Abbreviation Expansion in Lexical Annotation of Schema

Maciej Gawinecki*

University of Modena and Reggio Emilia,
Maciej.Gawinecki@unimore.it

Abstract. Lexical annotation of schema elements can improve effectiveness of schema matching. However, it cannot be applied is those schema elements that contain abbreviations. In this work we address this problem by providing a new technique for abbreviation expansion in the context of schema of structured and semi-structured data.

1 Introduction

The aim of data integration systems is creation of global schema successfully integrating schemata from different structured and semi-structured data sources [1]. This process requires understanding of the meaning standing behind the names of schema elements. In this situation lexical annotation helps explicate these meanings by labeling schema elements with concepts from a lexical resource [2]. The lexical resource provide an agreement on the meaning and intended use of terms, making possible to match together different terms, but with the same or similar meaning. Unfortunately, it may not include *abbreviations*, while the effectiveness of annotation process heavily suffers from presence of such words in the schema [2].

Current schema integration and annotation systems either do not consider the problem of abbreviation expansion at all or they use non-scalable solution of a user-defined dictionary. In this paper we propose an algorithm for *automated abbreviation expansion*. Abbreviation expansion is an approach of finding a relevant expansion for a given abbreviation. Our contributions are as follows:

- We provide a method for expanding abbreviations using complementary sources of expansion candidates. Different sources of expansions are complementary to each other because they provide expansions for different types of abbreviations.
- We provide evaluation of effectiveness of each source separately and combinations of them to present pros and cons of each source.

Our method is implemented in MOMIS (Mediator envirOnment for Multiple Information Sources) system [1] where lexical annotation is done with respect to WordNet (WN) dictionary [3]. However, it may be applied in general in the

* PhD student of the ICT Doctorate School (University of Modena and Reggio-Emilia)

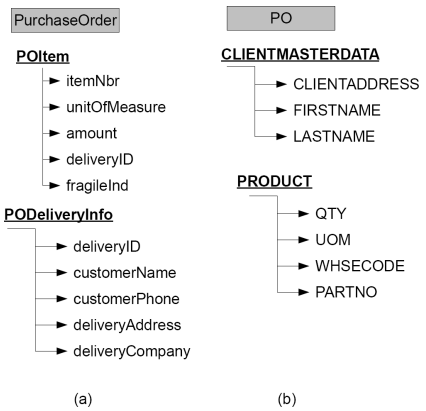

Fig. 1. Graph representation of two schemata with elements containing abbreviations: (a) relational database schema, (b) XML schema.

context of schema mapping discovery, ontology merging and data integration system. It may be also adapted for semantic annotations in general, i.e. annotations performed with respect to other shared models (industry standards, vocabularies, taxonomies, and ontologies).

The paper is organized as follows. In Section 2 we define the problem of abbreviation expansion occuring in schema labels. Section 3 describes the proposed algorithm for expanding abbreviations in schema elements names. Section 4 presents the current state of the art in the field of abbreviation expansion. Finally in Section 4 we provide evaluation of the proposed solution and we conclude with proposals of future work in Section 5.

2 Problem definition

Element names represent an important source of assessing similarity between schema elements. This can be done semantically by comparison of their meanings.

Definition 1 *Lexical annotation of a schema label is the explicit assignment of its meaning w.r.t. a lexical resource (a thesaurus).*

Definition 2 *An abbreviation (short form) is a shortened form of a word or phrase (long form), that consists of one or more letters taken from the long form.*

Figure 1 presents two schemata to be integrated, containing many labels with non-dictionary abbreviations, e.g. 'PO' (standing for "Purchase Order"), QTY ("Quantity"). They cannot be directly annotated, because they do not have an entry in WN, while their corresponding expansions may be easily recognized by WN. Hence, it is necessary to *identify* and *expand* all abbreviations appearing in schema labels before performing lexical annotation.

Definition 3 *Abbreviation identification is the task of determining whether a given word has been used for abbreviation in the given context.*

Very often *legitimate* English words are used for abbreviations in the schema context[1]. For instance, 'id' is a dictionary word in WN standing, among many others, for "primitive instincts and energies underlying all psychic activity", while in the prevalent number of analyzed schemata it is a short form of 'identifier' ("a symbol that establishes the identity of the one bearing it").

Definition 4 *Abbreviation expansion is the task of finding a relevant expansion for a given identified abbreviation.*

There may be several possible long forms for a given short form. For instance, *Abbreviations.com* online dictionary provides 56 different expansions for abbreviation 'PO', including: "Post Office", "Purchase Order", "Parents Of" and others. Therefore, automatic abbreviation expansion can be split into two sub-problems: (a) searching for potential long forms (expansions) for a given short form; and (b) selecting the most appropriate long form from the set of long form candidates.

3 Proposed solution for automatic abbreviation identification and expansion

Dealing with abbreviations appearing in a schema label involves two operations: (1) identifying whether it is a short form or it contains short forms and then (2) providing relevant long forms for identified short forms. These operations should be performed for each schema label as it has been described on Figure 2. In the following subsections we describe how each operation is realized.

3.1 Abbreviation identification

We consider a word to be an abbreviation if it belongs to the list of *standard abbreviations* or it is not a dictionary word. The list of well-known standard abbreviations is employed here to reduce the number of false negatives caused by legitimate English words used for abbreviations.

Note, that non-dictionary labels can consist of more then one word. We are tokezing them using one the pre-existing approaches [4]: *simple* – based on camel case and punctuation, and *greedy* – handling also multi-word names without clearly defined word boundaries, e.g. 'WHSECODE'. The latter iteratively looks for the biggest prefixing/suffixing dictionary words and user-defined abbreviations in non-dictionary words.

[1] Please also note that, besides abbreviations a schema may contain other non-dictionary terms such as: multi-word terms (including compound nouns), misspellings, numbers – '3D' and foreign language words that might affect the process of identifying abbreviations, but we are not dealing with them in this approach.

```
for each schema S iterate over classes and their direct attributes
  for the label l of each class/attribute in S:
    if l is a standard abbreviation then
      lf_k := selectLongForm(l)
    else if l is not a dictionary word then
      tokenize l into words (w_i)_i
      for each word w_i in (w_i)_i
        if w_i is a standard abbreviation or not a dictionary word then
          lf_k := selectLongForm(w_i)
        end if
      end for
    end if
  end for
end for
```

Fig. 2. Proposed solution.

Example 1 *Let us assume the algorithm tries to identify abbreviations in two selected labels of the schemata presented in Figure 1: 'PODeliveryInfo' and 'WHSECODE'. Simple tokenization works only for the first label and it returns: 'PO','Delivery' and 'Info' words. Greedy tokenization helps to tokenize the latter label, isolating: 'WHSE' and 'CODE' (the longest dictionary word in the label). 'PO' and 'WHSE' are later identified as abbreviations.*

3.2 Observations on abbreviation expansion

A schema can contain both *standard* and *ad hoc* abbreviations. However, only the standard abbreviations can be found in user-defined and online abbreviation dictionaries as they: either (a) denote important and repeating domain concepts or (b) are standard suffix/prefix words used to describe how a value of a given schema element is represented[2]. For instance 'Ind' (Indicator) defines a list of exactly two mutually exclusive Boolean values that express the only possible states of a schema element, like in 'FragileInd' on Figure 1. On the contrary, ad hoc abbreviations are mainly created to save space, from phrases that would not be abbreviated in a normal context [5, 6].

To observe how specifically ad hoc abbreviations can be handled automatically we analyzed short forms and their corresponding long forms in several open-source schemata. Based on our manual inspection, we found two sources relevant for finding possible long form candidates:

- *context* of short form occurrence, as it is common practice to prefix column an attribute name with a short form of a class name, for instance 'recentchanges' table contains 'rc_user' and 'rc_params', while the 'FIBranchID' attribute is an element of 'FinancialInstitutionType' complex type.

[2] e.g. *OTA XML Schema Design Best Practices*, http://www.opentravel.org.

INPUT: sf – short form occurrence
OUTPUT: lf – long form for sf
compute the list $L_{UD} := (< lf_{UD}, 1 >)$, where lf_{UD} is a matching long form in UD
compute the list $L_{CS} := (< lf_{CS}, 1 >)$, where lf_{CS} is a matching long form in CS
compute the list $L_C := (< lf_C, 1 >)$, where lf_C is a matching long form in C of sf
compute the list $L_{OD} := (< lf_{OD,i}, score_{OD}(lf_{OD,i}) >)_i$, where $lf_{OD,i}$ is
a matching long form in OD
$L = L_{UD} \cup L_{CS} \cup L_C \cup L_{OD}$ // combine long forms scores
$lf := \arg\max_{lf_i \in L} score(lf_i)$

Fig. 3. Proposed procedure for selecting a long form for the given short form.

– a *complementary schema* which we would integrate with inspected schema; for instance when integrating two schemata from Figure 1, we found that the XML schema contains abbreviations ('PO', 'uom'), which may be expanded with long forms from relational database schema ('Purchase Order', 'unit Of Measure').

3.3 Proposed algorithm for abbreviation expansion

To handle well-known standard abbreviations the algorithm uses an online abbreviation dictionary (`OD`) and user-defined dictionary (`UD`, with abbreviations typical for schema labels). The ad hoc abbreviations are expanded using context (`C`) and complementary schema (`CS`) as sources of long form candidates. The syntax of a short form itself does not provide any mean for distinguishing between ad hoc and standard abbreviations. Therefore, we are not able to choose in advance these sources which are more relevant for expansion of a certain short form. However, the context and complementary schema can be generally considered as the most relevant sources of long form candidates, because they closely reflect the intention of a schema designer.

For each identified abbreviation the algorithm inquires all four sources for long form candidates, scores candidates according to the relevance of the source, combines scores of repeating long forms and chooses the top-scored one. The whole process is shown in Figure 3.

Technically, for each identified short form sf the algorithm generates a list of long form candidates: $(< lf_i; score(lf_i) >)_i$ obtained from all the sources. The list is sorted descendingly according to the $score(lf_i) \in [0, 1]$ of a long form candidate lf_i. The algorithm selects the top-scored long form candidate from the list. If the list is empty, then the original short form is preserved. The score of lf_i is computed by combining scores from the single sources:

$$\begin{aligned} score(lf_i) = \alpha_{UD} \cdot score_{UD}(lf_i) + \alpha_{CS} \cdot score_{CS}(lf_i) \\ + \alpha_C \cdot score_C(lf_i) + \alpha_{OD} \cdot score_{OD}(lf_i) \end{aligned}$$

where $\alpha_{UD}+\alpha_{CS}+\alpha_C+\alpha_{OD} = 1$ are weights corresponding to different relevance of the sources.

To define a context let us suppose sf_i be a short form identified in a label l. The label l is either: (a) an attribute of a class c or (b) a class belonging to schemata s. Then the context of sf_i is the class c or schema s. The context is retrieved for possible long form candidates using the four abbreviation patterns proposed in [7]. The abbreviations patterns are, in practice, regular expression created from characters of a short form.

The labels in the schema complementary to the schema in which sf appears are retrieved for matching long form candidates using the same abbreviation patterns as in the context. Only the first matching candidate is considered.

For a user-defined dictionary, a context and a complementary schema sources the score of lf_i is 1, if the lf_i is found in the given source or 0 – otherwise. For an online dictionary we describe scoring procedure below.

Example 2 *When expanding the first abbreviation in 'PODeliveryInfo' element the algorithm receives the following information from the particular sources:*

source	***weight***	lf_i	***score***
online dictionary	$\alpha_{OD} = 0.10$	*'Purchase Order'*	0.33
online dictionary	$\alpha_{OD} = 0.10$	*'Parents Of'*	0.28
context	$\alpha_C = 0.30$	*'Purchase Order'*	1.0
complementary schema	$\alpha_C = 0.20$	*'Purchase Order'*	1.0

The context of 'PODeliveryInfo' is in this case the name of its schema, while 'PO' is a complementary schema. The user-defined dictionary does not return any long form candidate. Next, the algorithm merges lists of proposed candidates into a single one: [Purchase Order (0.53), Parents Of (0.028)]. Particularly, the score for the top-scored expansion has been computed in the following way: $0.53 \approx 0.10 \cdot 0.33 + 0.30 \cdot 1.0 + 0.20 \cdot 1.0$.

Scoring long form candidates from an online dictionary is more complex, as the dictionary may suggest more then one long form for a given short form. For this purpose we propose disambiguation technique based on two factors: (a) the number of domains a given long form shares with both schemata and (b) its popularity in these domains. We assume information about the domain of a long form and its popularity is given by the online dictionary.

Domain-based disambiguation is a popular approach for the problem of word-sense disambiguation [8, 2]. The intuition behind this approach is that only meanings of a word that belongs to the same domains that both schemata describe, are relevant. Please note that in our adaptation of this method we are selecting not a meaning, but a long form from a set of long form candidates. Practically, we may define score of long form candidate – $score_{OD}(lf_i)$ – as follows:

$$score_{OD}(lf_i) = \frac{1}{P_{schema}} \sum_{d \in CD(lf_i, schemata)} p(lf_i, d)$$

$$P_{schema} = \sum_i \sum_{d \in CD(lf_i, schemata)} p(lf_i, d),$$

$$CD(lf_i, schemata) = D(lf_i) \cap D(schemata)$$

where $D(schemata)$ is a list of prevalent WN Domains[3] associated with schemata to integrate (obtained using the algorithm from [2]). Computation of $CD(lf_i, schemata)$ — the intersection of prevalent domains and domains associated with long form lf_i — involves the mapping between the categorization system of an online abbreviation dictionary and WN Domains classification. If there is no shared domain for any long form candidate, then score is computed as a general popularity of a long form candidate.

There can be more than one online dictionary entry describing the same long form lf_i, but in different domains. Therefore, the entry can be modeled as a combination of a long form lf_i and a domain $d_{i,k} \in D(lf_i)$ in which it appears with the associated popularity. Formally, we define the t-th dictionary entry in the following form: $< e_t, p(e_t) >$, where $e_t =< lf_i; d_{i,k}) >$ and $d_{i,k} \in D(lf_i)$ is the k-th domain in the set of domains $(D(lf_i))$, in which the long form lf_i appears. The popularity $p(e_t)$ is not explicitly reported by the considered dictionary but can be easily estimated from the order of descending popularity in respect to which entries are returned by the dictionary. Thus we are able to calculate $p(e_t)$ using the following induction: $p(e_{t+1}) = p(e_t)/\kappa$, $p(e_1) = 1.0$, where $\kappa > 1$ is an experimentally defined factor[4].

Example 3 *The prevalent domains for the schemata in Figure 1 are the following: 'commerce', 'sociology' and 'metrology'. Let us assume we are scoring long form candidates from the online dictionary for a short form 'PO', for which the dictionary returns the following three corresponding entries:*

t	i	lf_i	*dict. categories*	$p(lf_i, d)$	$\{d_{i,k}\}_k$
1	1	*Purchase Order*	*Accounting*	1.0	{*economy, commerce, book_keeping*}
2	2	*Parents Of*	*Law*	$0.83 \approx 1/(1.2)$	{*sociology, law*}
3	1	*Purchase Order*	*Military*	$0.69 \approx 1/(1.2)^2$	{*military*}

The $score_{OD}$(*'Purchase Order'*) $= 1.0/3 \approx 0.33$*, where the third entry is not taken into account because it does not share any domain with the schemata;* $score_{OD}$(*'Parents Of'*) $= 0.83/3 \approx 0.28$ *as it shares the 'sociology' domain.*

4 Related work

The problem of abbreviation expansion has received much attention in different areas such as: machine translation, information extraction, information retrieval and software code maintanance.

Many techniques are based on the observation that in documents the short forms and their long forms usually occur together in patterns [9, 10]. As there are no such explicit patterns in data schemata, we adopted abbreviations patterns used for the similar problem of expanding abbreviated variable identifiers in program source codes [7]. Selecting the most relevant long form from a single source

[3] http://wndomains.itc.it/wordnetdomains.html

[4] In experiments we successfully use $\kappa := 1.2$

is made with respect to different factors such as: *inverted frequency* of a long form in both domain-specific and general corpora [11], size of *document scope* in which both a short form and a matching long form appear [7] or *syntactic similarity* between a long form and a short form (e.g. whether a short form has been created by removing internal vowels from the given long form) [6]. In our approach both context and complementary schema are returning only the first discovered match.

It has been observed that the presence of abbreviations in schema elements labels may affect the quality of *elements name matching* and requires additional techniques to deal with [12]. Surprisingly, current schema integration systems either does not consider the problem of abbreviation expansion at all or solve it in non-scalable way by inclusion of a simple user-defined abbreviation dictionary (e.g. Cupid [13], COMA/COMA++ [14]). Lack of scalability comes from the fact that: (a) the vocabulary evolves over the time and it is necessary to maintain the table of abbreviations and (b) the same abbreviations can have different expansions depending on the domain. Moreover, this approach still requires an intervention of a schema/domain expert.

According to our knowledge, the work of Ratinov and Gudes [5] is the only one that attacks the problem of abbreviations in the context of data integration. Their technique focuses on: (a) supervised learning of abbreviation patterns to deal with ad hoc abbreviations and (b) usage of external corpus as a source of matching long forms. However, the authors do not report any details nor evaluation of possible disambiguation of candidates. We claim that ad hoc abbreviations in schemata can be handled using abbreviation patterns proposed in [7] and context and complementary schema as more relevant sources of long forms.

5 Evaluation

We implemented our method for abbreviation identification and expansion in the MOMIS system [1]. The implementation uses the following external information: (a) WordNet 3.0, (b) a list and a dictionary of standard abbreviations, and (c) Abbreviations.com online abbreviation dictionary. For domain-based disambiguation of long forms we created a mapping between the category hierarchy of this dictionary and WordNet Domains[5]. We tested the performance of our method over the two relational schemata of the well known Amalgam integration benchmark for bibliographic data [16]. Table 1 summarizes the test schemata features that are particularly suitable for the test. We acknowledge that one of the schemata contains four elements with unknown meaning and thus not considered in the experiments. We performed our evaluation with the following questions in mind: (a) What is the effectiveness of abbreviation identification method?; (b) What is the effectiveness of each source in providing correct long forms?; (c) How effective are *external* sources (user-defined dictionary and

[5] [15] describes criteria and our survey of selection online abbreviation dictionary and procedure of creation of such a mapping. All mentioned information is accessible at: http://www.ibspan.waw.pl/~gawinec/abbr/abbr.html.

Number of	Labels	Non-dictionary words	Abbreviations
Schema 1	117	66	24
Schema 2	51	28	28

Table 1. Characteristics of test schemata.

online abbreviation dictionary) in comparison to *internal* sources (context and complementary schema) in dealing with different types of abbreviations (ad hoc and standard)?; (d) How can we assign relevance weights to each source to limit their individual weaknesses and take advantage of their strengths when combining them together?

To answer to these questions we performed two groups of experiments. One group for evaluation of abbreviation identification method (question 1) and one for abbreviations expansion method (questions 2-4). We evaluated these methods separately, because errors produced during the first one (incorrectly tokenized labels and identified abbreviations) gives a different input for the second method and thus may impact its effectiveness.

5.1 Evaluating abbreviation identification

In this group of experiments we measured the correctness of abbreviation identification. We consider a label correctly identified if — in respect to manual identification: (a) the label as a whole has been correctly identified, and (b) the label has been correctly tokenized, and (c) all abbreviations in the label have been identified. During manual identification multi-word abbreviations (abbreviations shortening more then one word) were tokenized into abbreviations representing single words (e.g. 'PID' standing for "Publication Identifier" was tokenized into 'P' and 'ID'), with exceptions to standard abbreviations (such as 'ISBN'), that were left untokenized.

For automated tokenization we use two competing methods: *simple* and *greedy* (see Section 3.1 for details). Therefore, we evaluated identification method in three variants depending on the tokenization method used: (1) `ST`: simple, (2) `GT/WN`: greedy with WN and (3) `GT/Ispell`: greedy with Ispell English words list[6] as a dictionary.

The `ST` reaches nearly the same correctness (0.92) as `GT/Ispell` (0.93), because the schemata contain relatively few labels with unclearly undefined word boundaries (e.g. 'bktitle'). On the contrary, the `GT/WN` remains far away from its competitors (0.70), because WN contains many short abbreviations (e.g. 'auth' is tokenized to: 'au', 'th'). All three variants of the method have been affected by the usage of legitimate English words for abbreviations, that were not defined on a list of standard abbreviations.

5.2 Evaluating abbreviation expansion

In this group of experiments we measured the correctness of abbreviations expansion method. We performed 7 experiments: 4 measuring correctness of each single source of long forms, 2 two experiments for evaluation of external and internal sources and 1 evaluating correctness of all sources combined together. An identified abbreviation was considered to be correctly expanded with respect to the manual expansion. The input for each experiment was manually tokenized and identified. The results of experiments are shown on Figure 4.

It can be observed that *internal* sources of long forms (context and complementary schema) provide correct long forms complementary to *external* sources (user-defined dictionary and online abbreviation dictionary). The user-defined dictionary provided correct expansions in 42% of abbreviations, but it still does not handle with ad hoc abbreviations such as 'Auth' ('Author'), 'coll' ('collection') or 'bk' ('book'), where internal sources behave better. Finally, online dictionary provided 19% of correct results, including well-known abbreviations such as 'ISBN' ("International Standard Serial Number") and 'ISSN' ("International Standard Serial Number"). The relevant long forms were chosen among many others provided by the online dictionary, because they share the highest number of domains with both schemata, namely: telecommunication and publishing.

When combining external sources together we considered user-defined dictionary (UD) as a more relevant source then online abbreviation dictionary (OD), because it describes expansions more typical for schemata. When combining internal sources we gave more importance to context (C) source over complementary schema (CS), because context source reflects better user intention about a given schema. Finally, when combining external and internal sources, we found that: (1) complementary schema may provide less relevant long forms then user-defined dictionary, (2) online dictionary is considered the last chance source of long forms, when no other source provide relevant long forms. Following this guidelines we experimentally setup the final weights for all the sources: $\alpha_{UD} := 0.40$, $\alpha_{C} := 0.30$, $\alpha_{CS} := 0.20$, $\alpha_{OD} := 0.10$. For such weights the whole abbreviation expansion provided 83% correct expansions for manually tokenized and identified input.

6 Conclusion and Future Work

Abbreviations appearing in schema labels are serious obstacle for correct lexical annotation of schema elements. Surprisingly, current data integration systems either ignores the problem of abbreviations or handle it with a usage of non-scalable user-defined dictionary. To overcome this problem we presented a method for identifying and expanding abbreviations appearing in schema element names. We proposed a usage of four sources of expansions: context of abbreviation occurrence in schema, a schema complementary to the annotated one, an online abbreviation dictionary and user-defined dictionary of standard

[6] Ispell is a popular tool for spelling errors correction: http://wordlist.sourceforge.net/.

abbreviations. We have experimentally shown that these sources are complementary in recognizing particular types of abbreviations and their combination can provide high correctness for abbreviation identification and expansion.

Currently we are working on improving the correctness of abbreviation identification method for words without clearly defined word boundaries.

Acknowledgements: The author would like to thank for valuable comments to Sonia Bergamaschi and Serena Sorrentino.

References

1. Bergamaschi, S., Castano, S., Vincini, M.: Semantic integration of semistructured and structured data sources. SIGMOD Rec. **28**(1) (1999) 54–59
2. Bergamaschi, S., Po, L., Sorrentino, S.: Automatic annotation for mapping discovery in data integration systems. In: SEBD 2008. (2008) 334–341
3. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Wordnet: An on-line lexical database. International Journal of Lexicography **3** (1990) 235–244
4. Feild, H., Binkley, D., Lawrie, D.: An Empirical Comparison of Techniques for Extracting Concept Abbreviations from Identifiers. In: SEA'06. (November 2006)
5. Ratinov, L., Gudes, E.: Abbreviation Expansion in Schema Matching and Web Integration. In: WI '04. (2004) 485–489
6. Uthurusamy, R., Means, L.G., Godden, K.S., Lytinen, S.L.: Extracting knowledge from diagnostic databases. IEEE Expert: Intelligent Systems and Their Applications **8**(6) (1993) 27–38
7. Hill, E., Fry, Z.P., Boyd, H., Sridhara, G., Novikova, Y., Pollock, L., Vijay-Shanker, K.: AMAP: automatically mining abbreviation expansions in programs to enhance software maintenance tools. In: MSR '08. (2008) 79–88
8. Magnini, B., Strapparava, C., Pezzulo, G., Gliozzo, A.: The role of domain information in word sense disambiguation. Nat. Lang. Eng. **8**(4) (2002) 359–373
9. Yeates, S., Bainbridge, D., Witten, I.H.: Using Compression to Identify Acronyms in Text. In: DCC'00. (2000)
10. Chang, J.T., Schtze, H., Altman, R.B.: Creating an online dictionary of abbreviations from MEDLINE. Journal of the American Medical Information Association **9**(6) (2002) 612–620
11. Wong, W., Liu, W., Bennamoun, M.: Integrated scoring for spelling error correction, abbreviation expansion and case restoration in dirty text. In: AusDM '06. (2006) 83–89
12. Do, H.H.: Schema Matching and Mapping-based Data Integration: Architecture, Approaches and Evaluation. VDM Verlag (2007)
13. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: VLDB. (2001) 49–58
14. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: SIGMOD'05. (2005) 906–908
15. Gawinecki, M.: On Selecting Online Abbreviation Dictionary. Technical Report XXX, University of Modena and Reggio-Emilia (2009)
16. Miller, R.J., Fisla, D., Huang, M., Kymlicka, D., Ku, F., Lee, V.: The Amalgam Schema and Data Integration Test Suite. www.cs.toronto.edu/ miller/amalgam (2001)

	user-def. dict.	online dict.	compl. schema	context	external sources	internal sources	all sources
Correct expansions	20	9	6	7	29	12	40
in %	42%	19%	13%	15%	60%	25%	83%
Label.short form							
AID.A	N	N	N	Y	N	Y	Y
AuthID.Auth	N	N	Y	Y	N	Y	Y
bktitle.bk	N	N	Y	N	N	Y	Y
collID.coll	N	N	N	Y	N	Y	Y
ConfInfo.Conf	N	N	N	N	N	N	N
CountryPub.Pub	N	Y	N	N	Y	N	N
DID.D	N	N	N	Y	N	Y	Y
inst.inst	N	N	N	N	N	N	N
ISBN.ISBN	N	Y	N	N	Y	N	Y
ISSN.ISSN	N	Y	N	N	Y	N	Y
JournalAnn.Ann	N	N	N	N	N	N	N
LID.L	N	N	N	Y	N	Y	Y
loc.loc	N	N	Y	N	N	Y	Y
PID.P	N	N	N	Y	N	Y	Y
RID.R	N	N	N	Y	N	Y	Y
techreport.tech	N	Y	N	N	Y	N	Y
TitleExt.Ext	N	Y	N	N	Y	N	Y
vol.vol	N	Y	N	N	Y	N	Y
AbstractInd.Ind	Y	N	N	N	Y	N	Y
AccessionNum.Num	Y	N	N	N	Y	N	Y
AID.ID	Y	N	N	N	Y	N	Y
articleID.ID	Y	N	N	N	Y	N	Y
AuthID.ID	Y	N	N	N	Y	N	Y
bookID.ID	Y	N	N	N	Y	N	Y
collID.ID	Y	N	N	N	Y	N	Y
ConfInfo.Info	Y	N	N	N	Y	N	Y
ContractNum.Num	Y	N	N	N	Y	N	Y
DID.ID	Y	N	N	N	Y	N	Y
LID.ID	Y	N	N	N	Y	N	Y
num.num	Y	N	N	N	Y	N	Y
NumRef.Num	Y	N	N	N	Y	N	Y
NumRef.Ref	Y	N	N	N	Y	N	Y
PID.ID	Y	N	N	N	Y	N	Y
RID.ID	Y	N	N	N	Y	N	Y
techID.ID	Y	N	N	N	Y	N	Y

Fig. 4. Evaluation of abbreviation expansion for: (a) each long form sources separately, (b) *internal* sources (context and complementary schema) and *external* sources (user-defined dictionary and online abbreviation dictionary) and (c) all sources combined together. The letters in cells stands for: Y – expanded correctly; N – expanded incorrectly.