The algorithm must be able to deal with ignorance, which means modeling not only what a WSD algorithm knows but also what it does not know (i.e its uncertainty or ignorance).

These conditions are satisfied by both the Dempster-Shafer theory [Shafer, 1976] and Bayes' theorem. However, several works [Comber et al., 2004, Hoffman and Murphy, 1993] have pointed out that while Bayes' theorem is most suited to problems where there are probabilities for all events, it is least suited to problems where there is partial or complete ignorance, and limited or conflicting information. The Dempster-Shafer theory, on the other hand, can model various types of partial ignorance and limited or conflicting evidence: it is a more flexible model than Bayes' theorem.

This ability to explicitly model the degree of ignorance makes the theory very appealing. It has been applied in NLP [Le et al., 2006] to combine several WSD algorithms. In [Preiss, 2006] a comparison is made between performances of the WSD system which combine information sources by employing Dempster-Shafer

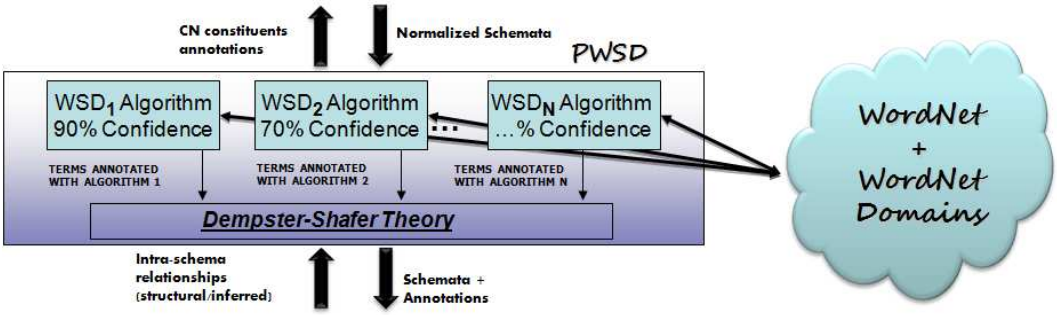

Figure 6.4: Lexical annotation performed by PWSD

theory and performances where sources are combined through Bayes theorem and weighted interpolation. On a text corpus (i.e. the Senseval-2 [5]) both the Dempster-Shafer and the Bayes' theorem combination methods outperform the linear interpolation combination. However, even though the Bayes' theorem combination method eliminates a bias towards the most frequent meaning, it does not lead to a great improvement over the Dempster-Shafer combination method.

Based on these motivations, the Dempster-Shafer combination method has been selected.

6.4.1 WSD Algorithms

The main feature of PWSD is its flexibility, as it is composed of multiple probabilistic algorithms: such modularity is made possible by the application of Dempster-Shafer theory.

As it is needed that all of our algorithms produce a probability distribution (in order to be combined by PWSD), the five WSD algorithms have been slightly modified developed and tested in [Bergamaschi et al., 2007c, Beneventano et al., 2008b].

- The *Structural Disambiguation Algorithm* tries to disambiguate source terms by exploiting the structural relationships extracted from the sources (see Section 4.2.1).
- The *WordNet Domain Disambiguation Algorithm* tries to disambiguate terms by exploiting domain information supplied by WND (see Section 4.2.2).
- The *WordNet First Sense Heuristic* selects the first WordNet meaning (that is, the most frequent sense) for a given term. It has been observed that it is

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

quite difficult for a WSD algorithm to beat the WordNet first sense heuristic. The WordNet first sense heuristic provides a good default value for words which do not obviously have another meaning (from another module), and thus WordNet first sense heuristic forms a part of our PWSD.

- The *Gloss Similarity Algorithm* is based on mining the glosses (i.e. textual definitions) given for terms in an online dictionary (WordNet in our case). The method is inspired by Lesk's method [Lesk, 1986] and is based on maximizing the similarity of the meanings assigned to schema elements. The rationale of this method is that the glosses of the possible meanings for the elements in the "vicinity" of a given element *t* should contain more words related to a particular gloss of a meaning of *t*. For example, when disambiguating the element "Bank" in "DB1.Account"("Bank", "Branch", "Number", "IBANcode"), we can expect the glosses of "Account", "Branch", "Number" and "IBANcode" to collectively contain more terms related to the meaning of Bank as a financial institution than to its meaning as a hydraulic engineering artifact. The "vicinity" of an element *t* has been detected to be the set of all the elements of the data source where *t* is contained (or a subset of it, if it contains more then 200 elements).

- The *Iterative Gloss Similarity Algorithm* [Beneventano et al., 2008b] is an iterative relaxation labeling technique [Hummel and Zucker, 1983] based on the Markov Random Field theory. It consists in an iterative algorithm: the synsets of the elements are initially attributed by means of the Gloss Similarity Algorithm, and they are then refined iteratively by assuming that the meanings attributed to the other elements at the previous iteration are correct.

At present, each WSD algorithm produces a set of probabilistic annotations for the target element, which are computed according to its confidence (in our case, the confidence of each algorithm is selected as the precision of the algorithm evaluated on a benchmark). As a matter of fact, not every algorithm finds a meaning to assign to each term. In addition, each algorithm may be more appropriate to certain situations, so its behavior is not 100% trustworthy. To model the uncertainty of the WSD algorithms, PWSD uses the confidence of the algorithm and its ignorance (that is, the complementary value of the confidence).

Let us consider the term "name". In WordNet, we find six different meanings for "name" ($name_{\#1}$, $name_{\#2}$, .., $name_{\#6}$). Suppose we have to combine three algorithms that give different outputs (as shown in Figure 6.5): WSD1 chooses a set of meanings formed by $name_{\#1}$, $name_{\#2}$; WSD2 provides $name_{\#1}$ as the correct meaning; and WSD3 does not give any result. Since WSD1 has a 70%

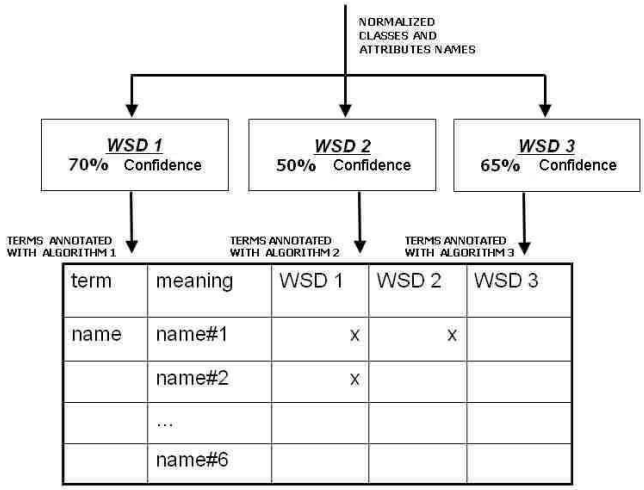

Figure 6.5: An example of the application of a set of WSD algorithms

of confidence this means that the algorithm has a 30% of ignorance. The probabilistic distribution on meanings for WSD1 is a 70% assignment to ($name_{\#1}$, $name_{\#2}$), while for WSD2 is a 50% assignment to ($name_{\#1}$). What it is necessary to obtain is a rate of confidence to be assigned to each possible meaning of the term under consideration.

6.4.2 The Dempster-Shafer Theory

The Dempster-Shafer theory of evidence [Shafer, 1976, Parsons and Hunter, 1998] provides a mechanism for modeling and reasoning on uncertain information in a numerical way.

The theory tacitly assumes that the probabilities being combined are independent, an assumption which does not usually hold. However, factoring out dependencies in general is extremely hard, as they are usually hidden [Preiss, 2006]. In [Altinay, 2006], the role of independence for classifier combination in Dempster-Shafer evidence theory has been studied. The paper has demonstrated that the independence of the classifiers should not be considered necessary in multiple classifier combinations using probabilistic evidence representation and Dempster's rule of combination.

As WSD can be viewed as a classification task (where the meanings are the classes, and an automatic classification method is used to assign each occurrence of a term to one or more classes), the method can safely adopt the independence assumption on the set of WSD algorithms.

The theory deals with the so-called *frame of discernment*, the set of base ele-

ments θ (in this, the set of all possible meanings for the term under consideration), and its power set 2^θ, which is the set of all subsets of the interesting elements (all the possible subsets of the set of possible meanings).

The foundation of the Dempster-Shafer theory is a *probability mass function m(·)* that assigns zero mass to an empty set and a value [0,1] to each element of 2^θ, the total mass distributed being 1 so that:

$$\sum_{A \subseteq \theta} m(A) = 1$$

The belief ca be defined as a subset A of the set of all propositions as the sum of all the probability masses that support its constituents:

$$Bel(A) = \sum_{B \subseteq A} m(B)$$

Each of our WSD algorithms will assign a belief mass to a meaning or a set of meanings for every term. The method derives the belief mass function from the output and the confidence of the WSD algorithms. Combining the outputs of several WSD algorithms means combining more probability assignments, then it is necessary to use the Dempster-Shafer's rule of combination:

$$m(a) = K \sum_{\bigcap A_i = a} \prod_{1 \leq i \leq n} m_i(A_i)$$

$$K^{-1} = 1 - \sum_{\bigcap A_i = \emptyset} \prod_{1 \leq i \leq n} m_i(A_i) = \sum_{\bigcap A_i \neq \emptyset} \prod_{1 \leq i \leq n} m_i(A_i)$$

where n is the number of the WSD algorithms that supplied a disambiguation output for the term under analysis.

To obtain the probability assigned to independent meaning, the method eventually needs to smooth the belief mass function concerning a set of meanings.

$$P(syn) = \sum_{syn \in A} \frac{m(A)}{\|A\|}$$

The following provides an example of how the PWSD algorithm is applied. As shown in Figure 6.5, source elements are automatically annotated by the application of a set of WSD algorithms. PWSD combines the outputs without considering the algorithms that do not supply any annotation for the element. So, in this case, the evaluation will only be executed on the outputs of WSD1 and WSD2. If a WSD algorithm has a 70% confidence this means that the algorithm has a 30% ignorance.

mass function	WSD 1	WSD 2	Dempster combination
m{name#1}		0.5	0.5
m{name#1,name#2}	0.7		0.35
m{name#2}			0
…			
m{name#1,name#2, …name#6} = m{ignorance}	0.3	0.5	0.15

Figure 6.6: An application of the Dempster-Shafer theory.

probability function	PWSD
P{name#1}	0.67
P{name#2}	0.17
P{ignorance}	0.15

Figure 6.7: Generation of probabilistic annotations.

The application of the Dempster-Shafer's rule of combination is shown in Figure 6.6. As WSD1 supplies a set composed of two meanings, the probability will be assigned to this set.

The results obtained after the application of the Dempster-Shafer's rule of combination show the probabilities assigned to different sets of meanings. In order to use this result for computing lexical relationships, the method has to go back to the case of probabilities assigned to individual meanings. As shown in Figure 6.7, the probability assigned to the set of meanings $\{name_{\#1}$,$name_{\#2}\}$ will be split in the single probabilities assigned to $name_{\#1}$ and $name_{\#2}$.

6.5 PCT Generation

Starting from the annotations for schema elements, the method uses these probability distributions over the set of possible meanings to infer probability distributions for lexical relationships among elements. The annotation output of PWSD is used to compute the lexical relationships that will be included in the PCT. From the semantic relationships defined in WordNet between meanings, it is possible to derive the lexical relationship defined in MOMIS as described in Section 3.1.1.

However, in contrast with the traditional MOMIS approach described in Chapter 5, PWSD associates a set of probabilistic meanings to a term in a source. So,

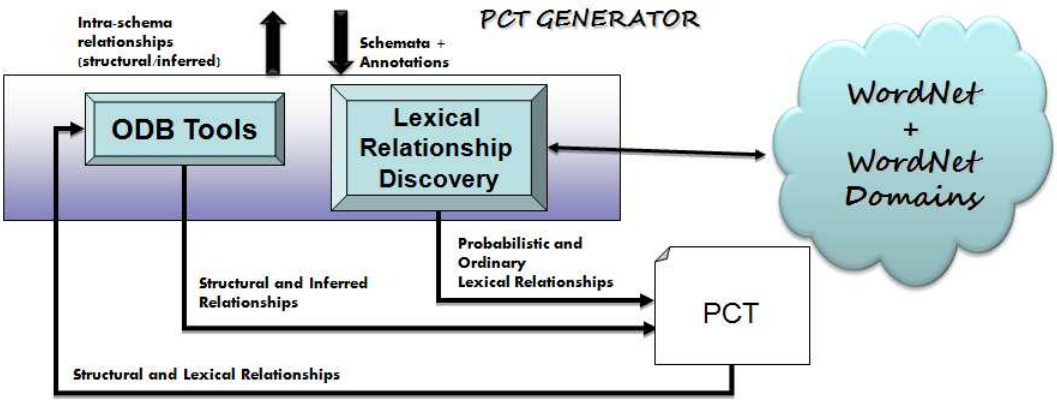

Figure 6.8: The PCT generation.

a term *t* is described by the meaning $t_{\#i}$ with a certain probability. Since all the provided meanings are included in the lexical resource WordNet, each of them is located within a network of lexical relationships.

A probabilistic lexical relationship exists between two elements, if there exists a lexical relationship between their meanings in WordNet. The probability assigned to lexical relationships depends on the probability value of the meaning under consideration for an element. Thanks to the formula of the *joint probability*, the probability value associated to an relationship holding among $t_{\#i}$ and $s_{\#j}$ can be defined as:

$$P(Rel(t_i, s_j)) = P(t_i) * P(s_j)$$

Probabilistic lexical ODL$_{I^3}$ relationships are collected in the PCT, as well as the ordinary structural relationships extracted from schemata by ODBTools (see Section 3.1.1). Eventually, to minimize the introduction of errors, probabilistic relationships with a probability value under a certain threshold can be filtered.

6.6 Experimental Evaluation

The prototype that implements our method has been developed within the MOMIS system in order to test our approach on real-world sources.

Our evaluation goals are: (1) to evaluate the effectiveness of automatic lexical annotation, (2) to verify whether, by handling uncertainty during the lexical annotation phase, our method improves the lexical relationship discovery process, (3) to evaluate the performance and the computational complexity of our method.

	Amalgam	OAEI
Number of Elements	146	462
Non-dictionary words	94	137
Abbreviations	86	23
CNs	62	132
Endocentric CNs	49	74

Table 6.1: Characteristics of Amalgam and OAEI data sets.

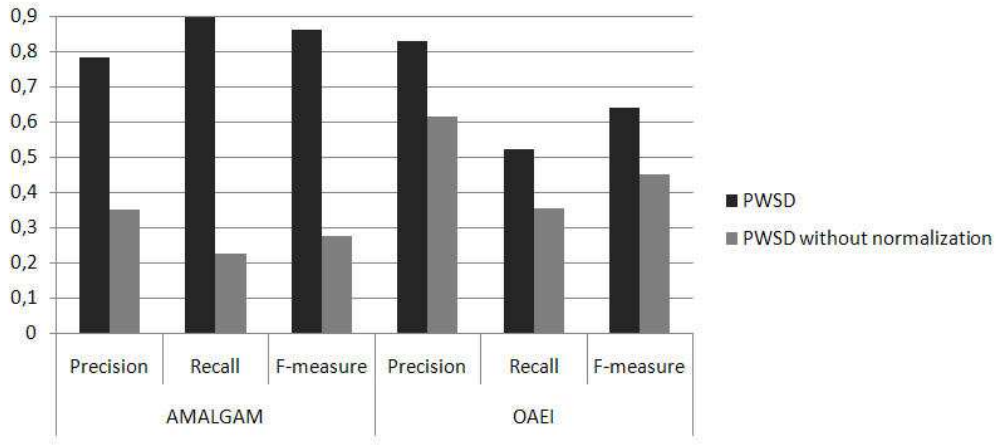

Figure 6.9: Evaluation of automatic annotation.

The evaluations have been executed on two test cases: the first is a set of three ontologies from the benchmark OAEI 2008[6]; the second is composed of two relational schemata of the well-known Amalgam integration benchmark for bibliographic data [Miller et al., 2001]. Details on the data sets are given in Table 6.1.

To evaluate the effectiveness of the disambiguation results and the lexical relationship discovered, the measures of precision, recall and F-measure defined in Section 2.3 are used. The results obtained by our method are compared with manually determined annotations/relationships. The true positives, i.e. correctly identified annotations/relationships, as well as the false positives, and the false negatives are computed.

6.6.1 Lexical Annotation Evaluation

The effectiveness of the automatic annotation performed by PWSD has been evaluated by using as preprocessing the NORMS tool. The results of PWSD has been compared with those obtained by an expert (i.e. a domain expert that knows the

[6] 101, 205 209 ontologies available at `http://oaei.ontologymatching.org/2008/benchmarks`.

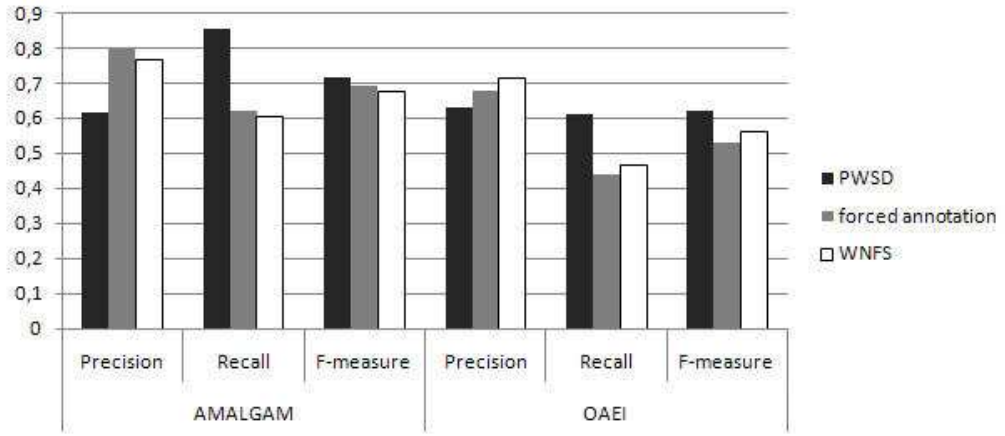

Figure 6.10: PWSD annotation (threshold of 0.2).

sources and their characteristics). The expert manually normalized each schema label and then associated one or more WordNet synsets to each element.

Let us suppose that for the element "name" the expert selected the synsets "$name_{\#1}$" and "$name_{\#2}$", while PWSD selected the synsets "$name_{\#1}$", "$name_{\#2}$" and "$name_{\#3}$". In this case, the precision of the automatic annotation is 0.66, the recall is 1 and the F-measure is 0.80. When the automatically normalized label does not correspond to the manually normalized one, the annotation is considered to be completely wrong (precision and recall equal 0).

In order to discard incorrect annotations, a threshold is applied. The output of PWSD has been investigated by varying the threshold between 0.05 and 0.4 in steps of 0.05. The optimal threshold value was found to be 0.2, which generated the results shown in Figures 6.9 and 6.10. However, the results did not substantially deviate w.r.t those returned with other thresholds in the range.

The importance of the normalization process is highlighted in Figure 6.9, where the results obtained by PWSD combined with the normalization process is compared with respect to the results of PWSD without the normalization process. Without schema label normalization, the recall obtained by PWSD rapidly decreases, due to a high number of non-dictionary words contained in the schemata (as shown in Table 6.1). However, the recall value improvement for the OAEI ontologies is significantly smaller than the improvement for the Amalgam schemata. This result is due to the presence in these ontologies of several non-endocentric CNs such as "writtenBy", "publishedBy", "InProceeding" (also called "prepositional verbs" in the literature). In contrast with NLP and with our experience on relational and XML schemata, the endocentric CNs in these ontologies represent only 56% of all CNs present in the sources. This result does not exclude the possibility of using normalization for ontologies, since the recall value is improved

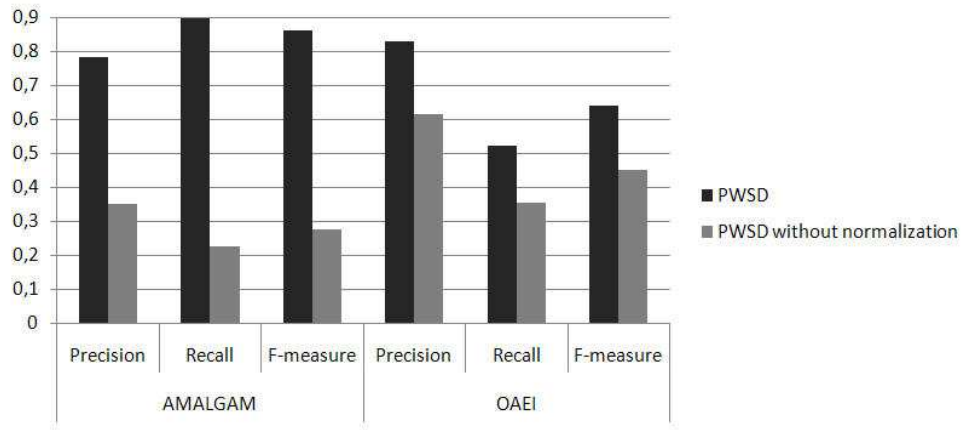

Figure 6.11: Evaluation of the lexical relationship discovery process.

anyway, but points to the necessity, especially for ontologies, to study a suitable method to annotate other kinds of multi-word labels[7].

In Figure 6.10, PWSD is compared with respect to: (1) forced annotation, which associates only the best annotation to each term (i.e. the one with the highest probability value) and (2) WNFS (a traditional baseline to evaluate WSD algorithms [McCarthy and Carroll, 2003]). The results show that PWSD outperforms both forced annotation and WNFS. By selecting more meanings for each term, the recall of PWSD outperforms the recall obtained by forced annotation and WNFS. Thus, even if the precision of PWSD decreases, the F-measure still dominates the one obtained by the other methods.

6.6.2 Lexical Relationship Discovery Evaluation

In this step of the evaluation, we only analyzed the relationships between two relational schemata of Amalgam and the relationships between two ontologies from OAEI (101, 209). We considered a limited set of schemata, as the process to manually determine lexical relationships is a very complex and time-consuming task, especially when more than two schemata are considered.

The evaluation was focused on the performance of our method with and without schema label normalization and on the loss of information caused by a forced approach (i.e. an approach that only maintains the relationship between two schema elements that is characterized by the highest probability value). During the evaluation, a probabilistic lexical relationship was considered correct if it was

[7]In the future, to improve the performance of the CN annotation method, we could study a method that is also able to deal with the presence of CNs such as “writtenBy”, “InProceeding” and conjunctions such as “and” or “or” in schema and ontology labels.

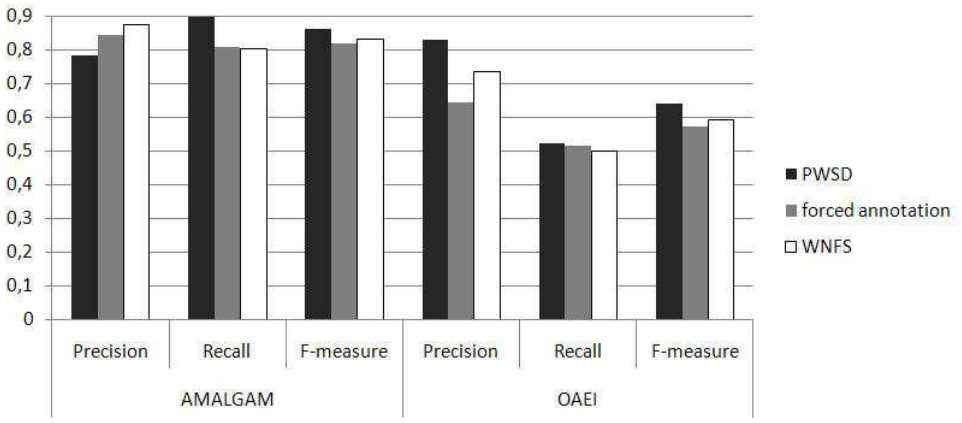

Figure 6.12: Evaluation of the relationship discovery process (threshold of 0.15).

present in the set of manually determined lexical relationships. For the evaluation of OAEI ontologies, we also considered the ontology alignments provided by OAEI, which we interpreted as SYN lexical relationships.

We investigated the value of the best threshold by varying it between 0.05 and 0.3 in steps of 0.05. The optimal threshold was found to be 0.15 which generated the results in Figure 6.11 and 6.12. However, as for the lexical annotation process, the results did not substantially deviate w.r.t those returned with other thresholds in the range.

In Figure 6.11, we compare the results of our own probabilistic lexical relationship discovery method with and without the normalization process on both the evaluation data sets (Amalgam and OAEI ontologies). Without schema label normalization, we discover few lexical relationships with low precision due to the presence of many false positive and false negative relationships. In particular, for the Amalgam schemata, where the majority of schema elements are labeled with CNs and abbreviations, we obtain very low values for both recall and precision.

In Figure 6.12, we compare the probabilistic lexical relationships discovered by our method with those discovered by using forced annotation and WNFS. With forced annotation and WNFS, the loss of information due to the annotation method is propagated to the lexical relationship discovery process. This caused a dramatic reduction in performance in terms of precision, recall and F-measure. On the contrary, by handling uncertainty during the lexical relationship discovery process, our method outperformed significantly both forced annotation and WNFS.

We obtained very good results for the relationship discovery process on Amalgam schemata, while the recall on OAEI ontologies was quite low. In the 209 ontology of OAEI, we find several terms, such as “publishedBy” and “writtenBy”,

	Time (sec)-Amalgam	Time (sec)-OAEI
Normalization	4.98	12.72
PWSD	9.13	30.19
Lexical Relationships	2.73	11.52
Total Time	16.84	54.43

Table 6.2: Average time performance of the whole method and its individual phases.

that should be associated by a SYN relationship with the terms of the 101 ontology "publisher" and "author". However, our method only found a RT relationship between these terms. On the Amalgam data set, we surprisingly obtained very good results for both precision and recall w.r.t lexical annotation. We discovered that the majority of terms wrongly annotated in the first Amalgam schema are not related to any term in the second Amalgam schema. As a consequence, the majority of wrong annotations do not affect the results of the lexical relationship discovery process.

6.6.3 Performance Evaluation

In order to evaluate the performance of our method, we have computed the average execution time of the whole probabilistic relationship discovery process and of each individual phase (schema label normalization, PWSD, lexical relationship discovery) for both the Amalgam and OAEI sources. The size of the data sets (146 elements for Amalgam, and 462 for OAEI ontologies) is sufficiently large to evaluate the running time of our method. It has been implemented using Java and the experiments have been carried on a PC compatible machine, with Intel Core Duo Processor (2.00 GHz), 2 GB RAM, running Windows Vista and JRE 1.6. Table 6.2 shows the average running times of our method and its phases. As it can be seen, the total average time to discover lexical relationships is reasonably low. Moreover, the results in Table 6.2 suggest that the execution time depends on the number of schema elements: the time needed to process OAEI schemata is higher than that for Amalgam as they contain more than three times the Amalgam schema elements.

In general, the performance and execution time of our method may be affected by the following factors:

- *Number of schema elements.* The number of schema elements represents the number of labels that have to be processed by our method. As a consequence, the greater the number of elements, the higher the time needed to complete the lexical relationship discovery process.

- *Number of non-dictionary words and their complexity*. Our method is able to directly annotate all the labels that exist in WordNet. All the other labels need to be normalized. Thus, the greater the number of non-dictionary words in the schema, the greater the time needed to process them. Moreover, the running time increases with the complexity of the labels (e.g., CNs composed by more than two words or complex abbreviations).
- *Polysemy of the labels.* The polysemy of a label indicates the number of possible meanings for the given label in WordNet. The more synsets exist for a label in WordNet, the longer PWSD will take during the WSD process for the identification of the probabilistic annotations.

The computational complexity of our method can be expressed as:

$$O(Normalization(n) + PWSD(n) + ProbabilisticRelationships(n))$$

where n is the number of source schema elements, $Normalization(n)$ is the cost of the normalization process, $PWSD(n)$ is the cost of the WSD algorithms plus the cost of the Dempster-Shafer's theory of combination and finally, $ProbabilisticRelationships(n)$ is the cost of the lexical relationship discovery process. The asymptotic complexities of the normalization process (equal to $O(n^2)$), of the WSD algorithms (equal to $O(n^4)$) and of the lexical relationship discovery process ($O(n^2)$) are polynomial. The asymptotic running time of the whole our method is determined by the implementation of the Dempster-Shafer's theory of combination that, as sketched in [Shafer, 1976], is (at worst) exponential (since the problem has been proved be #P-complete[8] in [Orponen, 1990]). However, the running times in Table 6.2 show that our method has performed well for the considered data sets.

In [Wilson, 2000] exact and approximate methods to reduce the Dempster-Shafer's theory complexity have been proposed. The study and the implementation of these approximate methods represent an interesting future line of research to enable our method to deal with very large data sets.

6.7 ALA: an Automatic Lexical Annotator

PWSD has been implemented in a tool integrated within the MOMIS system called ALA (Automatic Lexical Annotator). ALA extends the lexical annotation

[8] A problem is #P-complete if and only if it is in #P and every problem in #P can be reduced to it in polynomial time.

module of the MOMIS data integration system. However, it may be applied in general in the context of schema mapping discovery, ontology merging and data integration system and it is particularly suitable for performing "on-the-fly" data integration or probabilistic ontology matching.

ALA implements all the functionalities previously described and, in particular the five WSD algorithms described in Section 6.4.1. ALA assigns to each algorithm a reliability value (the default value of the reliability is the precision of the algorithm evaluated on a benchmark). The user can choose all or a subset of these algorithms and combine the algorithms outputs by using different operators:

- *Pipe* operator combines the annotation outputs of different algorithms provided in a given order. The pipe operator uses the output of the first algorithm and for the terms where no annotation is provided, executes the second algorithm and so on. With this operator, each term is disambiguated at most by a single algorithm. With pipe operator, if the SD algorithm is selected, it is the first algorithm applied. On the contrary, if the WFS algorithm is selected, it is always applied at the end. This rule is derived from the obvious observation that after the application of WFS every term that is contained in WordNet is disambiguated, so it is useless to apply other algorithms after.

- *Parallel* operator combines the annotation results from different algorithms by using the Dempster's rule of combination. With the parallel operator, each term is disambiguated with the contribution of all the selected algorithms.

- *Threshold* operator filters out the annotations with a probability under a given value.

By exploiting both structural and lexical knowledge, ALA provides a good quality probabilistic annotation drastically reducing human intervention and discovers probabilistic lexical relationships among schemata. A demo of ALA is available at `http://www.dbgroup.unimo.it/ALA/ALATool.mp4`, where for sake of simplicity, we consider only three data sources from the benchmark 2008 of the OAEI project[9], but the process is scalable and can be performed on several scenarios, thus, the user can provide her or his own set of data sources (the sources may be expressed on XML, OWL, RDF or the main formats for DBMS). The demo starts with the extraction and conversion in ODL_{I^3} of the schemata of the given set of data sources and the automatic extraction and inference of structural relationships. Then, the demo shows how the user may select among three different execution modalities: (1) *Default/Sequential* - the inexpert

[9] http://oaei.ontologymatching.org/2008/

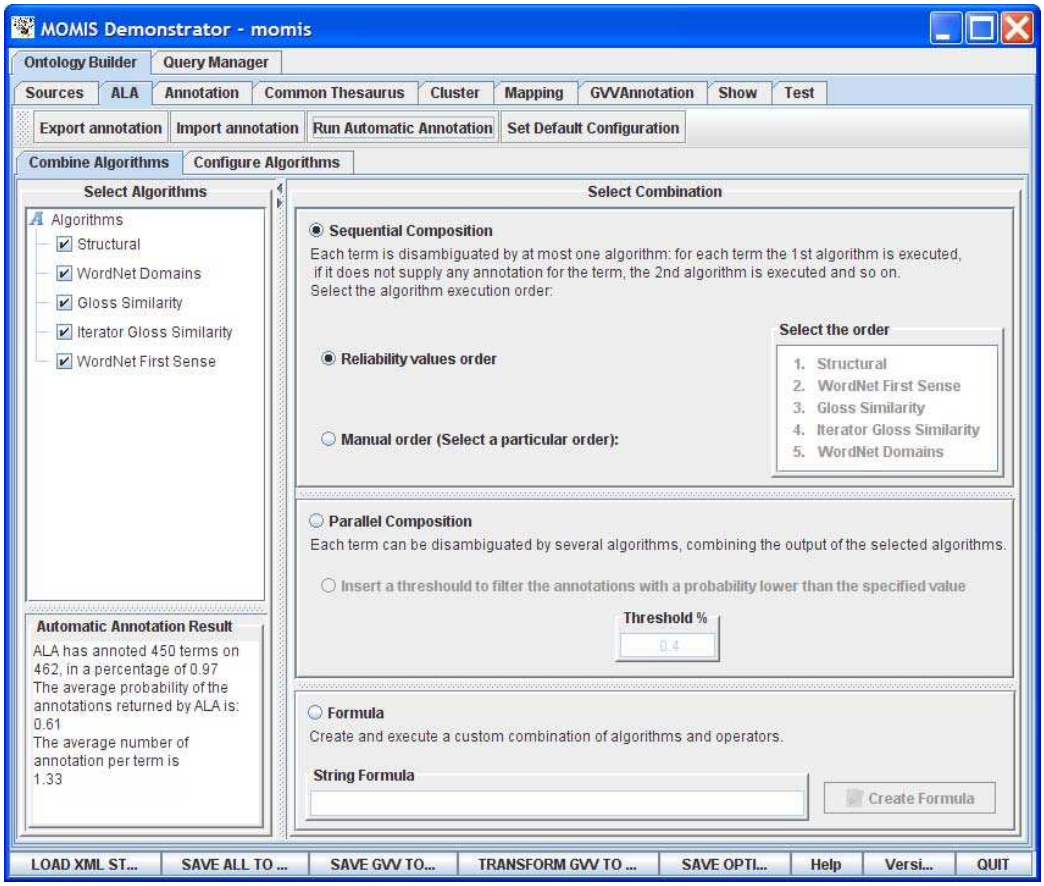

Figure 6.13: ALA and the PCT

user does not set any parameters; algorithms are executed by using the pipe operator following the reliability order (or a manual order); (2) *Parallel* - the skilled user may select the algorithms to be applied; the parallel execution can be performed without/with threshold filtering; (3) *Formula* - the skilled user may combine algorithms and operators as she/he wishes, using the GUI or directly writing the formula.

ALA is an effective annotation analysis tool. As shown in the demo, through the GUI the user may have an estimation of the quality of the obtained annotations in terms of the number of annotated terms, the average probability of the annotations and the number of annotations per term. Thus, a user may easily determine the right combination of WSD algorithms to optimize the process.

After the annotation, ALA computes the lexical relationships extraction: we demonstrate as the PCT is enriched with the discovered probabilistic lexical relationships.

6.8 Related Work

Modeling uncertainty in probabilistic schema matching has been an active area of research for some years [Nagy et al., 2006].

Our method takes inspiration from [Sarma et al., 2008, Dong et al., 2009], where the concept of probabilistic schema mapping is introduced and an algorithm for uncertain query answering is presented. This paper starts from initial probabilistic schema mappings, and without dealing with the generation of probabilistic mappings, proposes a probabilistic query answering method. The paper describes the requirements of a data integration system to support uncertainty; the authors maintain that data integration systems need to handle uncertainty at three levels: uncertain schema mappings, uncertain data and uncertain queries.

In the paper, probabilistic mediated schemata are formally defined and the problem of generating a probabilistic mapping between a source schema and a mediated schema is tackled. A mapping is constructed from a set of weighted attribute correspondences between a source schema and the mediated schema.

The goal of our paper is the generation of a set of probabilistic (and ordinary) relationships that can be assimilated to weighted correspondences and represent the first step in calculating a set of probabilistic mappings aimed at generating a probabilistic mediated schema.

In the literature many tools for automatic ontology mapping are offered, but only a few use a probabilistic approach.

Some authors working on ontology matching have proposed a method to resolve semantic ambiguity in order to filter the appropriate mappings between different ontologies [Gracia et al., 2007]. The limitation of this method is that it does not disambiguate the labels of the ontology classes, but only evaluates the possible meanings. In [Castano et al., 2008] a method for discovering schema mappings, based on the lexical relationships extracted from WordNet, is proposed. However, since it considers all the synsets associated to a term by WordNet, this approach does not realize any sort of disambiguation. The main disadvantage is the inclusion of wrong synsets and therefore the extraction of lexical relationships that can define erroneous mappings. For these reason, we propose a probabilistic WSD algorithm that ensures that more accurate relationships are identified.

[Preiss, 2004] proposes a combination of probabilistic WSD algorithms based on Bayes' theorem, demonstrating that it is a strong competitor to state-of-the art WSD algorithms.

Chapter 7

Conclusions and Future work

The main contributions of this thesis are the presentation and evaluation of lexical annotation methods in order to improve the accuracy of traditional schema matching systems.

As it was discussed, the implicit semantics of schema labels plays a fundamental role in the activity of discovering of mappings among data sources. In order to explicit such semantics a method to perform lexical annotation has to be devised. Starting from this consideration, I proposed a method to perform WSD called CWSD. CWSD is a method and a tool to perform semi-automatic annotation of structured and semi-structured data sources, which is composed by two main WSD algorithms: SD and WND. Instead of being targeted to textual data sources like most of the traditional WSD algorithms, CWSD exploits the structured knowledge of the data sources together with the lexical knowledge supplied by the lexical thesaurus WordNet and its extension WordNet Domains. Moreover, CWSD associates more than one meaning to a term and thus differs from the traditional WSD approaches. We evaluated CWSD within the MOMIS system. The experimental results showed the effectiveness of CWSD. Moreover, the structural knowledge of data sources significantly improves the disambiguation results (i.e., enhances the WND algorithm results).

As we argued in the thesis, the weakness of a thesaurus, like WordNet, is that it does not cover different domains of knowledge with the same detail and that many domain-dependent words, or non-dictionary words (such as compound nouns, abbreviations, and acronyms), may not be present in it. The result of automatic lexical annotation techniques is strongly affected by the presence of non-dictionary words in schemata. To address this problem, in this thesis, I presented an innovative method for schema label normalization which expands abbreviations and automatically annotates Compound Nouns (CNs) by enriching WordNet with new meanings. The experimental results showed the effectiveness of our method, which significantly improves the results of the automatic lexical an-

notation method, and, as a consequence, enhances the quality of the discovered inter-schema lexical relationships by reducing the number of false positive and false negative relationships. Moreover, the effectiveness of our method becomes even more evident for larger schemata. I showed that, due to the frequency of non-dictionary words in schemata, a schema matching system cannot ignore CNs and abbreviations without compromising *recall*.

As described in Chapter 2, there are scenarios where we need to discover mappings *on-the-fly*; as a consequence a fully automatic method to perform schema matching is required. The use of fully automatic methods gives rise to the problem of dealing with *uncertainty* during the lexical annotation process. In this thesis, I described an automatic lexical annotation method called PWSD. PWSD is a probabilistic WSD algorithm which associates to each annotation a probability value representing the reliability of the annotation itself. PWSD consists of five self-contained modules (the WSD algorithms) each producing a probability distribution on meanings, and it can be easily extended to the use of other WSD algorithms. PWSD combines the results of the single WSD algorithms by exploiting the Dempster-Shafer's theory of combination. Starting from the probabilistic annotations, it is possible to discover probabilistic lexical relationships among schemata. The probabilistic lexical relationships are collected in the PCT MOMIS component, as well as the ordinary structural relationships, that we extract from schemata by the description logic tool ODBTools.

PWSD has been evaluated within the MOMIS system, on two different data sets: three ontologies from the OAEI ontology alignment benchmark and two relational schemata from the Amalgam data integration benchmark. The experimental results showed the effectiveness of our method, which significantly improves the results of the automatic lexical annotation process and, as a consequence, also improves the quality of the discovered inter-schema lexical relationships. We verified that handling the uncertainty during the lexical annotation processes is beneficial and that, on complex integration problems, the information loss caused by the removal of uncertainty leads to a worsening of the schema relationship discovery process.

All the methods described and proposed in this thesis have been evaluated in the context of the MOMIS data integration system. However, they can be easily applied in the general context of schema and ontology matching, data warehouse, Web interface integration etc.

As regards to the label normalization method, future work will investigate two main problems identified during the experimental evaluation: (1) the presence of stop words (e.g. "to", "at", "and" etc.) and digits in schema labels; and (2) the problem of false negative non-dictionary words during the identification step (e.g. "RID" and "AID"). Moreover, future effort will be also devoted to: the inclusion and integration of other domain-specific resources (such as ontologies,

thesauri, glossaries and Wikipedia) to address the problem of the presence of specific domain terms in schemata (e.g., the biomedical term "aromatase" which is an enzyme involved in the production of estrogen); the use of multi-language lexical resources in order to be able to normalize and annotate schemata in different languages.

As regards to PWSD, we will also investigate an approach for clustering uncertain data by exploiting the discovered probabilistic relationships. The problem of clustering is well known and important in the data mining and management communities. The incorporation of uncertainty into the clustering techniques can significantly affect the quality of the underlying results as shown in [Kumar et al., 2002].

Another relevant future research line regards the inclusion of *instance-based matching techniques* to improve the automatic annotation and relationship discovery processes.

In particular, during the annotation process instance-based techniques can be used to solve the problem of *non-informative schema labels*: it is a common practice for companies to label the columns of a table by using codes (e.g. "IDS_XF02") which are not informative about the semantics of the schema. These labels cannot be automatically annotated and they are difficult to normalize even for a schema designer. Moreover, schemata may contain *misleading labels*: for instance, the label "phone", which in WordNet has the meaning of "electronic equipment", is often used in schemata to refer to phone numbers. The instance analysis may help to discover that the label "phone" refers to telephone numbers.

However, as previously described, one of the main drawback of the traditional instance-based techniques, such as Duplicate Detection [Bilke and Naumann, 2005], is that they are computationally expensive especially when we have to deal with a large set of data sources. To address this problem, future work will be devoted to employ RELEVANT [Bergamaschi et al., 2007e, Bergamaschi et al., 2007b], a tool for calculating the "relevant values" among the string values of an attribute. The tool has been conceived for improving the users knowledge of the attributes of database tables: by means of clustering techniques, RELEVANT provides to the designer a synthetic representation of the values of the attribute. By using RELEVANT, we can enrich the schema description by adding as metadata the relevant values of the schema attributes[1].

In this way, it will be possible to exploit the lexical and semantic information provided by these metadata in order to annotate "non-informative and misleading schema labels".

Relevant values can be exploited also during the normalization process in or-

[1]This is a partial solution as RELEVANT can be applied only on string values.

der to discover the correct expansion for a given abbreviation. Finally, during the lexical relationship discovery process, we can employ RELEVANT, to validate or reject the previously determined probabilistic relationships, thus improving accuracy.

Glossary

ALA (Automatic Lexical Annotator) is a tool integrated within MOMIS, implementing the lexical annotation methods proposed in this thesis. It permits to annotate schemata by combining several WSD algorithms.

Abbreviation Expansion is the task of finding a relevant expansion (long form) for a given abbreviation (short form).

BT (Broader Term) is an ODL_{I^3} relationship defined between two elements where the meaning of the first is more general (i.e. is a hypernym) than the meaning of the second (the opposite of BT is NT (Narrower Term)).

CN (Compound Noun) is a word composed of two or more words, called CN constituents. It is used to denote a concept, and can be interpreted based on the meanings of its constituents.

Common Thesaurus is the MOMIS repository of intra- and inter-schema relationships .It is an *Associative Network*, where nodes (class or attribute names) are connected through bidirectional relationships.

Compound Noun Interpretation is the task of determining the semantic relationship holding among the constituents of a CN.

CWSD (Combined Word Sense Disambiguation) is a method and a tool to perform semi-automatic annotation of structured and semi-structured data sources.

Lexical Annotation is the explicit assignment of meanings to a schema label with respect to a thesaurus or a reference lexical database.

Lexical Relationships are ODL_{I^3} relationships expressing intra- and inter-schema knowledge between two schemata. They are defined between

classes and attributes, and are specified by considering class/attribute labels. A lexical relationship exists between two schema labels, if there exists a semantic relationship between their meanings in WordNet.

Data Integration is the process of construction of a unified *global view*, starting from a set of independently developed schemata (with different structures, terminologies and semantics).

Dempster-Shafer's theory is a mathematical theory of evidence. It allows to combine evidence from different sources arriving at a degree of belief and considering all the available evidences.

Endocentric Compound Noun is a kind of CNs consisting of a head (i.e. the categorical part that contains the basic meaning of the whole CN) and modifiers, which restrict the meaning of the head. An endocentric CN exhibits a *modifier-head structure*, where the head noun occurs always after the modifiers.

FN (False Negative) are correct results not identified by the automatic/semi-automatic method.

FP (False Positive) are results falsely proposed by the automatic/semi-automatic method.

F-Measure is a weighted average of the Precision and Recall measures: $(F - Measure = 2 * (Precision * Recall)/(Precision + Recall)$.

GAV (Global-As-View) is a data integration approach based on the idea that the content of each element of the Global Schema should be characterized in terms of a view over the data sources. In this case, the mapping explicitly tells the system how to retrieve the data. Queries are processed by means of unfolding, i.e., by expanding the atoms according to their definitions (so as to come up with local schema relationships).

GLAV (Global-Local-As-View) is a data integration approach which mixes the GAV and LAV approaches: it can be considered as a variation of the LAV approach that allows the head of the view definition to contain any query on the local schemata.

Global Class is a class belonging to the Global Schema.

Global Schema (also called Global Virtual View) is a reconciled, integrated, and virtual view of the local sources, and offers a way to deal with the heterogeneity in the sources.

Gloss is the definition and explanation in natural language of the meaning of a term. In WordNet, each synset, is associated with one and only one gloss which can optionally include some example sentences.

Homonym is one of a group of words that share the same spelling and the same pronunciation but have different meanings.

Hypernym is a semantic relationship, which expresses that the meaning of a term A includes the meaning of another term B (i.e. A is hypernym of B); the opposite of hypernym is hyponym.

Hyponym is a semantic relationship, which expresses that the meaning of a term B is included in the meaning of another term A (i.e. B is hyponym of A); the opposite of hyponym is hypernym.

LAV (Local-As-View) is a data integration approach based on the idea that the content of each local source should be characterized in terms of a view over the Global Schema. Queries are processed by means of an inference mechanism that re-expresses the atoms of the Global Schema in terms of atoms of the local schemata.

Local Class is a class belonging to a local source.

Meronym is a semantic relationship denoting a constituent part of, or a member of something. That is, X is a meronym of Y if X is part/member of Y.

MOMIS (Mediator EnvirOment for Multiple Information Sources) is an Intelligent Data Integration framework designed for the integration of heterogeneous data sources that adopts a GAV approach. It has been developed by the DBGroup of the University of Modena and Reggio Emilia.

MT (Mapping Table) expresses mappings between local and global attributes. It is a table whose columns represent the Local Classes which belong to the Global Classes and whose rows represent the global attributes. Each table element represents how local attributes of a Local Class are mapped into a global attribute.

NeP4B (Network Peer for Business) is a MUR FIRB project[2]. Its goal is to create a network of semantic peers, related to each other by mappings. Each semantic peer exposes a Semantic Peer Data Ontology (SPDO), that is a Global Schema representing all the knowledge available in the peer, in terms of data sources, multimedia sources, and service related to the same domain.

[2] `http://www.dbgroup.unimo.it/nep4b`

NLP (Natural Language Processing) is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages.

Non-Dictionary Words are terms which do not have an entry in the lexical resource of reference. In this thesis, we recognize the following main kinds of non-dictionary words: compound nouns (e.g., "company address"), abbreviations and acronyms (e.g., "qty"), and domain specific terms (e.g., the biomedical term "aromatase", which is an enzyme involved in the production of estrogen).

NORMS (NORMalizer of Schemata) is a stand-alone tool implementing the schema label normalization method and providing an intuitive GUI that permits the designer to enhance the automatic results by correcting potential errors.

NT (Narrower Term) is an ODL_{I^3}relationship defined between two elements where the meaning of the first is included within the meaning of the second (i.e. is a hyponym) (the opposite of NT is BT (Broader Term)).

ODL_{I^3} (Object Definition Language) is an object-oriented language used in the MOMIS system to represent data; it is an extension of the *Object Definition Language* an object-oriented language developed by ODMG[3][4].

OWL (Web Ontology Language) is a family of knowledge representation languages for authoring ontologies. OWL is endorsed by the World Wide Web Consortium (W3C).

PCT (Probabilistic Common Thesaurus) is a probabilistic version of the CT containing which collects probabilistic relationships.

Precision is the number of the correct results returned by the automatic method divided by the total number of results that should have been returned.

Probabilistic Annotation is a lexical annotation where to each WordNet synset is associated a probability value which representing the reliability of the annotation itself.

Probabilistic Lexical Relationship is a lexical relationship which has associated a probability value representing the reliability of the relationship itself. The probability assigned to each relationships depends on the probability value of the meaning under consideration for an element.

[3] http://www.odmg.org/

[4] http://www.service-architecture.com/database/articles/odmg_3_0.html

PWSD (Probabilistic Word Sense Disambiguation) is an automatic algorithm that combines several WSD algorithms by using the Dempster-Shafer's theory of combination. It produces a probabilistic distribution on the possible meanings of a label.

Recall is the number of the correct results divided by the number of all the results returned by the automatic method

RT (Related Term) is an ODL$_{I^3}$relationship defined between two elements whose meanings are related in a meronymy (part-of/member-of) relationship or are sibling within a hierarchy.

Schema Label is the name of a class or an attribute in a schema.

Schema Label Normalization is the process of reduction of the form of each label to some standardized form. In particular, in this thesis, with schema label normalization, we mean the processes of abbreviation expansion and CN interpretation.

Schema Matching is a process that takes two heterogeneous schemata as input, and produces as output a set of *mappings*. Each mapping indicates that certain elements of a schema $S1$ are related to certain elements of the schema $S2$. Mappings may be obtained by using a set of semantic correspondences (e.g., location = area) between different schemata, as they capture the semantic relationships between concepts.

SD (Structural Disambiguation) is a WSD algorithm that tries to disambiguate schema labels by using semantic relationships inferred from the structure of data sources.

STASIS (SofTware for Ambient Semantic Interoperable Services)[5] is a IST FP6 STREP project (2006) which aims to create a comprehensive application suite which allows enterprises to simplify the mapping process between data schemata, by providing a GUI, allowing users to identify semantic elements in an easy way.

Structural Relationships are ODL$_{I^3}$relationships holding at intra-schema level. They are automatically extracted from the structure of schemata by analyzing each schema separately.

SYN (SYNonym) is an ODL$_{I^3}$relationship defined between two elements whose meanings are synonym.

[5] `http://www.stasis-project.net`

Synonym is a semantic relationship which expresses that two different words have the same meanings.

Synset is a WordNet term used to indicate a group of terms have the same cognitive meaning; a synset contains a group of synonymous words.

Thesaurus is a networked collection of controlled vocabulary terms with conceptual relationships between them; it arranges terms in a hierarchy and may contain other relationships such as antonyms, hypernyms etc.

TN (True Negative) are false results, which have also been correctly discarded by the automatic/semi-automatic method.

Tokenization is the process of breaking a stream of text up into words called tokens. Tokenization is a step of schema label normalization.

TP (True Positive) are the correct result that have been correctly returned by the automatic/semi-automatic method.

WND (WordNet Domain Disambiguation) is a WSD algorithm that tries to disambiguate schema labels by using the domain information supplied by the resource WordNet Domains.

WNEditor (WordNet Editor) is a tool integrated within the MOMIS system that permits to manually extend WordNet by creating new lemmas and synsets, and new relationships among the new elements and the rest of the WordNet hierarchy.

WordNet is an electronic lexical database for English based on psycholinguistic principles and maintained at Princeton University[6]. It is based on the notion of *synsets* or sets of synonyms. Its latest version, WordNet 3.0, contains about 155,000 words organized in over 117,000 synsets. WordNet also provides a *hypernym* (superconcept/subconcept) structure as well as other relationships such as *meronym* (part of relations).

WordNet Domains can be considered an extended version of WordNet realized by the Human Language Technology Unit of the Fondazione Bruno Kessler[7], in which each synset has been associated with one or more domain labels.

[6] `http://wordnet.princeton.edu/`
[7] `http://hlt.fbk.eu/`

Wrappers are MOMIS components which logically convert the source data structure into the ODL_{I^3} model. The wrapper architecture and interfaces are crucial, because wrappers are the focal point for managing the model diversity of data sources.

WSD (Word Sense Disambiguation) is the ability of identifying the meaning of words in a context by a computational technique.

WSD Algorithm (Word Sense Disambiguation Algorithm) is an effective method applied on a given set of words that makes use of one or more sources of knowledge to associate the most appropriate sense/senses with words in context.

Appendix A

The ODL$_{I^3}$ language syntax

The following is the BNF description of the ODL$_{I^3}$ description language. This object-oriented language, with an underlying Description Logic, is introduced for information extraction. The ODL$_{I^3}$ language is presented in [Bergamaschi et al., 2001], in the following I include the syntax fragments which differ from the original ODL grammar, referring to it for the remainder.

⟨interface_dcl⟩ ::= ⟨interface_header⟩
{[⟨ interface_body⟩]};
[**union** ⟨identifier⟩ { ⟨interface_body⟩ }**;**]

⟨interface_header⟩ ::= **interface** ⟨identifier⟩
[⟨inheritance_spec⟩]
[⟨type_property_list⟩]

⟨inheritance_spec⟩ ::= : ⟨scoped_name⟩
[,⟨inheritance_spec⟩]

Local schema pattern definition: the wrapper must indicate the kind and the name of the source of each pattern.

⟨type_property_list⟩ ::= **(** [⟨source_spec⟩]
[⟨extent_spec⟩]
[⟨key_spec⟩] [⟨f_key_spec⟩] [⟨c_key_spec⟩] **)**
⟨source_spec⟩ ::= **source** ⟨source_type⟩
⟨source_name⟩
⟨source_type⟩ ::= **relational** | **nfrelational**
| **object** | **file**
| **semistructured** | **multimedia**
| **glso**
⟨source_name⟩ ::= ⟨identifier⟩
⟨extent_spec⟩ ::= **extent** ⟨extent_list⟩
⟨extent_list⟩ ::= ⟨string⟩ | ⟨string⟩,⟨extent_list⟩
⟨key_spec⟩ ::= **key**[**s**] ⟨key_list⟩
⟨f_key_spec⟩ ::= **foreign_key** (⟨f_key_list⟩)
references ⟨key_list
⟩ [⟨f_key_spec⟩]
⟨c_key_spec⟩ ::= **candidate_key** ⟨identifier⟩
(⟨key_list⟩**)**

Global pattern definition rule, used to map the attributes between the global definition and the corrisponding ones in the local sources.

⟨attr_dcl⟩ ::= **[readonly] attribute**
[⟨domain_type⟩]
⟨attribute_name⟩ **[*]**
[⟨fixed_array_size⟩]
[⟨mapping_rule_dcl⟩]
⟨mapping_rule_dcl⟩ ::= **mapping_rule** ⟨rule_list⟩
⟨rule_list⟩ ::= ⟨rule⟩ | ⟨rule⟩**,**⟨rule_list⟩
⟨rule⟩ ::= ⟨local_attr_name⟩ |
'⟨identifier⟩**'**
⟨and_expression⟩ |
⟨union_expression⟩
⟨and_expression⟩ ::= **(** ⟨local_attr_name⟩ **and**
⟨and_list⟩ **)**
⟨and_list⟩ ::= ⟨local_attr_name⟩
| ⟨local_attr_name⟩ **and**
⟨and_list⟩
⟨union_expression⟩ ::= **(** ⟨local_attr_name⟩ **union**
⟨union_list⟩ **on** ⟨identifier⟩ **)**
⟨union_list⟩ ::= ⟨local_attr_name⟩
| ⟨local_attr_name⟩ **union**
⟨union_list⟩
⟨local_attr_name⟩ ::= ⟨source_name⟩**.**⟨class_name⟩**.**
⟨attribute_name⟩
...

Terminological relationships used to define the Common Thesaurus.

⟨relationships_list⟩ ::= ⟨relationship_dcl⟩**;** |
⟨relationship_dcl⟩**;**
⟨relationships_list⟩
⟨relationships_dcl⟩ ::= ⟨local_name⟩
⟨relationship_type⟩
⟨local_name⟩
⟨local_name⟩ ::= ⟨source_name⟩**.**
⟨local_class_name⟩
[**.**⟨local_attr_name⟩]
⟨relationship_type⟩ ::= **SYN** | **BT** | **NT** | **RT**
...

Extended base type definition for multimedia objects.

⟨BaseTypeSpec⟩ ::= ⟨FloatingPtType⟩ |
⟨IntegerType⟩ |
⟨CharType⟩ |
⟨BooleanType⟩ |
⟨OctetType⟩ |
⟨RangeType⟩ |
⟨AnyType⟩ |
⟨MultiMediaType⟩
⟨MultiMediaType⟩ ::= **Text** |
Image

OLCD integrity constraint definition: declaration of rule (using *if then* definition) valid for each instance of the data; mapping rule specification (*or* and *union* specification rule).

⟨rule_list⟩ ::= ⟨rule_dcl⟩**;** | ⟨rule_dcl⟩**;** ⟨rule_list⟩
⟨rule_dcl⟩ ::= **rule** ⟨identifier⟩ ⟨rule_spec⟩
⟨rule_spec⟩ ::= ⟨rule_pre⟩ **then** ⟨rule_post⟩ |
{ ⟨case_dcl⟩ }
⟨rule_pre⟩ ::= ⟨forall⟩ ⟨identifier⟩ **in** ⟨identifier⟩ **:**
⟨rule_body_list⟩
⟨rule_post⟩ ::= ⟨rule_body_list⟩
⟨case_dcl⟩ ::= **case of** ⟨identifier⟩ **:** ⟨case_list⟩
⟨case_list⟩ ::= ⟨case_spec⟩ | ⟨case_spec⟩ ⟨case_list⟩
⟨case_spec⟩ ::= ⟨identifier⟩ **:** ⟨identifier⟩ **;**
⟨rule_body_list⟩ ::= **(** ⟨rule_body_list⟩ **)** |
⟨rule_body⟩ |
⟨rule_body_list⟩ **and**
⟨rule_body⟩ |
⟨rule_body_list⟩ **and**
(⟨rule_body_list⟩ **)**
⟨rule_body⟩ ::= ⟨dotted_name⟩
⟨rule_const_op⟩
⟨literal_value⟩ |
⟨dotted_name⟩
⟨rule_const_op⟩
⟨rule_cast⟩ ⟨literal_value⟩ |
⟨dotted_name⟩ **in**
⟨dotted_name⟩ |
⟨forall⟩ ⟨identifier⟩ **in**
⟨dotted_name⟩ **:**
⟨rule_body_list⟩ |
exists ⟨identifier⟩ **in**
⟨dotted_name⟩ **:**
⟨rule_body_list⟩
⟨rule_const_op⟩ ::= = | ≥ | ≤ | > | <
⟨rule_cast⟩ ::= **(**⟨simple_type_spec⟩**)**
⟨dotted_name⟩ ::= ⟨identifier⟩ | ⟨identifier⟩**.**
⟨dotted_name⟩
⟨forall⟩ ::= **for all** | **forall**

Publications related to this thesis

- Sorrentino, S., Bergamaschi, S., and Gawinecki, M. (2011). NORMS: an automatic tool to perform schema label normalization. In Press, Accepted Manuscript (Demo Paper), *IEEE International Conference on Data Engineering, ICDE 2011*, April 11-16, Hannover.

- Bergamaschi, S., Beneventano, D., Po, L., Sorrentino, S. (2011). Automatic Schema Mapping through Normalization and Annotation. In Press in *Second Search Computing Workshop: Challenges and Directions*, 2010, LNCS State-of-the-Art Survey.

- Po, L. and Sorrentino, S. (2011). Automatic generation of probabilistic relationships for improving schema matching. *Information Systems Journal, Special Issue on Semantic Integration of Data, Multimedia, and Services*, 36(2):192208.

- Sorrentino, S., Bergamaschi, S., Gawinecki, M., and Po, L. (2010). Schema label normalization for improving schema matching. *DKE Journal*, 69(12):12541273.

- Sorrentino, S., Bergamaschi, S., Gawinecki, M., and Po, L. (2009). Schema normalization for improving schema matching. In proceedings of the *28th International Conference on Conceptual Modeling*, *ER 2009*, Gramado, Brasil, 9-12 November, pages 280-293.

- Beneventano, D., Bergamaschi, S., and Sorrentino, S. (2009) Extending WordNet with compound nouns for semi-automatic annotation in data integration systems. In proceeding of the *IEEE NLP-KE Conference*, Dalian, China, 24-27 September 2009.

- Bergamaschi, S., Po, L., Sorrentino, S., and Corni, A. (2009). Dealing with Uncertainty in Lexical Annotation. Revista de Informtica Terica e Aplicada, RITA, *ER 2009 Poster and Demonstrations Session*, 16(2):9396.

- Beneventano, D., Orsini, M., Po, L., Antonio, S., and Sorrentino, S. (2009). An ontology-based data integration system for data and multimedia sources. In Proceeding of *the Third IEEE International Conference on Semantic Computing, ICSC* 2009, Berkeley, CA, USA - September 14-16, pages 606611. IEEE Computer Society.

- Beneventano, D., Orsini, M., Po, L., and Sorrentino, S. (2009). The MOMIS-STASIS approach for Ontology-Based Data Integration. In proceedings of the *1st International Workshop on Interoperability through Semantic Data and Service Integration, ISDSI 2009*, Camogli (GE), Italy June 25.

- Bergamaschi, S., Po, L., Sorrentino, S., and Corni, A. (2009). Uncertainty in data integration systems: automatic generation of probabilistic relationships. Proceedings of *the VI Conference of the Italian Chapter of AIS, Itais 2009*, Costa Smeralda, Italy, October 2-3.

- Bergamaschi, S., Po, L., Sorrentino, S., and Corni, A. (2009). Uncertainty in data integration systems: automatic generation of probabilistic relationships. *Management of the Interconnected World* (A. DAtri, M. De Marco, A. Maria Braccini, F. Cariddu eds.), (Book Chapter) Springer, ISBN/ISSN: 978-3-7908-2403-2, 2009.

- Po, L., Sorrentino, S., Bergamaschi, S., and Beneventano, D. (2009). Lexical knowledge extraction: an effective approach to schema and ontology matching. Proceedings of *the European Conference on Knowledge Management, ECKM 2009*, 3-4 September Vicenza.

- Sorrentino, S. and Bergamaschi, S. (2009). Semi-automatic compound nouns annotation for data integration systems. Proceedings of *the 17th Italian Symposium on Advanced Database Systems, SEBD 2009*, Camogli (Genova), pages 221228, Italy June 21-24.

- Bergamaschi, S., Po, L., and Sorrentino, S. (2008). Automatic annotation for mapping discovery in data integration systems. Proceedings of *the Sixteenth Italian Symposium on Advanced Database Systems, SEBD 2008*, 22-25 June, Mondello, PA, Italy, pages 334341.

- Bergamaschi, S., Po, L., Sala, A., and Sorrentino, S. (2007). Data source annotation in data integration systems. In Proceedings of t*he fifth International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P)*, at 33st International Conference on Very Large Data Bases (VLDB 2007). University of Vienna, Austria, September 24.

- Bergamaschi, S., Po, L., and Sorrentino, S. (2007). Automatic Annotation in Data Integration Systems. In Proceeding of the *OTM Workshops*, Portugal, November 27-28.

Bibliography

[Abels et al., 2008a] Abels, S., Abels, H., and Cranner, P. (2008a). Simplifying e-business collaboration by providing a semantic mapping platform. In *I-ESA '08 Workshop*. Available at http://www.stasis-project.net/publications/publication.cfm?id=76.

[Abels et al., 2008b] Abels, S., Campbell, S., and Abels, H. (2008b). Stasis - creating an eclipse based semantic mapping platform. In *eChallenges 2008*. Available at http://www.stasis-project.net/publications/publication.cfm?id=83.

[Afrati and Kolaitis, 2008] Afrati, F. N. and Kolaitis, P. G. (2008). Answering aggregate queries in data exchange. In Lenzerini, M. and Lembo, D., editors, *PODS*, pages 129–138. ACM.

[Altinay, 2006] Altinay, H. (2006). On the independence requirement in dempster-shafer theory for combining classifiers providing statistical evidence. In *Applied Intelligence*, volume 25 (1), pages 73–90. Springer Netherlands.

[Aumueller et al., 2005] Aumueller, D., Do, H. H., Massmann, S., and Rahm, E. (2005). Schema and ontology matching with COMA++. In *Proc. of Special Interest Group on Managenegment of Data, SIGMOD'05, New York, NY, USA*, pages 906–908. ACM.

[Banek et al., 2008] Banek, M., Vrdoljak, B., and Tjoa, A. M. (2008). Word sense disambiguation as the primary step of ontology integration. In Bhowmick, S. S., Küng, J., and Wagner, R., editors, *DEXA*, volume 5181 of *Lecture Notes in Computer Science*, pages 65–72. Springer.

[Banerjee and Pedersen, 2002] Banerjee, S. and Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In Gelbukh, A. F., editor, *CICLing*, volume 2276 of *Lecture Notes in Computer Science*, pages 136–145. Springer.

[Barker and Szpakowicz, 1998] Barker, K. and Szpakowicz, S. (1998). Semi-Automatic Recognition of Noun Modifier Relationships. In *Proc. of the 36th*

Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, COLING-ACL'98, pages 96–102, August 10-14, Montreal, Quebec, Canada.

[Batini and Lenzerini, 1984] Batini, C. and Lenzerini, M., editors (1984). *A Methodology for data schema integration in the Entity Relationship model*.

[Beneventano et al., 2003a] Beneventano, D., Bergamaschi, S., Guerra, F., and Vincini, M. (2003a). Synthesizing an Integrated Ontology. *IEEE Internet Computing*, 7(5):42–51.

[Beneventano et al., 2003b] Beneventano, D., Bergamaschi, S., Guerra, F., and Vincini, M. (2003b). Synthesizing an Integrated Ontology. *IEEE Internet Computing Journal*, pages 42–51.

[Beneventano et al., 2003c] Beneventano, D., Bergamaschi, S., and Sartori, C. (2003c). Description logics for semantic query optimization in object-oriented database systems. *ACM Trans. Database Syst.*, 28:1–50.

[Beneventano et al., 2009a] Beneventano, D., Bergamaschi, S., and Sorrentino, S. (2009a). Extending WordNet with compound nouns for semi-automatic annotation in data integration systems. In *Proceedings of the IEEE NLP-KE Conference, Dalian, China, 24-27 September 2009*.

[Beneventano et al., 2008a] Beneventano, D., Dahlem, N., Haoum, S. E., Hahn, A., Montanari, D., and Reinelt, M. (2008a). Ontology-driven semantic mapping. In *International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2008)*, pages 329–342. Springer - Enterprise Interoperability III.

[Beneventano et al., 2008b] Beneventano, D., Guerra, F., Orsini, M., Po, L., Sala, A., Gioia, M. D., Comerio, M., de Paoli, F., Maurino, A., Palmonari, M., Gennaro, C., Sebastiani, F., Turati, A., Cerizza, D., Celino, I., and Corcoglioniti, F. (2008b). Detailed design for building semantic peer. *Networked Peers for Business, Deliverable D.2.1, Final Version, available at* $http://www.dbgroup.unimo.it/publication/d2\_1.pdf$, pages 52–57.

[Beneventano and Lenzerini, 2005] Beneventano, D. and Lenzerini, M. (2005). Final release of the system prototype for query management. sewasie, deliverable d.3.5, final version. http://www.dbgroup.unimo.it/prototipo/paper/D3.5Final.pdf.

[Beneventano and Montanari, 2008] Beneventano, D. and Montanari, D. (2008). Ontological mappings of product catalogues. In *Ontology Matching Workshop (OM 2008) at the 7th International Semantic Web Conference*, pages 244–249.

[Beneventano et al., 2009b] Beneventano, D., Orsini, M., Po, L., Sala, A., and Sorrentino, S. (2009b). An ontology-based data integration system for data and multimedia sources. In *ICSC*, pages 606–611. IEEE Computer Society.

[Beneventano et al., 2009c] Beneventano, D., Orsini, M., Po, L., and Sorrentino, S. (2009c). The MOMIS-STASIS approach for Ontology-Based Data Integration. In *the 1st International Workshop on Interoperability through Semantic Data and Service Integration, ISDSI 2009, Camogli (GE), Italy June 25*.

[Bergamaschi et al., 2011a] Bergamaschi, S., Beneventano, D., Guerra, F., and Orsini, M., editors (2011a). *Data Integration*.

[Bergamaschi et al., 2011b] Bergamaschi, S., Beneventano, D., Po, L., and Sorrentino, S. (2011b). *Automatic Schema Mapping through Normalization and Annotation*. In Press in Second Search Computing Workshop: Challenges and Directions, 2010, LNCS State-of-the-Art Survey.

[Bergamaschi et al., 2007a] Bergamaschi, S., Bouquet, P., Giacomuzzi, D., Guerra, F., Po, L., and Vincini, M. (2007a). Melis: an incremental method for the lexical annotation of domain ontologies. *IJSWIS*, 3(3):57–80.

[Bergamaschi et al., 1999] Bergamaschi, S., Castano, S., and Vincini, M. (1999). Semantic integration of semistructured and structured data sources. *Special Interest Group on Management of Data, SIGMOD Record*, 28(1):54–59.

[Bergamaschi et al., 2001] Bergamaschi, S., Castano, S., Vincini, M., and Beneventano, D. (2001). Semantic integration of heterogeneous information sources. *Data Knowl. Eng.*, 36(3):215–249.

[Bergamaschi et al., 2007b] Bergamaschi, S., Guerra, F., Orsini, M., Sartori, C., and Vincini, M. (2007b). Relevantnews: a semantic news feed aggregator. In Semeraro, G., Sciascio, E. D., Morbidoni, C., and Stoermer, H., editors, *SWAP*, volume 314 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Bergamaschi and Maurino, 2009] Bergamaschi, S. and Maurino, A. (2009). Toward a unified view of data and services. In Vossen, G., Long, D. D. E., and Yu, J. X., editors, *WISE*, volume 5802 of *Lecture Notes in Computer Science*, pages 11–12. Springer.

[Bergamaschi et al., 2007c] Bergamaschi, S., Po, L., Sala, A., and Sorrentino, S. (2007c). Data source annotation in data integration systems. In *Proceeding of the fifth International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P)*.

[Bergamaschi et al., 2007d] Bergamaschi, S., Po, L., and Sorrentino, S. (2007d). Automatic Annotation in Data Integration Systems. In Meersman, R., Tari, Z., and Herrero, P., editors, *OTM Workshops (1)*, volume 4805 of *Lecture Notes in Computer Science*, pages 27–28. Springer.

[Bergamaschi et al., 2008] Bergamaschi, S., Po, L., and Sorrentino, S. (2008). Automatic annotation for mapping discovery in data integration systems. In Gaglio, S., Infantino, I., and Saccà, D., editors, *Proc. of the Sixteenth Italian Symposium on Advanced Database Systems, SEBD, 22-25 June 2008, Mondello, PA, Italy*, pages 334–341.

[Bergamaschi et al., 2009a] Bergamaschi, S., Po, L., Sorrentino, S., and Corni, A. (2009a). Dealing with Uncertainty in Lexical Annotation. *Revista de Informtica Terica e Aplicada, RITA, ER 2009 Poster and Demonstrations Session*, 16(2):93–96.

[Bergamaschi et al., 2009b] Bergamaschi, S., Po, L., Sorrentino, S., and Corni, A. (2009b). Uncertainty in data integration systems: automatic generation of probabilistic relationships. In *VI Conference of the Italian Chapter of AIS, Itais 2009, Costa Smeralda, Italy, October 2-3*.

[Bergamaschi et al., 1997] Bergamaschi, S., Sartori, C., Beneventano, D., and Vincini, M. (1997). ODB-Tools: A Description Logics Based Tool for Schema Validation and Semantic Query Optimization in Object Oriented Databases. In Lenzerini, M., editor, *AI*IA*, volume 1321 of *Lecture Notes in Computer Science*, pages 435–438. Springer.

[Bergamaschi et al., 2007e] Bergamaschi, S., Sartori, C., Guerra, F., and Orsini, M. (2007e). Extracting relevant attribute values for improved search. *IEEE Internet Computing*, 11(5):26–35.

[Bilke and Naumann, 2005] Bilke, A. and Naumann, F. (2005). Schema matching using duplicates. In *ICDE*, pages 69–80. IEEE Computer Society.

[Boser et al., 1992] Boser, B. E., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *COLT*, pages 144–152.

[Bouquet et al., 2003] Bouquet, P., Serafini, L., and Zanobini, S. (2003). Semantic coordination: A new approach and an application. In Fensel, D., Sycara,

K. P., and Mylopoulos, J., editors, *International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 130–145. Springer.

[Bouquet et al., 2006] Bouquet, P., Serafini, L., Zanobini, S., and Sceffer, S. (2006). Bootstrapping semantics on the web: meaning elicitation from schemas. In Carr, L., Roure, D. D., Iyengar, A., Goble, C. A., and Dahlin, M., editors, *Proc. of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 505–512. ACM.

[Brody et al., 2006] Brody, S., Navigli, R., and Lapata, M. (2006). Ensemble Methods for Unsupervised WSD. In *ACL*. The Association for Computer Linguistics.

[Cali et al., 2002] Cali, A., Calvanese, D., Giacomo, G. D., and Lenzerini, M. (2002). Data integration under integrity constraints. In *Information Systems*, pages 262–279. Springer.

[Castano et al., 2001a] Castano, S., Antonellis, V. D., and di Vimercati, S. D. C. (2001a). Global Viewing of Heterogeneous Data Sources. *IEEE Trans. Knowl. Data Eng.*, 13(2):277–297.

[Castano et al., 2001b] Castano, S., Antonellis, V. D., and di Vimercati, S. D. C. (2001b). Semantic Integration of Heterogeneous Data Sources. *IEEE Transactions on Data and Knowledge Engineering*, 13(2).

[Castano et al., 2008] Castano, S., Ferrara, A., Lorusso, D., Näth, T. H., and Möller, R. (2008). Mapping Validation by Probabilistic Reasoning. In Bechhofer, S., Hauswirth, M., Hoffmann, J., and Koubarakis, M., editors, *ESWC*, volume 5021 of *Lecture Notes in Computer Science*, pages 170–184. Springer.

[Castano et al., 2003] Castano, S., Ferrara, A., and Montanelli, S. (2003). H-match: an algorithm for dynamically matching ontologies in peer-based systems. In Cruz, I. F., Kashyap, V., Decker, S., and Eckstein, R., editors, *SWDB*, pages 231–250.

[Castano et al., 2006] Castano, S., Ferrara, A., and Montanelli, S. (2006). Matching ontologies in open networked systems: Techniques and applications. *Journal of Data Semantics V*, pages 25–63.

[Chai et al., 2008] Chai, X., Sayyadian, M., Doan, A., Rosenthal, A., and Seligman, L. (2008). Analyzing and revising data integration schemas to improve their matchability. *The Proceedings of the Very Large Database Endowment, PVLDB*, 1(1):773–784.

[Chang et al., 2002] Chang, J. T., Schtze, H., and Altman, R. B. (2002). Creating an online dictionary of abbreviations from MEDLINE. *Journal of the American Medical Information Association*, 9(6):612–620.

[Chiticariu et al., 2008] Chiticariu, L., Kolaitis, P. G., and Popa, L. (2008). Interactive generation of integrated schemas. In [Wang, 2008], pages 833–846.

[Clifton et al., 1997] Clifton, C., Housman, E., and Rosenthal, A. (1997). Experience with a combined approach to attribute-matching across heterogeneous databases. In *In Proc. of the International Federation for Information Processing (IFIP) Working Conference on Data Semantics (DS-7), Leysin, Switzerland 1997, Leysin, Switzerland*.

[Comber et al., 2004] Comber, A. J., Law, A. N. R., and Lishman, J. R. (2004). A comparison of bayes', dempster-shafer and endorsement theories for managing knowledge uncertainty in the context of land cover monitoring. *Computers, Environment and Urban Systems*, 28(4):311 – 327.

[Dalvi and Suciu, 2007] Dalvi, N. N. and Suciu, D. (2007). Management of probabilistic data: foundations and challenges. In Libkin, L., editor, *PODS*, pages 1–12. ACM.

[DBGroup, 2010] DBGroup (2010). The momis tutorial. http://www.datariver.it/pdf/MOMIS_Tutorial.pdf.

[Dhamankar et al., 2004] Dhamankar, R., Lee, Y., Doan, A., Halevy, A. Y., and Domingos, P. (2004). imap: Discovering complex mappings between database schemas. In Weikum, G., König, A. C., and Deßloch, S., editors, *SIGMOD Conference*, pages 383–394. ACM.

[Do, 2006] Do, H. H. (2006). *Schema Matching and Mapping-based Data Integration: Architecture, Approaches and Evaluation*. Vdm Verlag Dr. Müller.

[Do et al., 2002] Do, H. H., Melnik, S., and Rahm, E. (2002). Comparison of Schema Matching Evaluations. In Chaudhri, A. B., Jeckle, M., Rahm, E., and Unland, R., editors, *Web, Web-Services, and Database Systems, NODe 2002 Web and Database-Related Workshops, Erfurt, Germany, October 7-10*, volume 2593 of *Lecture Notes in Computer Science*, pages 221–237. Springer.

[Do and Rahm, 2002] Do, H. H. and Rahm, E. (2002). COMA - A System for Flexible Combination of Schema Matching Approaches. In *VLDB*, pages 610–621.

[Doan et al., 2001] Doan, A., Domingos, P., and Halevy, A. Y. (2001). Reconciling schemas of disparate data sources: A machine-learning approach. In *SIGMOD Conference*, pages 509–520.

[Doan et al., 2004] Doan, A., Madhavan, J., Domingos, P., and Halevy, A. Y. (2004). Ontology matching: A machine learning approach. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 385–404. Springer.

[Domenico et al., 2009] Domenico, B., Francesco, G., Andrea, M., and Matteo, P. (2009). Unified semantic search of data and services. In *Third International Conference on Metadata and Semantics Research*.

[Dong et al., 2007] Dong, X. L., Halevy, A. Y., and Yu, C. (2007). Data Integration with Uncertainty. In Koch, C., Gehrke, J., Garofalakis, M. N., Srivastava, D., Aberer, K., Deshpande, A., Florescu, D., Chan, C. Y., Ganti, V., Kanne, C.-C., Klas, W., and Neuhold, E. J., editors, *VLDB*, pages 687–698. ACM.

[Dong et al., 2009] Dong, X. L., Halevy, A. Y., and Yu, C. (2009). Data integration with uncertainty. *VLDB J.*, 18(2):469–500.

[Embley et al., 2001] Embley, D. W., Jackman, D., and Xu, L. (2001). Multifaceted Exploitation of Metadata for Attribute Match Discovery in Information Integration. In *Proc. of International Workshop on Information Integration on the Web, WIIW, Rio de Janeiro, Brazil, April 9-11*, pages 110–117.

[Escudero et al., 2000] Escudero, G., Màrquez, L., and Rigau, G. (2000). Naive bayes and exemplar-based approaches to word sense disambiguation revisited. In Horn, W., editor, *ECAI*, pages 421–425. IOS Press.

[Euzenat et al., 2004] Euzenat, J., Loup, D., Touzani, M., and Valtchev, P. (2004). Ontology alignment with OLA. In *Proc. of the 3rd International Workshop for Evaluation of Ontology-based Tools- EON'04 located at the 3rd International Semantic Web Conference, ISWC, Hiroshima, Japan*.

[Euzenat and Shvaiko, 2007] Euzenat, J. and Shvaiko, P. (2007). *Ontology Matching*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Fan et al., 2003] Fan, J., Barker, K., and Porter, B. W. (2003). The Knowledge Required to Interpret Noun Compounds. In *Proc. of the 18th International Joint Conference on Artificial Intelligence, IJCAI, Acapulco, Mexico, August 9-15*, pages 1483–1485.

[Feild et al., 2006] Feild, H., Binkley, D., and Lawrie, D. (2006). An Empirical Comparison of Techniques for Extracting Concept Abbreviations from Identifiers. In *Proc. of Software Engineering and Applications, SEA'06, November, Dallas Texas*.

[Finin, 1980] Finin, T. W. (1980). The Semantic Interpretation of Nominal Compounds. In *Proc. of the AAAI Conference of Artificial Intelligence*, pages 310–312.

[Friedman et al., 1999] Friedman, M., Levy, A., and Millstein, T. (1999). Navigational plans for data integration. In *In Proceedings of the National Conference on Artificial Intelligence (AAAI*, pages 67–73. AAAI Press/The MIT Press.

[Gale et al., 1992] Gale, W. A., Church, K. W., and Yarowsky, D. (1992). A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439.

[Gangemi et al., 2003] Gangemi, A., Guarino, N., Masolo, C., and Oltramari, A. (2003). Sweetening WORDNET with DOLCE. *AI Magazine*, 24(3):13–24.

[Gawinecki, 2011] Gawinecki, M. (2011). *Reuse of Public Web Service: People-Aware Approaches?* PhD thesis, International Doctorate School in Information and Communication Technologies of the University of Modena and Reggio Emilia.

[Giunchiglia et al., 2005] Giunchiglia, F., Shvaiko, P., and Yatskevich, M. (2005). S-Match: an algorithm and an implementation of semantic matching. In Kalfoglou, Y., Schorlemmer, W. M., Sheth, A. P., Staab, S., and Uschold, M., editors, *Semantic Interoperability and Integration*, volume 04391 of *Dagstuhl Seminar Proceedings*. IBFI, Schloss Dagstuhl, Germany.

[Giunchiglia et al., 2007] Giunchiglia, F., Yatskevich, M., and Shvaiko, P. (2007). Semantic matching: Algorithms and implementation. *J. Data Semantics*, 9:1–38.

[Gliozzo et al., 2005] Gliozzo, A. M., Giuliano, C., and Strapparava, C. (2005). Domain Kernels for Word Sense Disambiguation. In *ACL*. The Association for Computer Linguistics.

[Gliozzo et al., 2004] Gliozzo, A. M., Strapparava, C., and Dagan, I. (2004). Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. *Computer Speech & Language*, 18(3):275–299.

[Gracia et al., 2007] Gracia, J., Lopez, V., d'Aquin, M., Sabou, M., Motta, E., and Mena, E. (2007). Solving semantic ambiguity to improve semantic web based ontology matching. In Shvaiko, P., Euzenat, J., Giunchiglia, F., and He, B., editors, *OM*, volume 304 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Halevy, 2001] Halevy, A. Y. (2001). Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294.

[Hernández et al., 2001] Hernández, M. A., Miller, R. J., and Haas, L. M. (2001). Clio: A semi-automatic tool for schema mapping. In *SIGMOD Conference*, page 607.

[Hill et al., 2008] Hill, E., Fry, Z. P., Boyd, H., Sridhara, G., Novikova, Y., Pollock, L. L., and Vijay-Shanker, K. (2008). AMAP: automatically mining abbreviation expansions in programs to enhance software maintenance tools. In Hassan, A. E., Lanza, M., and Godfrey, M. W., editors, *Proc. of the International Working Conference on Mining Software Repositories, MSR 2008, Co-located with ICSE, Leipzig, Germany, May 10-11*, pages 79–88. ACM.

[Hoffman and Murphy, 1993] Hoffman, J. C. and Murphy, R. R. (1993). Comparison of bayesian and dempster-shafer theory for sensing: A practitioner's approach. In *in SPIE Proc. on Neural and Stochastic Methods in Image and Signal Processing II*, pages 266–279.

[Hummel and Zucker, 1983] Hummel, R. and Zucker, S. (1983). On the foundations of relaxation labeling processes. *PAMI*, 5(3):267–287.

[Inmon, 1997] Inmon, W. (1997). What is a Data Warehouse? *Prism Tech. Topic*, 1(1).

[Islam et al., 2008] Islam, A., Inkpen, D. Z., and Kiringa, I. (2008). Applications of corpus-based semantic similarity and word segmentation to database schema matching. *VLDB J.*, 17(5):1293–1320.

[Jian et al., 2005] Jian, N., Hu, W., Cheng, G., and Qu, Y. (2005). Falconao: Aligning ontologies with falcon. In *Integrating Ontologies*.

[Kim and Baldwin, 2005] Kim, S. N. and Baldwin, T. (2005). Automatic Interpretation of Noun Compounds Using WordNet Similarity. In Dale, R., Wong, K.-F., Su, J., and Kwong, O. Y., editors, *Proc. of the Second International Joint Conference of Natural Language Processing, IJCNLP 2005, Jeju Island, Korea, October 11-13*, volume 3651 of *Lecture Notes in Computer Science*, pages 945–956. Springer.

[Kolaitis, 2005] Kolaitis, P. G. (2005). Schema mappings, data exchange, and metadata management. In Li, C., editor, *PODS*, pages 61–75. ACM.

[Kumar et al., 2002] Kumar, M., Patel, N. R., and Woo, J. (2002). Clustering seasonality patterns in the presence of errors. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 557–563.

[Le et al., 2006] Le, A.-C., Huynh, V.-N., Shimazu, A., and Dam, H.-C. (2006). Weighted combination of classifiers for word sense disambiguation based on dempster-shafer theory. In *RIVF*, pages 133–138. IEEE.

[Le et al., 2004] Le, B. T., Dieng-Kuntz, R., and Gandon, F. (2004). On ontology matching problems for building a corporate semantic web in a multi-communities organization. In *Proc. of the 6th International Conference on Enterprise Information Systems, ICEIS, April 14-17, Porto - Portugal*, pages 236–243.

[Lenat et al., 1990] Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. (1990). Cyc: Toward programs with common sense. *Commun. ACM*, 33(8):30–49.

[Lenzerini, 2002] Lenzerini, M. (2002). Data Integration: A Theoretical Perspective. In Popa, L., editor, *PODS*, pages 233–246. ACM.

[Lesk, 1986] Lesk, M. E. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the Fifth International Conference on Systems Documentation*, pages 24–26, Toronto, CA. ACM.

[Levi, 1978] Levi, J. N. (1978). *The Syntax and Semantics of Complex Nominals*. Academic Press, New York.

[Li, 2004] Li, J. (2004). LOM: A Lexicon-based Ontology Mapping Tool. In *Proc. of Performance Metrics for Intelligent Systems, PerMIS'04, Gaithersburg, MD, August 24-26*.

[Li and Clifton, 2000] Li, W. and Clifton, C. (2000). SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data Knowl. Eng.*, 33(1):49–84.

[Lin and Sandkuhl, 2008] Lin, F. and Sandkuhl, K. (2008). A Survey of Exploiting WordNet in Ontology Matching. In Bramer, M., editor, *Artificial Intelligence in Theory and Practice II*, volume 276 of *IFIP, International Federation for Information Processing*, pages 341–350. Springer Boston.

[Louie et al., 2007] Louie, B., Detwiler, L., Dalvi, N. N., Shaker, R., Tarczy-Hornoch, P., and Suciu, D. (2007). Incorporating uncertainty metrics into a general-purpose data integration system. In *SSDBM*, page 19. IEEE Computer Society.

[Madhavan et al., 2001] Madhavan, J., Bernstein, P. A., and Rahm, E. (2001). Generic Schema Matching with Cupid. In Apers, P. M. G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., and Snodgrass, R. T., editors, *Proc. of the 27th International Conference on Very Large Data Bases (VLDB 2001), September 11-14, 2001, Roma, Italy*, pages 49–58. Morgan Kaufmann.

[Magnini, 2000] Magnini, B. (2000). Experiments in Word Domain Disambiguation for Parallel Texts.

[Magnini et al., 2002] Magnini, B., Strapparava, C., Pezzulo, G., and Gliozzo, A. (2002). The Role of Domain Information in Word Sense Disambiguation. In *Natural Language Engineering, special issue on Word Sense Disambiguation*, pages 359–373.

[Mandreoli et al., 2005] Mandreoli, F., Martoglia, R., and Ronchetti, E. (2005). Versatile structural disambiguation for semantic-aware applications. In Herzog, O., Schek, H.-J., Fuhr, N., Chowdhury, A., and Teiken, W., editors, *CIKM*, pages 209–216. ACM.

[McCarthy and Carroll, 2003] McCarthy, D. and Carroll, J. (2003). Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences. *Computational Linguistics*, 29(4):639–654.

[Melnik et al., 2002] Melnik, S., Garcia-Molina, H., and Rahm, E. (2002). Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE*, pages 117–128. IEEE Computer Society.

[Mihalcea and Csomai, 2007] Mihalcea, R. and Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 233–242.

[Mihalcea and Moldovan, 2000] Mihalcea, R. and Moldovan, D. I. (2000). An Iterative Approach to Word Sense Disambiguation. In Etheredge, J. N. and Manaris, B. Z., editors, *FLAIRS Conference*, pages 219–223. AAAI Press.

[Miller et al., 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). WordNet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244.

[Miller et al., 2001] Miller, R. J., Fisla, D., Huang, M., Kalmuk, D., Ku, F., and Lee, V. (2001). The Amalgam Schema and Data Integration Test Suite.

[Milne et al., 2006] Milne, D. N., Medelyan, O., and Witten, I. H. (2006). Mining domain-specific thesauri from wikipedia: A case study. In *Web Intelligence*, pages 442–448. IEEE Computer Society.

[Mitra et al., 2005] Mitra, P., Noy, N. F., and Jaiswal, A. R. (2005). OMEN: A Probabilistic Ontology Mapping Tool. In Gil, Y., Motta, E., Benjamins, V. R., and Musen, M. A., editors, *Proc. of the International Semantic Web Conference, ISWC*, volume 3729 of *Lecture Notes in Computer Science*, pages 537–547. Springer.

[Moldovan et al., 2004] Moldovan, D., Badulescu, A., Tatu, M., Antohe, D., and Girju, R. (2004). Models for the semantic classification of noun phrases. In *Proc. of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics- Workshop on Computational Lexical Semantics, HLT-NAACL'04, Boston, Massachusetts*, pages 60–67.

[Nagy et al., 2006] Nagy, M., Vargas-Vera, M., and Motta, E. (2006). Dssim-ontology mapping with uncertainty. In Shvaiko, P., Euzenat, J., Noy, N. F., Stuckenschmidt, H., Benjamins, V. R., and Uschold, M., editors, *Ontology Matching*, volume 225 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Nastase et al., 2006] Nastase, V., Sayyad-Shirabad, J., Sokolova, M., and Szpakowicz, S. (2006). Learning Noun-Modifier Semantic Relations with Corpus-based and WordNet-based Features. In *Proc. of the 21th National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference, AAAI, July 16-20, Boston, Massachusetts, USA*.

[Naumann et al., 2004] Naumann, F., Freytag, J. C., and Leser, U. (2004). Completeness of integrated information sources. *Inf. Syst.*, 29(7):583–615.

[Navigli, 2009] Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).

[Navigli and Velardi, 2005] Navigli, R. and Velardi, P. (2005). Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1075–1086.

[Niles and Pease, 2001] Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of 2nd International Conference on Formal Ontology*

in Information Systems, FOIS 2001, Ogunquit, Maine, USA, October 17-19, pages 2–9.

[Nottelmann and Straccia, 2005] Nottelmann, H. and Straccia, U. (2005). splmap: A probabilistic approach to schema matching. In Losada, D. E. and Fernández-Luna, J. M., editors, *ECIR*, volume 3408 of *Lecture Notes in Computer Science*, pages 81–95. Springer.

[Novischi, 2004] Novischi, A. (2004). Combining Methods for Word Sense Disambiguation of WordNet Glosses. In Barr, V. and Markov, Z., editors, *FLAIRS Conference*. AAAI Press.

[Noy, 2004] Noy, N. F. (2004). Semantic Integration: A Survey Of Ontology-Based Approaches. *SIGMOD Record*, 33(4):65–70.

[Orponen, 1990] Orponen, P. (1990). Dempster's rule of combination is p-complete. *Artificial Intelligence*, 44:245–253.

[Orsini, 2004] Orsini, M. (2003/2004). Interoperabilitá tra ontologie eterogenee: i traduttori OWL - ODLI3. Master's thesis, University of Modena and Reggio Emilia.

[Orsini, 2009] Orsini, M. (2009). *Query Management in Data Integration Systems: the MOMIS approach*. PhD thesis, Ph.D. Dissertation. International Doctorate School in Information and Communication Technologies of the University of Modena and Reggio Emilia.

[Ó Séaghdha, 2008] Ó Séaghdha, D. (2008). *Learning compound noun semantics*. PhD thesis, Computer Laboratory, University of Cambridge. Published as University of Cambridge Computer Laboratory Technical Report 735.

[Pahikkala et al., 2005] Pahikkala, T., Pyysalo, S., Ginter, F., Boberg, J., Järvinen, J., and Salakoski, T. (2005). Kernels Incorporating Word Positional Information in Natural Language Disambiguation Tasks. In Russell, I. and Markov, Z., editors, *FLAIRS Conference*, pages 442–448. AAAI Press.

[Pantel and Lin, 2002] Pantel, P. and Lin, D. (2002). Discovering word senses from text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 613–619.

[Parsons and Hunter, 1998] Parsons, S. and Hunter, A. (1998). A review of uncertainty handling formalisms. In Hunter, A. and Parsons, S., editors, *Applications of Uncertainty Formalisms*, volume 1455 of *Lecture Notes in Computer Science*, pages 8–37. Springer.

[Pease et al., 2002] Pease, A., Niles, I., and Li, J. (2002). The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *In Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, page 2002.

[Pedersen et al., 2004] Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet: : Similarity - measuring the relatedness of concepts. In McGuinness, D. L. and Ferguson, G., editors, *AAAI*, pages 1024–1025. AAAI Press / The MIT Press.

[Plag, 2003] Plag, I. (2003). *Word-Formation in English*. Cambridge Textbooks in Linguistics. Cambridge University Press, New York.

[Po and Sorrentino, 2011] Po, L. and Sorrentino, S. (2011). Automatic generation of probabilistic relationships for improving schema matching. *In Press, Information Systems, Special Issue on Semantic Integration of Data, Multimedia, and Services*, 36(2):192–208.

[Po et al., 2009] Po, L., Sorrentino, S., Bergamaschi, S., and Beneventano, D. (2009). Lexical knowledge extraction: an effective approach to schema and ontology matching. In *Proceedings of the European Conference on Knowledge Management, 3-4 September Vicenza, ECKM 2009*.

[Preiss, 2004] Preiss, J. (2004). Probabilistic word sense disambiguation. *Computer Speech & Language*, 18(3):319–337.

[Preiss, 2006] Preiss, J. (2006). Probabilistic word sense disambiguation: Analysis and techniques for combining knowledge sources. Technical Report 673, University of Cambridge.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

[Rahm and Bernstein, 2001] Rahm, E. and Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The Very Large Data Bases-VLDB Journal*, 10(4):334–350.

[Ratinov and Gudes, 2004] Ratinov, L. and Gudes, E. (2004). Abbreviation expansion in schema matching and web integration. In *Proc. of the International Conference on Web Intelligence WI 2004, 20-24 September, Beijing, China*, pages 485–489. IEEE Computer Society.

[Resnik and Yarowsky, 2000] Resnik, P. and Yarowsky, D. (2000). Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–133.

[Rigau et al., 1997] Rigau, G., Atserias, J., and Agirre, E. (1997). Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. *CoRR*, cmp-lg/9704007.

[Rivest, 1987] Rivest, R. L. (1987). Learning decision lists. *Machine Learning*, 2(3):229–246.

[Roget, 1852] Roget, P. M. (1852). Roget's international thesaurus. *1st ed. Cromwell, New York, NY*.

[Rosario and Hearst, 2001] Rosario, B. and Hearst, M. (2001). Classifying the semantic relations in noun compounds. In *Proc. of the Conference on Empirical Methods in Natural Language Processing, EMNLP'01, June 3-4 2001, PA USA*.

[Sala, 2010] Sala, A. (2010). *Data and Service Integration: Architectures and Applications to Real Domains*. PhD thesis, University of Modena and Reggio Emilia.

[Sarma et al., 2008] Sarma, A. D., Dong, X., and Halevy, A. Y. (2008). Bootstrapping pay-as-you-go data integration systems. In [Wang, 2008], pages 861–874.

[Shadbolt et al., 2006] Shadbolt, N., Berners-Lee, T., and Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101.

[Shafer, 1976] Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

[Shvaiko and Euzenat, 2004] Shvaiko, P. and Euzenat, J. (2004). A classification of schema-based matching approaches. In *Proceedings of the Meaning Coordination and Negotiation Workshop at ISWC04*.

[Shvaiko and Euzenat, 2005] Shvaiko, P. and Euzenat, J. (2005). A survey of schema-based matching approaches. 3730:146–171.

[Shvaiko et al., 2010] Shvaiko, P., Giunchiglia, F., and Yatskevich, M. (2010). Semantic matching with S-Match. *Semantic Web Information Management: a Model-Based Perspective*, XX:183–202.

[Soanes and Stevenson, 2003] Soanes, C. and Stevenson, A. (2003). Oxford dictionary of english. *Oxford University Press, Oxford, U.K. Eds. 2003*.

[Sorrentino and Bergamaschi, 2009] Sorrentino, S. and Bergamaschi, S. (2009). Semi-automatic compound nouns annotation for data integration systems. In Antonellis, V. D., Castano, S., Catania, B., and Guerrini, G., editors, *SEBD*, pages 221–228. Edizioni Seneca.

[Sorrentino et al., 2011] Sorrentino, S., Bergamaschi, S., and Gawinecki, M. (2011). NORMS: an automatic tool to perform schema label normalization. In *ICDE 2011, April 11-16, Hannover, Germany*. Press, Accepted Manuscript, (Demo paper).

[Sorrentino et al., 2009] Sorrentino, S., Bergamaschi, S., Gawinecki, M., and Po, L. (2009). Schema normalization for improving schema matching. In *Conceptual Modeling - ER 2009, 28th International Conference on Conceptual Modeling, Gramado, Brazil, November 9-12, 2009. Proceedings*, pages 280–293.

[Sorrentino et al., 2010] Sorrentino, S., Bergamaschi, S., Gawinecki, M., and Po, L. (2010). Schema label normalization for improving schema matching. *DKE Journal*, 69(12):1254–1273.

[Straccia and Troncy, 2005] Straccia, U. and Troncy, R. (2005). omap: Results of the ontology alignment contest. In Ashpole, B., Ehrig, M., Euzenat, J., and Stuckenschmidt, H., editors, *Integrating Ontologies*, volume 156 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Su and Gulla, 2004] Su, X. and Gulla, J. A. (2004). Semantic Enrichment for Ontology Mapping. In *Proc. of the 9th International Conference on Applications of Natural Languages to Information Systems, NLDB 2004, Salford, UK, June 23-25*, pages 217–228.

[Su Nam Kim, 2008] Su Nam Kim, T. B. (2008). Standardised evaluation of english noun compound interpretation. In *Proc. of the International Conference on Language Resources and Evaluation, LREC, Workshop on Multiword Expressions MWEs*, pages 39–42, Marrakech, Morrocco.

[Suchanek et al., 2008] Suchanek, F. M., Kasneci, G., and Weikum, G. (2008). Yago: A large ontology from wikipedia and wordnet. *J. Web Sem.*, 6(3):203–217.

[Toutanova and Manning, 2000] Toutanova, K. and Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proc. of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP/VLC 2000*, pages 63–70, Morristown, NJ, USA. Association for Computational Linguistics.

[Ullman, 2000] Ullman, J. D. (2000). Information integration using logical views. *Theor. Comput. Sci.*, 239(2):189–210.

[Wang, 2008] Wang, J. T.-L., editor (2008). *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*. ACM.

[Widdows and Dorow, 2002] Widdows, D. and Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *COLING*.

[Wikipedia, 2004] Wikipedia (2004). Wikipedia, the free encyclopedia. [Online; accessed 22-July-2004].

[Wilson, 2000] Wilson, N. (2000). Algorithms for dempster-shafer theory. In *Algorithms for Uncertainty and Defeasible Reasoning*, pages 421–475. Kluwer Academic Publishers.

[Wong et al., 2006] Wong, W., Liu, W., and Bennamoun, M. (2006). Integrated scoring for spelling error correction, abbreviation expansion and case restoration in dirty text. In *Proc. of the fifth Australasian conference on Data mining and analystics, AusDM '06*, pages 83–89, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

[Yarowsky, 1994] Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In *ACL*, pages 88–95.

[Yeates et al., 2000] Yeates, S., Bainbridge, D., and Witten, I. H. (2000). Using Compression to Identify Acronyms in Text. In *Proc. of the Data Compression Conference, DCC'00*, Washington, DC, USA. IEEE Computer Society.