Schema Label Normalization for Improving Schema Matching

Serena Sorrentino, Sonia Bergamaschi, Maciej Gawinecki, Laura Po

Dipartimento di Ingegneria dell'Informazione
University of Modena and Reggio Emilia
via Vignolese 905, 41125 Modena, Italy

Abstract

Schema matching is the problem of finding relationships among concepts across heterogeneous data sources that are heterogeneous in format and in structure. Starting from the "hidden meaning" associated with schema labels (i.e. class/attribute names) it is possible to discover relationships among the elements of different schemata. Lexical annotation (i.e. annotation w.r.t. a thesaurus/lexical resource) helps in associating a "meaning" to schema labels. However, the performance of semi-automatic lexical annotation methods on real-world schemata suffers from the abundance of non-dictionary words such as compound nouns, abbreviations, and acronyms. We address this problem by proposing a method to perform schema label *normalization* which increases the number of comparable labels. The method semi-automatically expands abbreviations/acronyms and annotates compound nouns, with minimal manual effort. We empirically prove that our normalization method helps in the identification of similarities among schema elements of different data sources, thus improving schema matching results.

Keywords: schema matching, normalization, natural language for DKE, lexical annotation, interoperability, heterogeneity.

Email address: `firstname.lastname@unimore.it` (Serena Sorrentino, Sonia Bergamaschi, Maciej Gawinecki, Laura Po)

1. Introduction

Schema matching is a critical step in many applications, including: data integration, data warehousing, e-business, semantic query processing, peer data management, and semantic web applications. In this work, we focus on schema matching in the context of data integration [1], where the goal is the creation of mappings between heterogeneous data sources (heterogeneous in format and structure). Mappings are obtained by a schema matching system by using a set of semantic matches (e.g. location = area) between different schemata. A powerful means to discover matches is the understanding of the "meaning" behind the names denoting schema elements, "labels" in the following [2]. In this context, lexical annotation, i.e. the explicit association of a meaning to a label w.r.t. a thesaurus (WordNet [3] in our case) is a key tool.

The strength of a thesaurus, like WordNet (WN), 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 Compound Nouns (CNs) (e.g. "company address"), abbreviations (e.g. "QTY") and acronyms (e.g. WSD-Word Sense Disambiguation).

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. In the following, we 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 mapping accuracy.

A manual process of schema label normalization is laborious, time consuming and itself prone to errors. Starting from our previous work on semi-automatic lexical annotation of structured and semi-structured data sources [4], we propose a semi-automatic method for the normalization of schema labels that is able to expand abbreviations and acronyms, and to enrich WN with new CNs. Our approach uses only schema-level information and can thus be used in scenarios where data instances are not available [5].

Our method is implemented in the MOMIS (Mediator envirOnment for Multiple Information Sources) system [1, 6]. However, it may be applied in

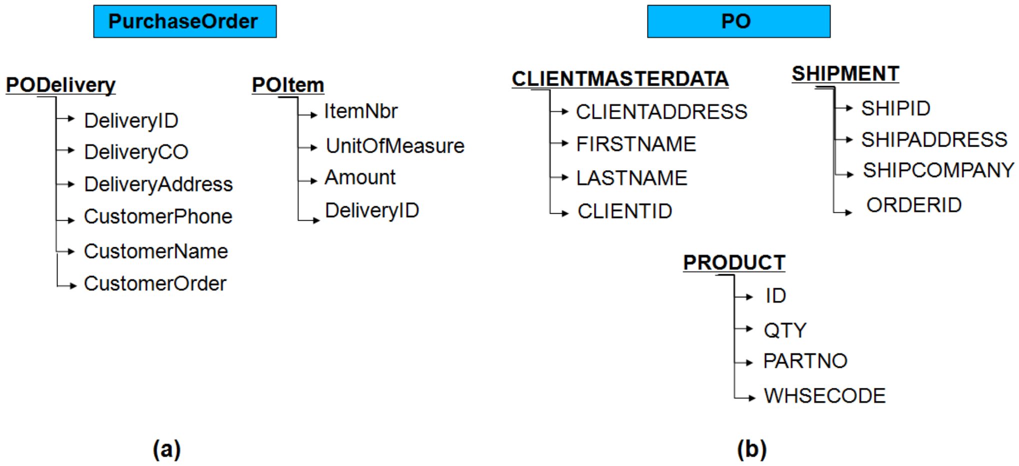


Figure 1: Graph representation of two schemata with elements containing abbreviations and CNs: (a) relational database schema, (b) XML schema.

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.

The rest of the article is organized as follows. In Section 2, we define the problem in the context of schema matching; in Section 3, a brief overview of the method is given; in Sections 4, 5, and 6, we describe the subsequent phases of our method: label preprocessing, abbreviation expansion and CN interpretation. In Section 7, we demonstrate the effectiveness of the method with extensive experiments on real-world data sets. A comparison of our method with related work is presented in Section 8. Finally, in Section 9, we make some concluding remarks and illustrate our future work.

2. Problem definition

Element labels represent an important source for assessing similarity between schema elements. This can be done semantically by comparing their meanings.

Definition 1. *Lexical annotation of a schema label is the explicit assignment of a meaning to the label w.r.t. a thesaurus.*

Starting from the lexical annotation of schema labels, we can derive lexical relationships between them on the basis of the semantic relationships defined in WN between their meanings.

Definition 2. *Let S and T be two heterogeneous schemata, and $E_S = \{s_1, ..., s_n\}$ and $E_T = \{t_1, ..., t_k\}$, respectively, the set of labels of S and T. A lexical relationship is defined as the triple $< s_i, t_j, R >$ where $s_i \in E_S$, $t_j \in E_T$ and R specifies a lexical relationship between s_i and t_j. The lexical relationships are:*

- *SYN (SYNonym-of): defined between two labels that are synonymous (it corresponds to a WN synonym relationship);*
- *BT (Broader Term): defined between two labels where the first is more general then the second (the opposite of BT is NT, Narrower Term; it corresponds to a WN hypernym/hyponym relationship);*
- *RT (Related Term): defined between two labels that are in a meronym hierarchy (it corresponds to a WN meronym relationship).*

Definition 3. *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.*

Definition 4. *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.*

In the following we will refer to both abbreviations and acronyms with the term *abbreviations*.

Figure 1 shows two schemata to be integrated which contain 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” w.r.t. WN. Schema label normalization (also called *linguistic normalization* in [7]) is the reduction of the form of each label to some standardized form. In our case, with label normalization we mean the processes of abbreviation expansion and CN interpretation.

Definition 5. *The interpretation of a CN is the task of determining the semantic relationship that hold among the constituents of a CN.*

Definition 6. *Abbreviation expansion is the task of finding a relevant expansion (long form) for a given abbreviation (short form).*

Schema label normalization improves the schema matching process by reducing the number of discovered *false positive/false negative relationships.*

Definition 7. *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.*

For example, let us consider the two schema labels “CustomerName” and “CLIENTADDRESS”, to be found in the schemata “PurchaseOrder” a nd “PO” respectively (Figure 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 WN meaning. In this way, a false positive relationship is discovered because these two CNs represent “semantically distant” schema elements.

Other approaches in the literature [8, 9] 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 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 8. *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 1). Without abbreviation expansion, we would not discover that there exists a SYN relationship between the elements “amount” and “QTY”.

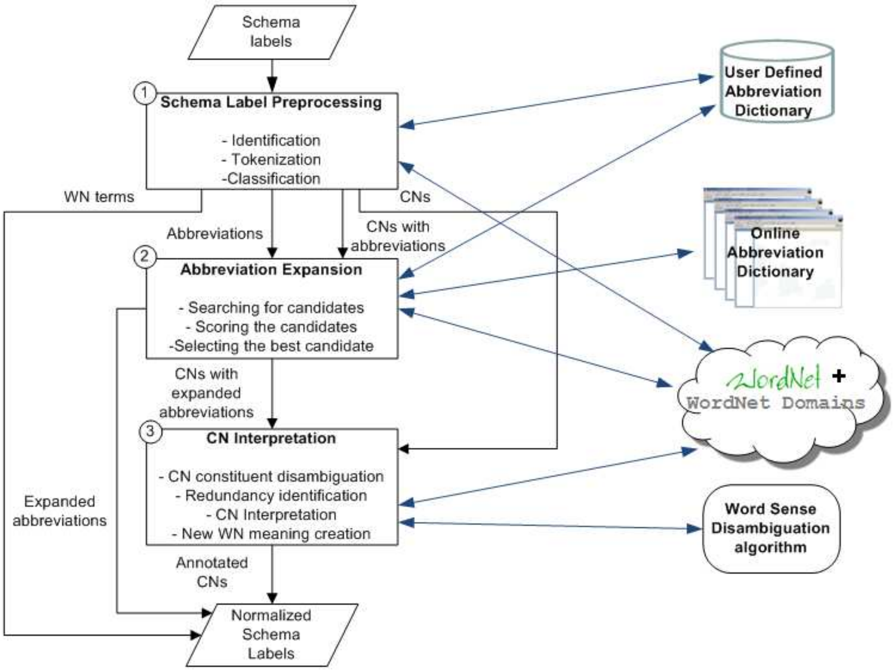

Schema labels
1
Schema Label Preprocessing
- Identification
- Tokenization
-Classification
WN terms
CNs
Abbreviations
CNs with abbreviations
2
Abbreviation Expansion
- Searching for candidates
- Scoring the candidates
-Selecting the best candidate
CNs with expanded abbreviations
3
CN Interpretation
- CN constituent disambiguation
- Redundancy identification
- CN Interpretation
- New WN meaning creation
Expanded abbreviations
Annotated CNs
Normalized Schema Labels
User Defined Abbreviation Dictionary
Online Abbreviation Dictionary
WordNet +
WordNet Domains
Word Sense Disambiguation algorithm

Figure 2: Overview of the schema label normalization method.

3. Overview of the schema label normalization method

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

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 "DeliveryCO" belonging to the "PurchaseOrder" schema in Figure 1.

Schema Label Preprocessing
The input of schema label preprocessing is the set of schema element labels. During this phase, we automatically select the labels to be normalized (for details see Section 4). The output of this module are the tokenized labels classified into four groups (as shown in Figure 2): *WN terms* (i.e. labels having an entry in WN 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 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). For instance, the abbreviation "CO" of the CN "DeliveryCO", is expanded as "Company".

CN Interpretation
The input of CN interpretation are the schema labels classified as CNs and CNs with expanded abbreviations(as shown in Figure 2). During this phase, the constituents of a CN are annotated w.r.t. WN by applying a WSD (Word Sense Disambiguation) algorithm; then, starting from these annotations we discover a semantic relationship between the constituents (for details see Section 6). For instance, for the CN "Delivery Company", the semantic relationship "MAKE" is selected. In the end, a new WN meaning for the CN is created and inserted in WN.

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").

4. 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 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 WN 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 WN need no normalization. Moreover, we identify a set of exceptions (called *schema standard abbreviations* in Section 5) that, although they have an entry in WN are mostly used as abbreviations in the context of schemata (e.g. "id", which is a state in the Rocky Mountains in WN, is often used as a short form of "identifier" in schemata). All such abbreviations are gathered in a "user-defined dictionary" and automatically identified for normalization.

Definition 9. *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 WN.*

Step 2. Tokenization

This step tokenizes the previously identified labels by using one of the approaches described in [10]: 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. We consider two alternative variants of greedy tokenization: GT/WN, which makes use of WN to identify dictionary words, and GT/Ispell, that makes use of the English word list in Ispell[1].

Step 3. Classification

This step classifies tokenized labels into four groups: dictionary words that exist in WN, 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 WN 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[2].

For instance, let us assume we are preprocessing the "DeliveryCO" label (shown in Figure 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".

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

[2] We decided to omit this further classification step in Figure 2 for increased clarity - the Figure would have contained a confusion of arrows.

Schema from	**Type**	**URL**
Freeway	relational	http://freeway.sourceforge.net
Zen Cart E-Commerce	relational	http://zencart.sourceforge.net
phpMyAdmin	relational	http://phpmyadmin.net
MediaWiki	relational	http://mediawiki.org
eCanteen	relational	http://ecanteen.sourceforge.net
Freeside	relational	http://freshmeat.net/projects/freeside
ImpressCMS	relational	http://impresscms.sourceforge.net
Open Travel Alliance	XSD	http://opentravel.org
Geography Markup Language	XSD	http://www.opengeospatial.org/standards/gml
XML Common Business Language	XSD	http://xcbl.org
XWebTD Web service	XSD	http://ws.xwebservices.com/XWebTD/V1/Order_Types.xsd
Extended camera ontology	OWL	http://hnsp.inf-bb.uni-jena.de/opossum/

Table 1: Schemata analyzed for manual abbreviation expansion.

5. Abbreviation expansion

A schema can contain both *standard* and *ad hoc* abbreviations. Standard abbreviations either denote important and frequent domain concepts (*domain standard abbreviations*), e.g. "Co" (Company), or are commonly used by schema designers but do not belong to any specific domain (*schema standard abbreviations*), e.g. "Nbr" (Number). For instance, the OTA standard[3] contains a list of recommended schema standard abbreviations. On the other hand, ad hoc abbreviations are mainly created by a schema designer to save space, and include from phrases that would not be abbreviated in textual sources (where there are no term length limitation) [11, 12].

5.1. Expansion resources

To observe how different types of abbreviations can be handled automatically, we analyzed short forms and their corresponding long forms in several open-source schemata (see Table 1 for a list). Based on our manual inspection, we found four *expansion resources* relevant for finding possible long form candidates: (1) *local context* (LC), (2) *complementary schemata*

[3] OpenTravel Alliance XML schema for the travel industry. Available online at http://www.opentravel.org/.

(CS), (3) an *online abbreviation dictionary* (OD), and (4) a *user-defined dictionary* (UD). To define the *local context* and the *complementary schemata* resources, let us suppose sf to be a short form identified in a schema label l. The label l is either an attribute name of a class c or a class name belonging to a schema s. The local context of sf is then the class c or the schema s. The complementary schemata are the other schemata that have to be integrated with the schema s. Local context and complementary schemata are particularly relevant for expanding ad hoc abbreviations and may contain relevant expansions because they belong to the same domain or related domains. It is common practice to abbreviate a class name when it is used in an attribute name (for instance, the "SHIPMENT" class in Figure 1 contains the attributes "SHIPADDRESS", "SHIPID" and "SHIPCOMPANY", where "SHIP" is an abbreviation for "SHIPMENT"). For the complementary schemata, we observed, for instance, that the short form "UOM" in the XML schema (Figure 1b) can be expanded with long form "Unit Of Measure" from the relational database schema (Figure 1a). An online abbreviation dictionary (in our case Abbreviations.com[4]) is particularly useful for expanding domain standard abbreviations. Online abbreviation dictionaries are co-authored by online communities and thus scale well to the number of abbreviations, possible expansions and covered domains.

Finally, the user-defined dictionary is initially bootstrapped with schema standard abbreviations from schema design guidelines for the OTA standard. Since real-world schemata in companies often use company-specific codes (e.g. "X09CCDE") that will not appear in any public dictionary, the designer may enrich the user-defined dictionary with such abbreviations.

5.2. Abbreviation expansion algorithm

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 2): (1) searching candidate long forms; (2) scoring the candidate long forms; (3) selecting the most appropriate long form. In the following, we describe each step in detail.

Step 1. Searching candidate long forms from expansion resources We look for possible long form candidates in the local context and in the com-

[4] `http:\\www.abbreviations.com`

Pattern	**Regular expression**	**Short form**	**Long form**
Acronym	c_0[a-z]+c_1[a-z]+$\ldots$[a-z]+c_n	*mfag*	***m**edical* ***f**irst* ***a**id* ***g**uide*
Prefix	sf[a-z]+	*dep*	***dep**artment*
Dropped Letter	c_0[a-z]*c_1[a-z]*$\ldots$[a-z]*c_n	*dept*	***dep**artment*
Combination Word	c_0[a-z]*?c_1[a-z]*?$\ldots$[a-z]*?c_n	*pdef*	***p**eriod* ***def**ined* *first*

Table 2: List of abbreviation patterns given in the order in which they are evaluated for a short form [13]. A pattern is a regular expression created from the characters of a short form: $sf = c_o c_1 \ldots c_n$. All regular expressions are case insensitive.

plementary schemata using the four abbreviation patterns proposed in [13] and listed in Table 2. These abbreviation patterns cover the most common strategies of abbreviating a phrase in English[5] and are used in a certain order, starting from the most conservative to avoid matching incorrect expansions. Only the first matching candidate in the text is considered. Moreover, the algorithm tries to find an entry for the target sf into the online and user-defined dictionaries; it returns all the long forms returned by these two expansion resources.

Let us focus on the expansion of the "CO" abbreviation contained in the "DeliveryCO" label. The local context of "DeliveryCO", in this case, is its schema, while the schema "PO" is the complementary schema. The abbreviation expansion algorithm receives the following expansions: (a) {"Company", "Colorado", and "Check Out"} from the online dictionary (b) no expansion from the local context, (c) {"Company"} from the complementary schemata. Next, the algorithm merges the lists of long form candidates into a single list: {"Company", "Colorado", "Check Out"}.

Step 2. Scoring the candidate long forms

For the user-defined dictionary, the local context and the complementary schemata the score of lf_i is 1 if lf_i is found in the given resource; otherwise, the score is 0.

[5] Abbreviation patterns may vary depending on the language they are written in. In this work, we focus on the English language, however, as future work we are interested to extend our method to deal with multi-language schemata.

The online dictionary may suggest more than one long form for a given short form. When multiple long forms are suggested, we propose a technique based on two factors: (a) the number of domains a given long form shares with the schemata to be integrated and (b) its "popularity" in these domains[6]. The information about the domain of a long form and its domain-specific relevance can be found in online dictionaries like Abbreviations.com.

The entries in the online dictionary for a short form sf can be modeled as a combination of a long form lf_i and the domain d in which it appears with the associated popularity $p(< lf_i; d >)$. To compute a score for each online dictionary expansion, we need to identify the main domains of the schemata to be integrated. We use WordNet Domains (WN Domains)[7], which is an extension of WN that assigns one or more domains to each WN synset. Using the algorithm proposed in [4], we compute the prevalent domains for the schemata. The algorithm examines all possible WN synsets connected to all the labels in the schemata and extracts all domains associated with these synsets. Next, it returns the top m prevalent domains[8].

We define the score of a long form candidate, $sc_{OD}(lf_i)$, as follows:

$$sc_{OD}(lf_i) = \sum_{d \in D(lf_i) \cap D(schemata)} \frac{p(< lf_i; d >)}{P_{schemata}}$$

where $D(schemata)$ is the list of prevalent domains (of WN Domains) associated with the schemata to integrate and $D(lf_i)$ is the list of domains associated by the online dictionary to the long form lf_i. $P_{schemata}$ is used as a normalizing constant, and is defined as the sum of the popularity of all the long form candidates for the short form sf:

$$P_{schemata} = \sum_{j} \sum_{d \in D(lf_i) \cap D(schemata)} p(< lf_j; d >)$$

The domain taxonomy used by Abbreviations.com is different from the one in WN Domains. To translate the dictionary domains of a long form

[6] The term "popularity" is used in the Abbreviations.com dictionary to denote the relevance of an expansion for a given abbreviation in a specific domain. Unfortunately, the online dictionary does not explain how the popularity value has been estimated (e.g. if it is based on the frequency of expanded forms in some domain-specific corpora).

[7] `http://wndomains.itc.it/`

[8] We use $m := 3$.

into WN Domains, we have manually defined 102 mappings between the two taxonomies.

For example, “Commerce”, “Sociology”, and “Metrology” are the prevalent domains for the schemata in Figure 1. For the abbreviation “CO” the online dictionary returns the following expansions: “Company” (Business), “Colorado” (Regional), “Check Out” (Medical). However, among these three entries only the first is relevant, because its category is mapped onto the “Commerce” domain of WN Domains (one of the schema domains); we obtain $sc_{OD}(Company) = 1$.

Step 3. Selecting the most appropriate long form
During this step, for each previously identified long form lf_i the algorithm computes a combined score $sc(lf_i) \in [0, 1]$. Then, the algorithm selects the top-scoring long form candidate. If the list of long form candidates is empty, the original short form is preserved. Hence, the score $sc(lf_i)$ is computed by combining scores from the individual resources:

$$sc(lf_i) = \alpha_{UD} \cdot sc_{UD}(lf_i) + \alpha_{CS} \cdot sc_{CS}(lf_i) + \alpha_{LC} \cdot sc_{LC}(lf_i) + \alpha_{OD} \cdot sc_{OD}(lf_i)$$

where $\alpha_{UD} + \alpha_{CS} + \alpha_{LC} + \alpha_{OD} = 1$ are weights of *resource relevance*. The resource relevance corresponds to degree to which a resource provides relevant expansions. As we have observed (see Section 5.1), particular expansion resources are more relevant for expanding ad hoc abbreviations as opposed to domain standard abbreviations. However, the syntax of an abbreviation does not provide any means for distinguishing between abbreviation types (standard or ad hoc), therefore, we are unable to choose in advance the expansion resource that is more relevant. Alternatively, we started by the following considerations: expansions provided by the user-defined dictionary best correspond to the user’s intention, as she/he can fully control the dictionary content; the context is the second relevant resource as it uses a vocabulary that closely reflects the intention of the schema designer who abbreviated the labels; finally, the complementary schema is less relevant than the context because it is (usually) designed by other schema designers, but it still uses vocabulary from the same, or a related, domain. Starting from this analysis, we selected the following as default weights of decreasing value[9]: $\alpha_{UD} = 0.4$,

[9] We verified our assumptions in preliminary, informal experiments over several data sources (different from the ones used in Section 7) and we found this to be the optimal configuration for evaluated schemata.

$\alpha_{LC} = 0.3$, $\alpha_{CS} = 0.2$ and $\alpha_{OD} = 0.1$. These weights can be further modified by the designer.

For example, the score for the long form candidate “Company” of the abbreviation “CO” becomes:

$$sc(Company) = 0.3 * 1 + 0.2 * 1 = 0.5$$

6. Compound noun interpretation

In the NLP (Natural Language Processing) literature different CN classifications have been proposed [14, 15]. In this work, we use the classification introduced in [14], where CNs are classified in four distinct categories: *endocentric*, *exocentric*, *copulative*, and *appositional.*

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* [16].

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. “bittersweet”, “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[10].

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"). We consider 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 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 2): (1) CN constituent disambiguation; (2) redundant constituent identification; (3) CN interpretation via semantic relationships; (4) creation of a new WN meaning for a CN.

Step 1. CN constituent disambiguation

In this step, the WN synset of each constituent is chosen in two moves:

1. *Compound Noun 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. We use the Stanford part of speech tagger [18] [11]. 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;
2. *Disambiguating head and modifier*: this step is part of the general lexical disambiguation problem. By applying our Combined Word

[10] 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 [17], available at `http://dit.unitn.it/~accord/Experimentaldesign.html`

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

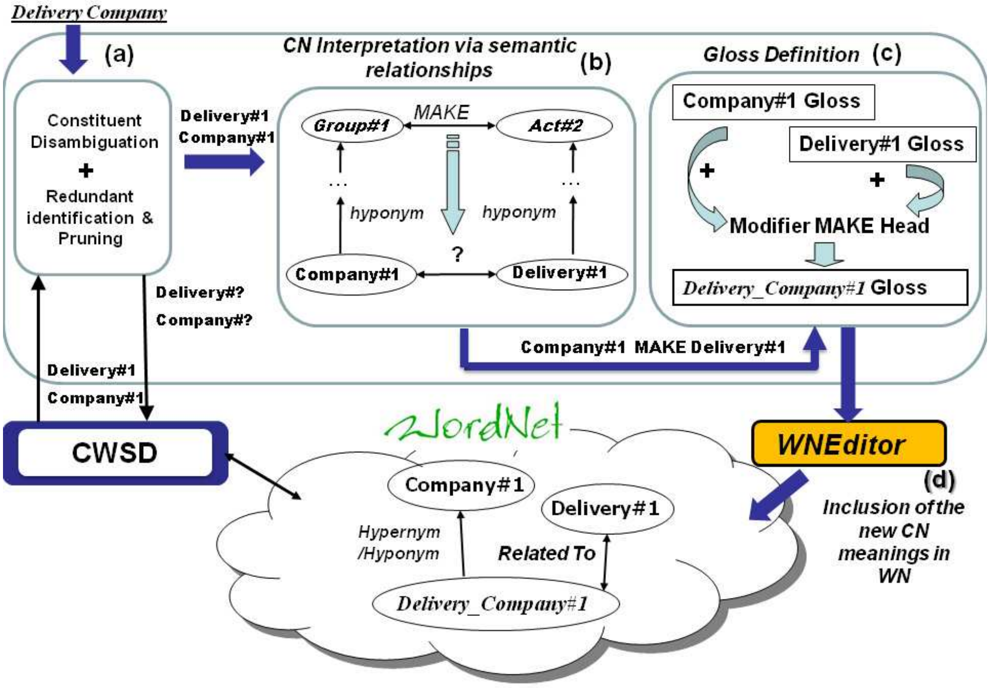


Figure 3: The CN interpretation process.

Sense Disambiguation (CWSD) algorithm [4] each word is automatically mapped onto its corresponding WN synset.

CWSD is an algorithm and a tool for the automatic annotation of structured and semi-structured data sources. Instead of being targeted to textual data sources like most of the traditional WSD algorithms, CWSD exploits the structure of data sources together with the lexical knowledge associated with schema elements to discover the right meaning to be associated with each word. CWSD is composed of two algorithms: SD (Structural Disambiguation) and WND (WordNet Domains Disambiguation). SD tries to disambiguate source terms by using semantic relationships inferred from the structure of data sources (intra-schema relationships) and WND tries to disambiguate the terms using domain information supplied by WN Domains.

We agree with [19] that WSD can significantly improve the accuracy of CN interpretation.

For example, as shown in Figure 3a, for the schema elements "DeliveryCO", previously expanded as "Delivery Company", we obtain the two constituents annotated with the correspondent WN meanings (i.e. "$Company_{\#1}$" and "$Delivery_{\#1}$"[12]).

Step 2. Redundant constituent identification and pruning
During this step, we control whether a CN constituent is a *redundant word.*
Definition 10.. *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 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 WN 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 [15] (shown in Table 3) In contrast, Finin claims an unlimited number of semantic relationships [20]. In [14] the problem of identifying a set of relationships is sidestepped: the semantics of a CN is then simply the assertion of an unspecified relationship between its constituents. Other sets of semantic

[12] #1 is a standard notation used in the WN literature to indicate the first WN meaning associated to a word; a similar way, the second WN 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 3: The Levi's set of semantic relationships (M= modifier, H= head).

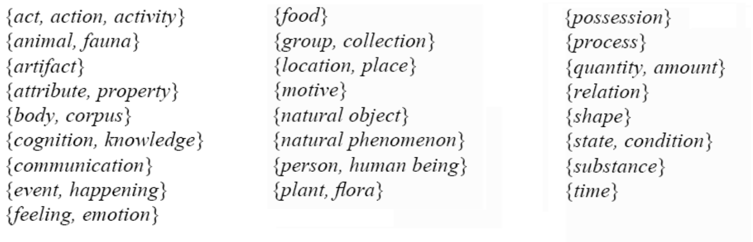
relationships to interpret CNs are proposed in [19, 21, 22].

The choice of the set of semantic relationships for CN interpretation has frequently been discussed in the NLP literature [23]: 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 [24].

In our method, we decided to use 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 [25], our 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 WN nouns in the WN 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 WN nouns hierarchy has been proven to be very useful in the CN interpretation task [23]. The top level concepts of the WN hierarchy are the 25 *unique beginners* (e.g. act, animal, artifact etc.) for WN English nouns defined by Miller in [3] (see Figure 4). In particular, in WN a unique beginner is a noun synset (i.e. a synset belonging to the noun syntactic category, thus belonging to the WN noun hierarchy) with no hypernymy synsets. These



{*act, action, activity*}
{*animal, fauna*}
{*artifact*}
{*attribute, property*}
{*body, corpus*}
{*cognition, knowledge*}
{*communication*}
{*event, happening*}
{*feeling, emotion*}
{*food*}
{*group, collection*}
{*location, place*}
{*motive*}
{*natural object*}
{*natural phenomenon*}
{*person, human being*}
{*plant, flora*}
{*possession*}
{*process*}
{*quantity, amount*}
{*relation*}
{*shape*}
{*state, condition*}
{*substance*}
{*time*}

Figure 4: The 25 unique beginners for the WN noun hierarchy.

unique beginners are related to other synsets through hyponym relationships (e.g. in Figure 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 [3]. As these unique beginners cover all noun synsets in WN, 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 WN.

Our 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 WN 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 3, 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 WN meaning creation (see below Step 4).

For each possible pair of unique beginners, we manually associate the relationship from Levi's set that best describes the meaning of the pair. For example, for the unique beginner pair "group and act" we chose Levi's relationship MAKE (e.g. "group MAKE act"), which can be expressed as "a group that performs an act". In this way, as shown in Figure 3b, we are able to interpret the label "Delivery Company" 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 WN has 25 unique beginners we associated a semantic relationship from Levi's set to 625 pairs of unique beginners[13]. This human intervention may be considered acceptable, when compared with the effort required by traditional approaches based on a pre-tagged corpora [21, 26], as will be discussed in Section 8. Moreover, our 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 WN meaning for a CN

During this step, we automatically create a new WN meaning for a CN starting from the meanings of its constituents and using the discovered relationship. We distinguish the following two steps:

1. *Gloss definition*: a WN *gloss* is the definition and explanation in natural language of the meaning of a term[14]. Starting from the relationship associated to a CN and exploiting the glosses of the CN constituents, we create 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, we chose to interpret CNs according to Levi's relationships, which can be used directly in the gloss. As shown in Figure 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 WN*: the insertion of a new CN meaning into the WN hierarchy implies the definition of its relationships with the other WN meanings. As the concept denoted by a CN is a subset of the concept denoted by the head, we assume that a CN inherits most of its semantics from its head [14]. Starting from this consideration, we can infer that the CN is related, in the WN hierarchy, to its head by a hyponym relationship. Moreover, we represent the CN semantics related to its modifier by inserting a generic relationship RT (*Related term*), corresponding to the WN relationships *member*

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

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

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, we also automatically control if other new CNs with the same head have been previously inserted in WN. For example, if we need to insert in WN 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 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 w.r.t. the other WN meanings. To this end, we use the WNEditor tool to create/manage the new meaning and to set relationships between it and the existing WN meanings [6]. The WNEditor automatically retrieves a list of candidate WN meanings sharing similarities with the new meaning. The designer is then asked to explicitly declare the type of relationship (e.g. hyponymy or meronymy[15]) to be established between the new meaning and the others, if any. Figure 3d illustrates this step with and example.

7. 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. To achieve these goals we conducted detailed experiments. The method was integrated within the MOMIS system. Schema label normalization is performed during the lexical annotation phase of MOMIS: in particular, during this phase, each schema element of a local source is semi-automatically annotated by the CWSD algorithm.

7.1. Experimental setup

We tested the effectiveness of our method in several real integration scenarios.

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

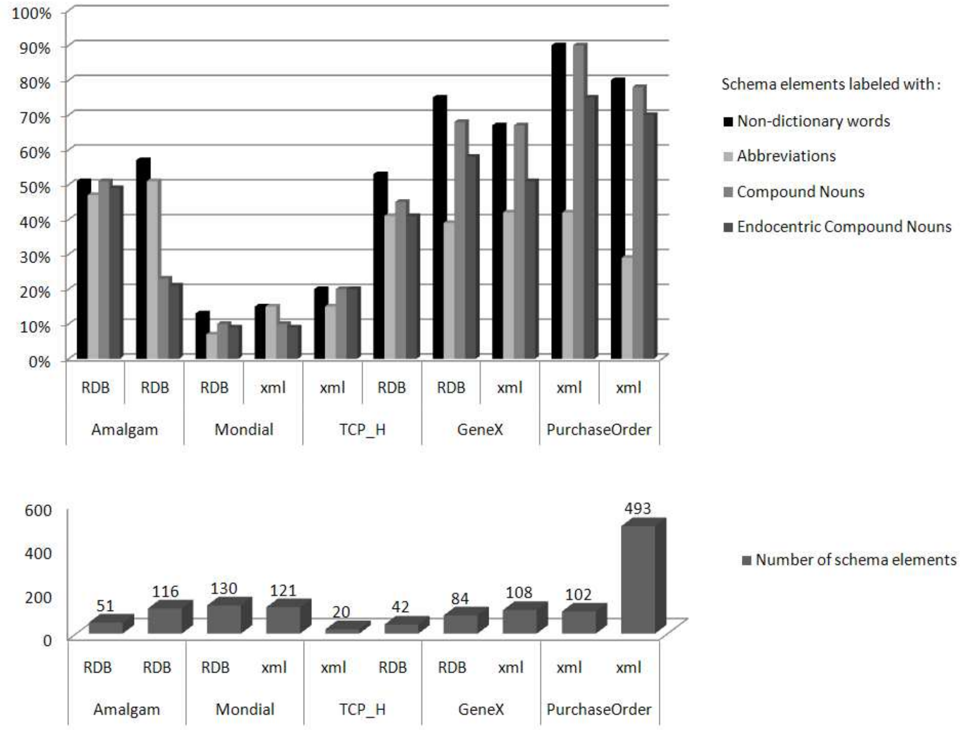


Figure 5: Feature summary of the data sets.

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 [27]), (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[16] have been used in several schema matching experiments [28, 29]. Figure 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).

[16] 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`

Experimental methodology. To assess the quality of our method, gold standards were created for each normalization phase as well as for the lexical annotation and the lexical relationship discovery process. The gold standards were manually generated by a human expert. The results obtained in each experiment were compared w.r.t. the corresponding gold standard.

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

Experimental Measures. To evaluate the performance of our method we used the quality measures defined in [30]. We compared the gold standards with the automatic results obtained by using our method. For each experimental phase we determined: the true positives, i.e. correct results (TP), as well as the false positives (FP) and the false negatives (FN). Based on the cardinalities of these sets, the following quality measures were computed:

- *Precision*= $\frac{|TP|}{|TP|+|FP|}$
- *Recall*= $\frac{|TP|}{|FN|+|TP|}$
- *F-Measure*= $2 * \frac{Precision*Recall}{Precision+Recall}$

Precision and recall originate from the field of information retrieval but have also been commonly used in schema matching and NLP studies.

7.2. 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.

7.2.1. 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 4).

Evaluating tokenization. We evaluated three tokenization methods: (1) *ST*—simple, (2) *GT/WN*—greedy with WN and (3) *GT/Ispell*—greedy with the Ispell English word list as a dictionary (see Section 4). We evaluated tokenization only for labels identified for normalization in the gold standard. The GT/WN shows the worst F-Measure (on average 64%), because WN

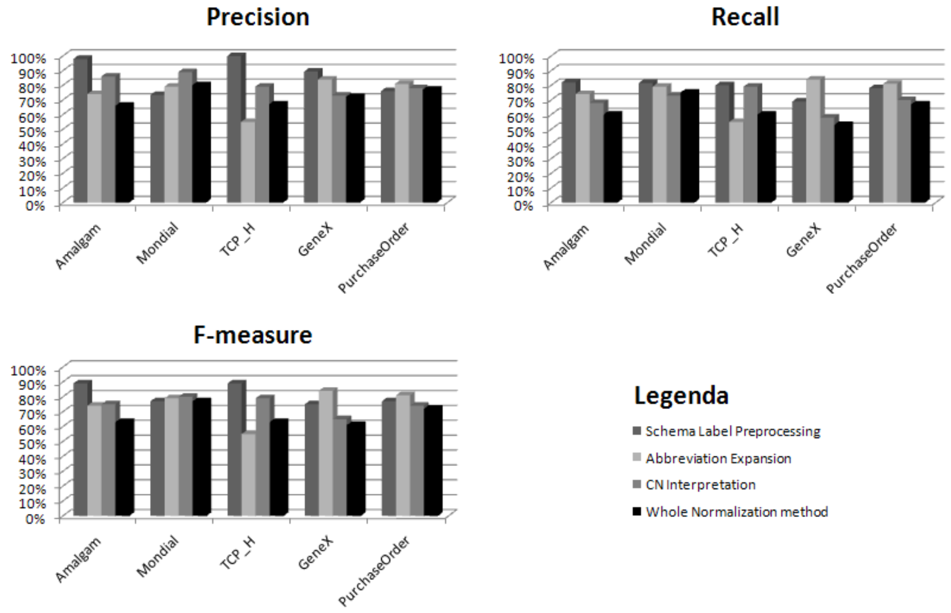


Figure 6: Performance of schema normalization for the different phases (schema label preprocessing (with GT/Ispell tokenization), abbreviation expansion, CN interpretation) and for the whole normalization method.

contains many short abbreviations (e.g. "auth" is tokenized to: "au", "th"). For the remaining two tokenization methods, the F-Measure was affected by the nature of the schema labels, but the GT/Ispell (F-Measure 86%) was on average 7% more accurate and more stable (6% standard deviation in F-Measure in contrast with 26% for the ST method).

Evaluating identification and classification. If a label is not identified for normalization, the whole normalization method might return incorrect expansions or unnecessary interpreted CNs. To estimate how relevant the problem of identification and classification is we tested those steps separately. Identification achieved 96% recall (averaged over all evaluated schemata), meaning that 4% of the labels with abbreviations were not recognized. This is because some abbreviations are ambiguous since they are also dictionary words in WN. 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 WN. 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 WN.

7.3. Evaluating abbreviation expansion

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 ($\alpha_{UD} = 0.4$, $\alpha_{LC} = 0.3$, $\alpha_{CS} = 0.2$, $\alpha_{OD} = 0.1$). 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 6. The algorithm provided correct expansions on average for 74% of the abbreviations. Evaluating the output of the algorithm, we found that 99% of the errors were caused by the lack of correct expansions from expansion resources, while only 1% of the errors were caused by incorrect selection (from among the long form candidates). This indicates that the default relevance weights lead to the selection of a relevant expansion in most cases.

Since most errors were caused by deficiencies in the quality of expansion resources, we investigated the contribution of each expansion resource to the final performance of the algorithm. We individually evaluated the F-Measure for the single expansion resources, for the internal resources (LC plus CS) and for the external resources (OD plus UD). Results are presented in Table 4. If we treat the user-defined dictionary as a baseline for our test, we observe that other resources are less correct in providing expansions, but their combination is a good strategy: it leads to significant improvement over the baseline.

It thus make sense to focus on the quality of each individual expansion resource. TCP-H data sets, with an F-Measure of 55%, revealed the poor quality of the chosen online abbreviation dictionary. For instance, the short form "mfgr" does not have any expansion in Abbreviations.com, but we found the correct expansion "manufacturer group" in AcronymFinder[17], an alternative dictionary. When we used the local context or the complementary

[17] `http://www.acronymfinder.com/`

UD	**OD**	**LC**	**CS**	**external**	**internal**	**all**
0.41	0.13	0.09	0.22	0.46	0.27	0.71

Table 4: F-Measure of the use of particular expansion resources in the abbreviation expansion algorithm (the value shown is the average value over all schemata).

schemata, the algorithm suggested words that are also abbreviations in the schemata as candidate expansions. This points to the need of improving the algorithm for the extraction of long form candidates. However, there are deficiencies on which a user integrating schema may not have influence. For instance, the PurchaseOrder data sets benefit from the complementary schemata resource much less (4%) then all other data sets (on average 22%). The results in Figure 6 also show that the F-Measure of a particular expansion resource ranges widely among schemata. The only general strategy to provide relevant expansions for a variety of schemata is thus to combine a diversity of expansion resources.

7.3.1. Evaluating CN interpretation

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 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 WN, 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

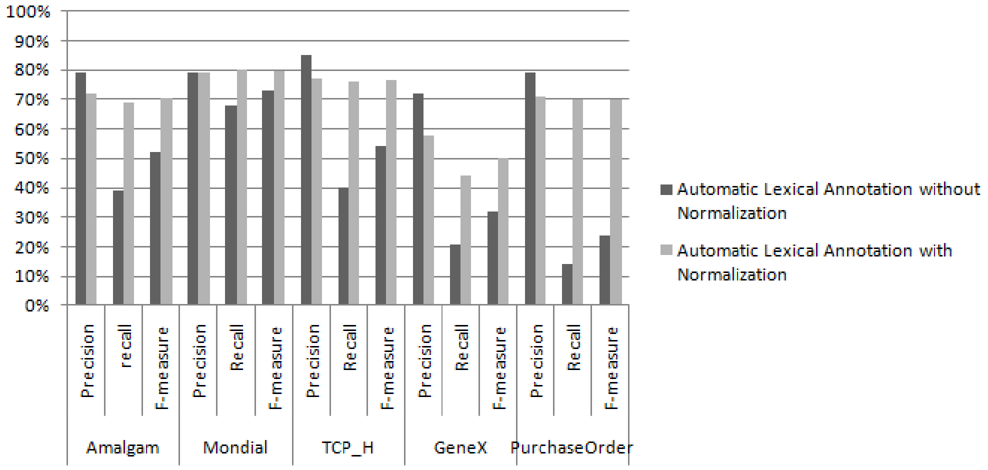


Figure 7: Lexical Annotation Evaluation.

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 6 - 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").

7.3.2. Evaluating the whole schema normalization method

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 results for the considered schemata. Figure 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%).

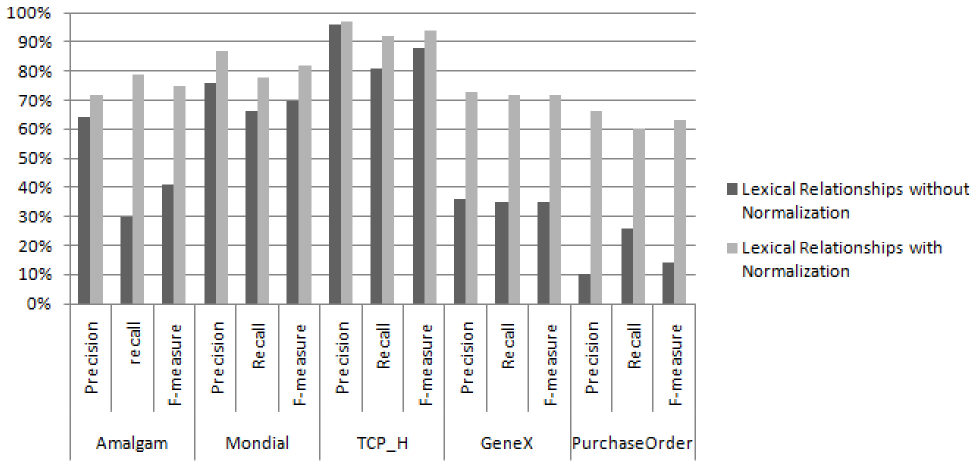


Figure 8: Lexical Relationship Evaluation.

7.4. Evaluating the impact of normalization on lexical annotation

The evaluation of the lexical annotation process was carried out by comparing the annotations returned by CWSD (starting from automatically normalized schemata) w.r.t. the gold standard. The gold standard was created by manually annotating each schema element w.r.t WN starting from manually normalized schemata. During the evaluation, a schema element annotation was considered a TP (i.e. *correctly annotated*) if the WN 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 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.

7.5. Evaluating the impact of normalization on lexical relationships

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 w.r.t. 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 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 WN 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.

8. Related work

Work related to the issues discussed in this article are in the area of linguistic normalization, normalization techniques in schema matching and finally the use of WN in schema matching.

8.1. 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, take advantage of a set of assumptions commonly verified in textual sources. For example, words in text documents have clearly defined word boundaries and thus the tokenization process is trivial. Some approaches suppose that abbreviations are words employing a specific syntax: for example, Taghva and Gilbreth [31] propose to consider only upper-case words of three to ten characters as acronyms. Other approaches assume that expansions often occur together in close proximity, i.e. in explicit position patterns, as for instance *“long form (short form)”* [32, 33]. 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. More specifically, schema labels do not have clearly defined word boundaries, thus making tokenization a more challenging task. Furthermore, it is not possible to classify words as abbreviations based on their syntax, because schemata often contain upper case labels regardless of whether they are short forms or long forms. Finally, alternative strategies need to be applied to find expansions, because their position in the schema is not explicitly given or even there might even be no corresponding expansions in the content of the schema (for instance, the expansion “Number” for the schema element "ItemNbr" in Figure 1 is present neither in the context schema nor in the complementary schema). Our solution addresses all those limitations.

When expanding abbreviations, potential expansions cannot always be found in a single source. [34] 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 this 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.

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 [21] 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 [26], 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 6).

There are three other main problems with corpus-based methods: (1) there has been 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 [25], 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.

8.2. Normalization techniques in schema matching systems

As 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 [35].

Surprisingly, current schema integration 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 [17] and COMA [28] schema matching systems overcome the problem of abbreviation by relying on the availability of a complete user-defined dictionary or a tool for abbreviation expansion.

Dealing with short forms using a user-defined dictionary only suffers from a lack 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 [36] 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 [11] 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 [37] 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 w.r.t. 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 interpretation in schema matching area. In [8] 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 [38] perform attribute matching by using WN 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 interpretation are manually executed.

The S-Match [2, 39] and CtxMatch [40] algorithms discover semantic matching by analyzing the meaning codified in the entities and the structures of ontolgies and by using WN as an external semantic source. CNs that are not present in WN are split into single words and their meaning is represented as the intersection of the single constituent meanings. In contrast with our method, they do not make distinctions between head and modifier nor enrich WN with the new CNs. H-Match [41], 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, and CN interpretation and does not enrich WN with new CNs.

Other schema and ontology matching tools do not interpret nor normalize CNs but they treat the constituents of a CN in isolation [8, 9, 42, 43]. This oversimplification leads to the discovery of false positive relationships, thus negatively affecting the matching results (as shown in the example in Section 2).

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 [5], 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). The main drawback of these approaches is that there are several situations where data instances are not available due to security reasons or restricted license authorizations [44].

8.3. The use of WN in schema matching systems

Semantic taxonomies and thesauri such as WN [3] are a key source of knowledge for NLP applications, and provide structured information about semantic relations between concepts [45]. We opted to use WN because it is the most commonly used English lexical thesaurus for the task of WSD [46]; however, our normalization method can be easily adapted to use other thesauri that provide a network of semantic relationships among meanings like WN does.

Potentially, all matchers that exploit WN or some other thesaurus to discover semantic relationships can integrate the techniques of abbreviation expansion and/or CN interpretation and thus refine the relationships involving non-dictionary words (e.g. some of these matchers are CtxMatch [40], S-Match [39], H-Match [41], OLA [47]).

For instance, CtxMatch uses a semantic matching approach that is a sequential composition of two techniques. At the element level it uses WN to find initial matching among classes (CtxMatch2 [48] improves on CtxMatch by handling ontology properties). At the structure 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. Our method can be apply even on a matching system that does not disambiguate the labels. As CtxMatch is not able to deal with abbreviations, it could take advantage of our method to find the best candidate long form for the nodes labeled with abbreviations. Then it can process the normalized label to compute the meaning of an element.

9. Conclusion and future work

In this article, we presented a method for the semi-automatic normalization of schema elements labeled with abbreviations and CNs in a data integration environment. Our method can be applied to several other contexts, including ontology merging, data-warehouses and web interface integration. The experimental results have shown the effectiveness of our method, which significantly improves the results of the automatic lexical annotation method, and, as a consequence, enhances the quality of the discovered inter-schema lexical relationships. Moreover, the effectiveness of our method becomes even more evident for larger schemata. We showed that, due to the frequency of non-dictionary words in schemata, a schema matching system cannot ignore CNs and abbreviations without compromising recall.

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 [49]; and (2) the problem of false negative non-dictionary words during the identification step (e.g. “RID” and “AID”). Future efforts will be also devoted to : the inclusion of other domain-specific resources to address the problem of the presence of specific domain term in schemata (e.g. the biomedical term “aromatase”, an enzyme involved in the production of estrogen); the use of multi-language thesauri in order to be able to normalize schemata in a language than different English; and finally, in the specific context of abbreviation expansion, we are also interested in the use of the information provided by data instances (when available) as discussed in Section 8.

10. Acknowledgments

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 (`http://www.dbgroup.`

unimo.it/keymantic) and by the MIUR FIRB Network Peer for Business project(http://www.dbgroup.unimo.it/nep4b).

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

[2] F. Giunchiglia, P. Shvaiko, M. Yatskevich, S-Match: an algorithm and an implementation of semantic matching, in: Y. Kalfoglou, W. M. Schorlemmer, A. P. Sheth, S. Staab, M. Uschold (Eds.), Semantic Interoperability and Integration, Vol. 04391 of Dagstuhl Seminar Proceedings, IBFI, Schloss Dagstuhl, Germany, 2005.

[3] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller, WordNet: An on-line lexical database, International Journal of Lexicography 3 (1990) 235–244.

[4] S. Bergamaschi, L. Po, S. Sorrentino, Automatic annotation for mapping discovery in data integration systems, in: S. Gaglio, I. Infantino, D. Saccà (Eds.), Proc. of the Sixteenth Italian Symposium on Advanced Database Systems, SEBD, 22-25 June 2008, Mondello, PA, Italy, pp. 334–341.

[5] E. Rahm, P. A. Bernstein, A survey of approaches to automatic schema matching, The Very Large Data Bases (VLDB) Journal 10 (4) (2001) 334–350.

[6] D.Beneventano, S. Bergamaschi, F. Guerra, M. Vincini, Synthesizing an Integrated Ontology, IEEE Internet Computing 7 (5) (2003) 42–51.

[7] J. Euzenat, P. Shvaiko, Ontology Matching, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[8] X. Su, J. A. Gulla, 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, 2004, pp. 217–228.

[9] J. Li, LOM: A Lexicon-based Ontology Mapping Tool, in: Proc. of Performance Metrics for Intelligent Systems, PerMIS'04, Gaithersburg, MD, August 24-26, 2004.

[10] H. Feild, D. Binkley, D. Lawrie, An Empirical Comparison of Techniques for Extracting Concept Abbreviations from Identifiers, in: Proc. of Software Engineering and Applications, SEA'06, November 2006, Dallas Texas.

[11] L. Ratinov, E. Gudes, Abbreviation expansion in schema matching and web integration, in: Proc. of the International Conference on Web Intelligence WI 2004, 20-24 September 2004, Beijing, China, IEEE Computer Society, pp. 485–489.

[12] R. Uthurusamy, L. G. Means, K. S. Godden, S. L. Lytinen, Extracting knowledge from diagnostic databases, The IEEE Expert Journal: Intelligent Systems and Their Applications 8 (6) (1993) 27–38.

[13] E. Hill, Z. P. Fry, H. Boyd, G. Sridhara, Y. Novikova, L. L. Pollock, K. Vijay-Shanker, AMAP: automatically mining abbreviation expansions in programs to enhance software maintenance tools, in: A. E. Hassan, M. Lanza, M. W. Godfrey (Eds.), Proc. of the International Working Conference on Mining Software Repositories, MSR 2008, Co-located with IEEE International Conference on Software Engineering, Leipzig, Germany, May 10-11, 2008, ACM, pp. 79–88.

[14] I. Plag, Word-Formation in English, Cambridge Textbooks in Linguistics, Cambridge University Press, New York, 2003.

[15] J. N. Levi, The Syntax and Semantics of Complex Nominals, Academic Press, New York, 1978.

[16] K. Barker, S. Szpakowicz, 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, August 10-14, 1998, Montreal, Quebec, Canada, pp. 96–102.

[17] J. Madhavan, P. A. Bernstein, E. Rahm, Generic Schema Matching with Cupid, in: P. M. G. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao, R. T. Snodgrass (Eds.), Proc. of the 27th International Conference on Very Large Data Bases, VLDB 2001, September 11-14, 2001, Roma, Italy, Morgan Kaufmann, pp. 49–58.

[18] K. Toutanova, C. D. Manning, 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, Association for Computational Linguistics, Morristown, NJ, USA, pp. 63–70.

[19] S. N. Kim, T. Baldwin, Automatic Interpretation of Noun Compounds Using WordNet Similarity, in: R. Dale, K.-F. Wong, J. Su, O. Y. Kwong (Eds.), Proc. of the Second International Joint Conference of Natural Language Processing, IJCNLP 2005, Jeju Island, Korea, October 11-13, 2005, Vol. 3651 of Lecture Notes in Computer Science, Springer, pp. 945–956.

[20] T. W. Finin, The Semantic Interpretation of Nominal Compounds, in: Proc. of the First National Conference of Artificial Intelligence, AAAI, 1980, pp. 310–312.

[21] D. Moldovan, A. Badulescu, M. Tatu, D. Antohe, R. Girju, 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, 2004, pp. 60–67.

[22] B. Rosario, M. Hearst, 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, 2001.

[23] V. Nastase, J. Sayyad-Shirabad, M. Sokolova, S. Szpakowicz, 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 2006, Boston, Massachusetts, USA.

[24] D. Ó Séaghdha, Learning compound noun semantics, Ph.D. thesis, Computer Laboratory, University of Cambridge, published as University of Cambridge Computer Laboratory Technical Report 735 (2008).

[25] J. Fan, K. Barker, B. W. Porter, The Knowledge Required to Interpret Noun Compounds, in: Proc. of the 18th International Joint Conference

on Artificial Intelligence, IJCAI, Acapulco, Mexico, August 9-15, 2003, pp. 1483–1485.

[26] T. B. Su Nam Kim, Standardised evaluation of english noun compound interpretation, in: Proc. of the International Conference on Language Resources and Evaluation, LREC, Workshop on Multiword Expressions MWEs, Marrakech, Morrocco, 2008, pp. 39–42.

[27] R. J. Miller, D. Fisla, M. Huang, D. Kalmuk, F. Ku, V. Lee, The Amalgam Schema and Data Integration Test Suite (2001).

[28] D. Aumueller, H. H. Do, S. Massmann, E. Rahm, Schema and ontology matching with COMA++, in: Proc. of Special Interest Group on Managenegment of Data, SIGMOD'05, New York, NY, USA, ACM, 2005, pp. 906–908.

[29] L. Chiticariu, P. G. Kolaitis, L. Popa, Interactive generation of integrated schemas, in: J. T.-L. Wang (Ed.), Proc. of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008, ACM, 2008, pp. 833–846.

[30] H. H. Do, S. Melnik, E. Rahm, Comparison of Schema Matching Evaluations, in: A. B. Chaudhri, M. Jeckle, E. Rahm, R. Unland (Eds.), Web, Web-Services, and Database Systems, NODe 2002 Web and Database-Related Workshops, Erfurt, Germany, October 7-10, Vol. 2593 of Lecture Notes in Computer Science, Springer, 2002, pp. 221–237.

[31] K. Taghva, J. Gilbreth, Recognizing acronyms and their definitions, International Journal on Document Analysis and Recognition, IJDAR 1 (4) (1999) 191–198.

[32] S. Yeates, D. Bainbridge, I. H. Witten, Using Compression to Identify Acronyms in Text, in: Proc. of the Data Compression Conference, DCC'00, IEEE Computer Society, Washington, DC, USA, 2000.

[33] J. T. Chang, H. Schtze, R. B. Altman, Creating an online dictionary of abbreviations from MEDLINE, Journal of the American Medical Information Association 9 (6) (2002) 612–620.

[34] W. Wong, W. Liu, M. Bennamoun, 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, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 2006, pp. 83–89.

[35] H. H. Do, Schema Matching and Mapping-based Data Integration: Architecture, Approaches and Evaluation, Vdm Verlag Dr. Müller, 2006.

[36] S. Melnik, H. Garcia-Molina, E. Rahm, Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching., in: Proc. of the 18th International Conference on Data Engineering, ICDE, February 26- March 1 2002, San Jose, CA, 2002, pp. 117–128.

[37] X. Chai, M. Sayyadian, A. Doan, A. Rosenthal, L. Seligman, Analyzing and revising data integration schemas to improve their matchability, The Proceedings of the Very Large Database Endowment, PVLDB 1 (1) (2008) 773–784.

[38] D. W. Embley, D. Jackman, L. Xu, 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, 2001, pp. 110–117.

[39] P. Shvaiko, F. Giunchiglia, M. Yatskevich, Semantic matching with S-Match, Semantic Web Information Management: a Model-Based Perspective XX (2010) 183–202.

[40] P. Bouquet, L. Serafini, S. Zanobini, Semantic coordination: A new approach and an application, in: D. Fensel, K. P. Sycara, J. Mylopoulos (Eds.), International Semantic Web Conference, Vol. 2870 of Lecture Notes in Computer Science, Springer, 2003, pp. 130–145.

[41] S. Castano, A. Ferrara, S. Montanelli, Matching ontologies in open networked systems: Techniques and applications, Journal of Data Semantics V (2006) 25–63.

[42] B. T. Le, R. Dieng-Kuntz, F. Gandon, 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 2004, Porto - Portugal, pp. 236–243.

[43] J. Euzenat, D. Loup, M. Touzani, P. Valtchev, 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, November 2004, Hiroshima, Japan.

[44] C. Clifton, E. Housman, A. Rosenthal, 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, 1997.

[45] F. Lin, K. Sandkuhl, A Survey of Exploiting WordNet in Ontology Matching, in: M. Bramer (Ed.), Artificial Intelligence in Theory and Practice II, Vol. 276 of IFIP, International Federation for Information Processing, Springer Boston, 2008, pp. 341–350.

[46] R. Navigli, Word sense disambiguation: A survey, ACM Computing Surveys 41 (2).

[47] J. Euzenat, P. Valtchev, Similarity-based ontology alignment in owl-lite, in: R. L. de Mántaras, L. Saitta (Eds.), Proc. of the 16th Eureopean Conference on Artificial Intelligence, ECAI 2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004, IOS Press, 2004, pp. 333–337.

[48] P. Bouquet, L. Serafini, S. Zanobini, S. Sceffer, Bootstrapping semantics on the web: meaning elicitation from schemas, in: L. Carr, D. D. Roure, A. Iyengar, C. A. Goble, M. Dahlin (Eds.), Proc. of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006, ACM, 2006, pp. 505–512.

[49] E. C. Dragut, F. Fang, A. P. Sistla, C. T. Yu, W. Meng, Stop word and related problems in web interface integration, The Proceedings of the Very Large Database Endowment, PVLDB 2 (1) (2009) 349–360.