Taming Data Explosion in Probabilistic Information Integration

Ander de Keijzer, Maurice van Keulen, and Yiping Li

Faculty of EEMCS, University of Twente
POBox 217, 7500AE Enschede, The Netherlands
{a.dekeijzer,m.vankeulen,liy}@ewi.utwente.nl

Abstract. Data integration has been a challenging problem for decades. In an ambient environment, where many autonomous devices have their own information sources and network connectivity is ad hoc and peer-to-peer, it even becomes a serious bottleneck. To enable devices to exchange information without the need for interaction with a user at data integration time and without the need for extensive semantic annotations, a probabilistic approach seems rather promising. It simply teaches the device how to cope with the uncertainty occurring during data integration. Unfortunately, without any kind of world knowledge, almost everything becomes uncertain, hence maintaining all possibilities produces huge integrated information sources. In this paper, we claim that only very simple and generic rules are enough world knowledge to drastically reduce the amount of uncertainty, hence to tame the data explosion to a manageable size.

1 Introduction

Data integration is a difficult problem. The number of available information sources is ever growing. Furthermore, the information sources themselves are becoming larger and larger. Information integration is often supported with tools, but largely remains a labor-intensive manual task. Fully automatic and accurate information integration is still not possible. At best, tools assist users with suggestions of matching data items or attributes, or with performing schema and data conversions based on given rules, etc. The need for world knowledge is nicely illustrated by the data integration challenges given by [Lev99]:

- overlapping and contradictory data,
- semantic mismatches among sources, and
- different naming conventions for data values

Concrete decisions are required to deal with these challenges and they need to be exact to determine unambiguously the number of resulting data items and exact values of all data items in an integration information source. Even techniques based on machine learning cannot make such decisions with certainty. See, for example, the two address books of Figure 2. In the integration of these two simple data sources, a decision is needed whether or not “John” and “Jon” are

the same person for knowing whether the resulting address book contains two or three persons.

In an ambient environment, where many autonomous devices have their own information sources and network connectivity is ad hoc and peer-to-peer, information integration becomes a serious bottleneck. Devices need to be able to exchange and integrate information whenever the opportunity arises and without human interaction. Therefore, we approach information integration rather differently: any decision to be made that needs world knowledge, is not resolved at information integration time. Instead, our integration approach stores the uncertainty about this decision and associates a probability to each possible outcome of the decision. The probability indicates to what level of certainty, the data element should be in the integrated information source. Using this approach we can express that multiple appearances of a data item are possible. When in doubt, the integration approach does not throw away information by attempting to choose between the various possibilities, but instead acknowledges its uncertainty by storing all possibilities.

Unfortunately, without any kind of world knowledge, almost everything becomes uncertain, hence maintaining all possibilities produces huge integrated information sources. This is due to the fact that many things, however remotely possible, are indeed *in principle* possible. In [vKdKA05], we calculated that for two information sources with each five data items, there are in theory 1546 possibilities how these may combine.

In this paper, we show that the size of the integrated result can be substantially reduced, just by adding some very simple knowledge rules about the real world to our system.

The remainder of this paper is organised as follows. First, we position our work among related research. Second, we briefly give a summary of our probabilistic XML datamodel and integration approach [vKdKA05]. We subsequently examine in Section 4 a practical scenario where several information sources with information on movies are integrated. Next, in Section 5 we introduce simple knowledge rules to be used during integration and attempt to quantify the effect of adding knowledge rules in Section 6. We conclude in Section 7, where we also indicate directions for future research.

2 Related Work

There is a large body of related work on information integration. [DH05] provides a nice survey. It is useful to distinguish schema matching and integration from data integration. In this paper, we focus on data integration, hence we presume that schema matching and integration has already been done. [RB01] is a good survey on schema matching techniques.

The challenges for data integration given earlier [Lev99] have received much attention in recent years. In our research, we attempt to deal with these challenges by explicitly handling the inherent uncertainties occurring in the data integration process using a probabilistic database approach. Suciu's tutorial at

SIGMOD 2005 comes with an extensive bibliography on the topic of probabilistic data management [SD05]. Many results from the logic programming and artificial intelligence communities are combined in [ELLS01] which proposes a probabilistic object database. The object-oriented data model is more expressive, but also less flexible, than the XML data model. Nevertheless, many things carry over to the XML world.

A probabilistic database is not a new idea, see for example [FKL97], but in recent years attention grew considerably. Originally, work concentrated on relational databases, but in [vKdKA05] we argue that XML can be made to express uncertainty in a more natural way. Other probabilistic XML databases are, for example, PXML [HGS03] and ProTDB [NJ02].

Although schema and data level matching and integration can be clearly separated, schema matching techniques (see [MBR01] for a nice taxonomy) can often be used or adapted to be applicable on data level. For example, [BN05] presents a technique to search for duplicate records and to use these duplicates for schema matching. As we will see in the sequel, an important problem in data integration is how to decide whether or not two data items refer to the same real-world object. Duplicate finding techniques can be applied to (partly) solve this problem. Also in many other areas duplicate finding techniques can be found, such as data warehousing [ACG02].

A problem in using probabilistic databases for data integration is how to determine the probabilities. Many schema matching techniques suitable for data integration, however, quantify the degree of matching. For example, instance-based matchers use classification techniques [DDH01]. If two data items from different information sources referring to the same real-world object conflict on some attribute value, and one of those values is classified with less certainty that the other in the class corresponding to the attribute, then that attribute value is less likely to be correct and should receive a smaller probability. The same holds for techniques where dictionaries or thesauri are used: if a possible data value is not present in the corresponding dictionary, it should receive a smaller probability. Schema matching techniques can also be used for data conversion as [MZ98] demonstrated.

Finally, an important source of schema and data integration techniques can be drawn from the Semantic Web community. As we argued in the introduction, world knowledge is required for making decisions in the integration process. In theory, annotating the data with sufficient world knowledge may also overcome the problem. The question remains if it is practical to demand from all information sources to be sufficiently annotated to resolve all uncertainty. Furthermore, it is an open problem how to determine beforehand when annotations suffice to resolve all uncertainty. We, therefore, approach the problem from the other end. The probabilistic data integration approach as such is independent of any world knowledge. Adding world knowledge can then be used to restrict uncertainty. In this paper, we claim that only very simple and generic rules are enough world knowledge to drastically reduce the amount of uncertainty, hence to tame the data explosion to a manageable size.

3 Probabilistic XML Information Integration

The key concepts necessary for information integration are summarized in this section.

In an ordinary XML document, all information is certain. When two XML information sources are integrated, however, they may conflict on information about certain real world objects. For example, when integrating two address books, one may claim that a person's telephone number is '1111' and the other claims it is '2222'. Therefore, after data integration, there may exist more than one possibility for a certain text node, or in general, for entire subtrees. We model this uncertainty in a probabilistic XML tree by introducing two special kinds of nodes:

1. probability nodes depicted as $\bigtriangledown$, and
2. possibility nodes depicted as $\circ$, which have an associated probability.

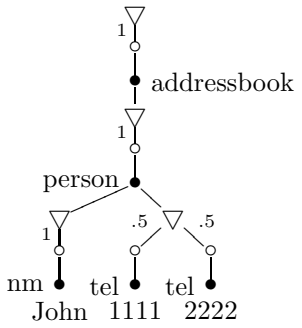

Fig. 1. Example probabilistic XML tree.

Ordinary XML nodes are depicted as $\bullet$. The children of a probability node enumerate all possibilities for that node. Figure 1 shows a probabilistic XML tree illustrating uncertainty about the telephone number of a person named 'John'. A probabilistic XML tree is well-structured, if the children of all probability nodes are possibility nodes, their combined probabilities add up to 1, the children of all possibility nodes are XML nodes, and the children of all XML nodes are probability nodes. In this way, on each level of the tree, you only find one kind of nodes.

A probabilistic XML tree can be seen as a device's knowledge about the 'real world'. The probabilistic XML tree of Figure 1 says that in the real world, there exists a person named "John" with telephone number "1111", or there exists a person named "John" with telephone number "2222". These are called *possible worlds*. The answer to a query on a probabilistic XML tree can be determined by executing the query on each possible world separately.

In [vKdKA05], we formally defined all these concepts and notions.

3.1 Probabilistic Integration

Uncertainty in data may arise when integrating two information sources. There are many possible causes for this uncertainty, for example, it is uncertain whether or not two elements refer to the same real-world object, or conflicts between the information sources are detected and it is uncertain which source is correct. In this section, we show how information sources can be integrated in such circumstances without knowledge of the real world.

In Figure 2 two information sources are shown. Both information sources contain address information on two persons (address information other than the persons' names is not shown). In principle, it is unknown if the persons to whom

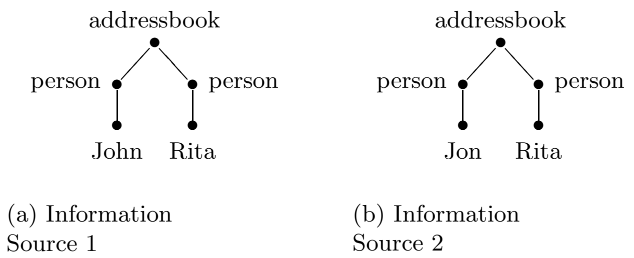

Fig. 2. Two address books

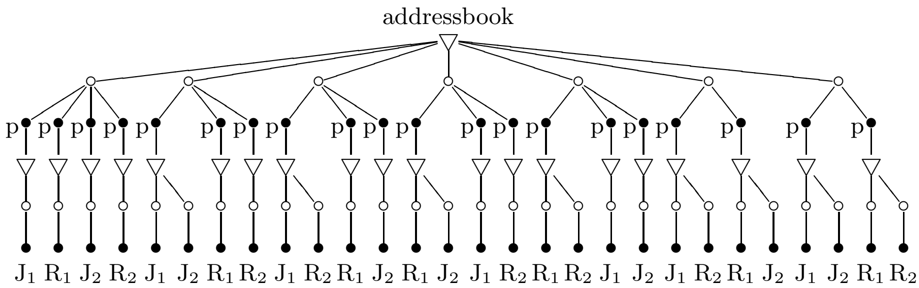

Fig. 3. Integration result of address books of figure 2

is referred are the same in both sources. It perfectly possible that somebody knows two different people who are both named “Rita”. Note furthermore, that it is equally well possible that “John” and “Jon” both refer to the same person, but that in one of the information sources, his name is mispelled. Without any more world knowledge, we simply do not know, so anything is possble.

Let us first assume no other information about the real world is available, which means that any element from the first address book can be matched with any element from the second address book. Note that this means that John from source 1 can refer to the same real-world object as Rita from source 2, because there is a remote possibility that a huge typing mistake has occurred, or that John changed his name to Rita. We do assume, however, that two elements from the same address book never refer to the same real world object. In other words, all elements within one address book refer to unique real world objects.

Figure 3 shows the result after integrating the two address books given in Figure 2. For readability, we left out the probabilities per possible world and only showed an outline of the document, omitting all text. The total number of different possible situations (*possible worlds* in our terminology [dKvK04,vKdKA05]) that can be derived after integration is 7. There is 1 possible world with 4 per-

Person 1	Person 2	Person 3	Person 4
$John_1$	$Rita_1$	Jon_2	$Rita_2$
$John_1$ / Jon_2	$Rita_1$	$Rita_2$	
$John_1$ / $Rita_2$	$Rita_1$	Jon_2	
$Rita_1$ / Jon_2	$John_1$	$Rita_2$	
$Rita_1$ / $Rita_2$	$John_1$	Jon_2	
$John_1$ / $Rita_2$	Jon_2 / $Rita_1$		
$John_1$ / Jon_2	$Rita_1$ / $Rita_2$		

Table 1. Possible worlds after integration

Source	#movies	Description of information offered
Internet Movie Database [IMDb]	470,000	movie details, plot summary, cast, other people involved, genre, goofs, quotes, trivia, user comments, awards, user rating, etc.
All Movie Guide [AMG]	290,000	movie details, plot summary, cast, other people involved, genre, keywords, themes, moods, etc.
Yahoo! movies [Yahoo]	*unknown*	movie details, plot summary, cast, other people involved, genre, user comments, photos, critics reviews, etc.
Simply Scripts [SS]	1,500	title, directors, transcript

Table 2. Some movie sources

sons, 4 possible worlds with 3 persons and 2 possible worlds with 2 persons. The possible worlds are summarized in Table 1. In this table each row represents a possible world and each column represents one person. The notation A/B indicates that A and B refer to the same person in the real world. It is then often uncertain if he/she is represented in the information source by A or by B. Subscripts in the table indicate from which information source, either 1 or 2, the element originates from.

The list of child elements of person in this example is limited to the name for simplicity. If we include more child elements, possible conflicts between data items grows. As a result, the number of possible worlds may grow exponentially.

4 Movie database scenario

Without world knowledge, integrated information sources can become very large. As explained, this is due to the fact that many things, however remotely possible, are *in principle* possible. The example of Section 3 shows that for the integration of two data sources with each two data items, there are already seven possible worlds. In [vKdKA05], we calculated that for two data sources with each five data items each carrying four children, there are in theory 1546 possible worlds.

But this is theory and these are small examples. To be able to get a feel for the size of the problem *in practice*, we investigate a scenario in which we attempt to integrate a number of data sources on the web containing movie information. Table 2 shows several interesting data sources that may be used. There are many movie data sources with similar kinds of information about many movies, such as IMDb, All Movie Guide and Yahoo! movies. But there are also data sources that have very specific data about only a limited number of movies, such as Simply Scripts offering transcripts of what is said in a movie. Integrating the information of such data sources may have much added value.

The main cause for explosion in the number of possible worlds is the *semantic equality problem*: How to decide whether or not two data items refer to the same real-world object? Without world knowledge, any movie data item may in theory be semantically equal to any other movie data item in another data source. We saw that in the previous section where 'John' and 'Rita' may in principle refer to the same person in reality, however unlikely this may be. When integrating sources with hundreds of thousands of movies, the number of possibilities are enormous, most importantly too many to handle.

Some features of a data item, however, are often known to be keys or key-like, hence can be used to determine whether or not two data items refer to the same real-world object. For movies, we found for example many data sources including the IMDb number. This number can be used to determine with absolute certainty that two movie data items actually refer to the same movie.[1] Unfortunately, not all movie data sources include the IMDb number, but they do include the title, an attribute that is also almost always correct and very discriminative. But it is not a key: there are, for example, three movies called "King Kong", namely the 1933, 1976, and 2005 versions. Together with the year attribute, which is almost always also included, we do have a good alternative key.

Other main causes for explosion in the number of possible worlds are differences in which attributes are included and actual attribute values that do not correspond. To get a feel for this problem, Table 3 investigates in more detail the data given for the 2005 "King kong" movie by the first three data sources of Table 2. What we can observe is the following:

- The 'Title/Year' information can indeed be used to exactly match the corresponding items in all three sources.
- The 'Genre'-attribute contains more differences than correspondences. In general, other movies show more correspondences, but they almost never completely agree. This is due to the subjective nature of the attribute and the usage of different terms. Assuming different strings indeed represent different genres, our information integration approach will result in a list of terms for this attribute whereby for each term it is uncertain whether or not the term is actually a member of the list. 'Adventure' will be the only

[1] There is an important assumption here not to be neglected: This assumes that the data is correct. If not, we may erroneously decide that two data items are the same. Furthermore, it may also happen that two data items that do refer to the same movie, are not detected as such, resulting in duplication of information.

Attribute	Comparison
Title/Year	Exactly equal in all three sources.
Genre	IMDb gives 'Action', 'Adventure', 'Drama', 'Fantasy', 'Sci-Fi', and 'Thriller'. AMG gives 'Adventure', 'Monster Film', and 'Period Film', hence only one in common. AMG has other genre-like attributes like keywords, themes, tones, and moods, but these do not overlap with any IMDb genre. Yahoo gives 'Action/Adventure', 'Romance', 'Thriller', and 'Remake'.
Cast	IMDb presents a cast of 15 people, AMG 11, and Yahoo 13. All provide both the names of the actors as well as whom they play in the movie. The 11 actors of AMG are all present in both IMDb and Yahoo. The 2 extra actors of Yahoo are different from the 4 extra of IMDb. Furthermore, there are three differences in spelling.
Location	IMDb has a 'Country'-attribute with value "New Zealand / USA". AMG has 'Filming location' with value "New Zealand". Yahoo has 'Filming Locations' with value "Wellington, New Zealand (Campertown Studios - Stone Street Studios)".
Plot summary	All three sources have a different description or plot summary.

Table 3. Comparison of information on the 2005 movie "King Kong".

one about which certainty exists (provided that the integrator separates the combined "Action/Adventure" genre description of Yahoo).

- The 'Cast'-attribute is technically speaking also a simple list of strings. The difference with 'Genre' is that it is much more factual data, hence much less differences can be observed. Even with factual data, however, we observe that the three sources do not fully agree on 'the cast' of the movie. There are 11 names that belong to the cast with certainty, but there are also 6 more names that are given by only one of the three sources.
- The 'Location'-attribute also concerns factual data. Although all sources agree that the filming location is "New Zealand", the actual string values are far from the same. Integrating this attribute with the approach of Section 3.1 results in three possibilities for the attribute. The uncertainty, however, is local. A query asking for movies filmed in New Zealand contains a predicate like `contains(location,'New Zealand')` or even `location='New Zealand'`. In our probabilistic XML approach, such a query *will find the movie* "King Kong" although the query answer of the latter will have a lower probability assigned to this movie [vKdKA05].
- Attributes like 'Plot summary' completely differ for the three sources. In principle, all information sources are correct: All descriptions are valid descriptions of the movie. Similar to 'Location', our integration approach treats this as three local possibilities, which does not have significant negative effects on querying.
- Finally, if we were to also integrate Simply Scripts, we observe that it is possible to find the corresponding movies using Title/Year. Probabilistic XML

trees can represent uncertainty about the existence of a subtree. Integration with Simply Scripts would result in local uncertainty about whether or not an attribute 'transcript' exists in the real world.

We can draw several conclusions from the analysis of the movie scenario. First of all, due to the existence of keys or key-like attributes, the explosion of possibilities resulting from the uncertainty about semantical equality of data items from different sources can be greatly reduced. The only uncertainty remaining is local for an attribute. With the compact representation of a probabilistic XML tree, the storage requirements for these local possibilities is not expected to be large [vKdKA05]. Furthermore, querying the resulting integrated data collection is not expected to suffer significantly from the incurred uncertainty. Items can still be found, some items may only have a reduced probability of being correct.

Most importantly, we expect that with some simple world knowledge statements such as 'Title/Year is a key', the number of possible worlds can be greatly reduced to a manageable size. Further world knowledge can even resolve local uncertainties, for example with statements like 'non-existence of a transcript is not a conflict, simply take it if available', genres and names in a cast analogously.

In Section 5, we present results of some experiments that attempt to quantify the effectiveness in uncertainty reduction of very simple and generic rules.

5 Simple Knowledge Rules

Initially [vKdKA05], we did not use any world knowledge when integrating information sources. As a result, the number of possibilities in the resulting information source was huge. The size of this result can be reduced drastically, just by using very simple rules about the real world.

We defined 3 kinds of knowledge rules. The first category contains the *Numeric rules*. An object is considered a description of the same real world object if some threshold is met. The second category of knowledge rules are domain specific rules. In the case of our address book example, we consider names to be keys. Data items are descriptions of the same real world object, if the name of the person is equal. The last category combines rules from the first two categories. These rules are called *Combination rules*.

We will describe the rules in both of the categories below. We start with the numeric rules.

- Single Element Rule
  This rule considers object descriptions to refer to the same real world object, if *one or more* of the elements in both sources have the same value.
- 50% Rule
  This rule considers object descriptions to refer to the same real world object, if *at least 50%* of the elements in both sources have the same value.

In our experiments, *person* elements have both a firstname and lastname childnode, hence possible name rules are as follows.

– Firstname rule
  This rule considers data items to refer to the same real world object, if the firstname of both data items is equal.
– Lastname rule
  This rule considers data items to refer to the same real world object, if the lastname of both objects is equal.

Finally, the combination rules are

– Combination rule 1
  This rule combines the 50% rule and the Firstname rule.
– Combination rule 2
  This rule combines the 50% rule and the Lastname rule.
– Combination rule 3
  This rule combines the Firstname rule and the Lastname rule and is therefore also referred to as the *Fullname rule*.

6 Experiments and Evaluation

In our experiments, we used documents with the following DTD.

```
<!DOCTYPE persons [
<!ELEMENT persons (person*) >
<!ELEMENT person (firstname, lastname, phone, room)>
<!ELEMENT firstname (#PCDATA) >
<!ELEMENT lastname (#PCDATA) >
<!ELEMENT phone (#PCDATA) >
<!ELEMENT room (#PCDATA) >
]>
```

To keep the examples in this paper readable, we only show the name attribute of a person. Whenever the other elements have an effect on the integration process or result, we mention them specifically.

In our experiments we used the two address book documents given in Figure 4. Using the integration method without knowledge rules, the number of possible worlds in the result document is 1815. This explosion of possible worlds is caused by the fact that every element from the first source can possibly refer to the same real-world object as any element from the second source. For example, there is a remote possibility that even Mark Hamburg and Allen Kingship refer to the same person in reality..

We performed 7 experiments: one for every knowledge rule.

The simplest of the knowledge rules, the single element rule, already reduced the number of possible worlds to 39. This is a reduction of almost 98%, while the actual knowledge introduced is minimal: if two data items do not agree on any attribute, we decide that they do not refer to the same real-world object.

The 50% rule reduced the number of possible worlds to just 15, as do the firstname rule and combination rule 1.

The best result is achieved by using the Lastname rule, combination rule 1, or combination rule 3. These rules reduce the number of possible worlds to only 3. Combination rule 2 can be compared with combination rule 1, in the

```
<persons>
<person>
<firstname>Mark</firstname>
<lastname>Hamburg</lastname>
<phone>1010</phone>
<room>3300</room>
</person>
<person>
<firstname>Allen</firstname>
<lastname>King</lastname>
<phone>2020</phone>
<room>3122</room>
</person>
<person>
<firstname>Stan</firstname>
<lastname>Choice</lastname>
<phone>3030</phone>
<room>3035</room>
</person>
<person>
<firstname>John</firstname>
<lastname>Friend</lastname>
<phone>4040</phone>
<room>3333</room>
</person>
</persons>
```

(a) Document 1 (660 bytes)

```
<persons>
<person>
<firstname>Mark</firstname>
<lastname>Hamburg</lastname>
<phone>1010</phone>
<room>3301</room>
</person>
<person>
<firstname>Allen</firstname>
<lastname>Kingship</lastname>
<phone>2020</phone>
<room>3035</room>
</person>
</persons>
```

(b) Document 2 (366 bytes)

Fig. 4. Address book documents used in experiments

sense that a special emphasis is placed on one of the elements, in this case the lastname element.

We should, however, avoid adding world knowledge that does not hold in general. For example, if document 1 would have had the data item 'John Kingship / 4030 / 3035', it is actually very likely that this data item does *not* refer to the same real world object as 'Allen Kingship / 2020 / 3035'. The 50% rule is in this case not a good knowledge rule, because it rules out possibilities that are likely to be true. Good knowledge rules are those that have little or no false positives.

7 Conclusion and future work

In this paper we have shown that the data explosion in probabilistic information integration can be reduced drastically by introducing some very simple knowledge rules. Rules like the 50% rule "data items refer to the same real-world

object, if at least 50% of the attributes correspond" reduced the number of possible worlds from 1815 to only 39, a reduction of almost 98%.

In the movie database scenario, we looked at some real-life data to be able to investigate the uncertainty occurring in practical information integration. We showed that although much conflicting information can be found, there is enough solid ground to be able to define simple rules with. For example, considering movies with a different name or year as different movies is enough to deal with the semantic equality problem. It is expected that the remaining uncertainty need not be resolved to be able to effectively answer the usual queries.

Although probabilistic information integration can function without user interaction at integration time, user interaction may still be beneficial. The user could indicate if certain possibilities are nonsense, i.e. not referring to a possible state of affairs in the real world. In such a case, those possibilities can be eliminated from the information source. As future research, we will investigate if user statements about a query result can be used to reduce uncertainty in the information source.

Conceptually our query approach is to pose the user query to each possible world and afterwards combine the possible answers. This method becomes increasingly inefficient when the number of possible worlds increase. Allowing queries to be posed on the compact representation can improve the efficiency.

References

[ACG02] Rohit Ananthakrishna, Surajit Chaudhuri, and Venkatesh Ganti. Eliminating fuzzy duplicates in data warehouses. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002), August 20-23, 2002, Hong Kong, China*, pages 586–597, 2002.

[BN05] Alexander Bilke and Felix Naumann. Schema matching using duplicates. In *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, pages 69–80. IEEE Computer Society, 2005.

[DDH01] A.H. Doan, P. Domingos, and A. Halevy. Reconciling schemas of disparate data sources: a machine-learning approach. In *Proc. ACM SIGMOD Conf.*, pages 509–520, 2001.

[DH05] AnHai Doan and Alon Y. Halevy. Semantic integration research in the database community: A brief survey. *AI Magazine*, 2005.

[dKvK04] A. de Keijzer and M. van Keulen. A possible world approach to uncertain relational data. In *SIUFDB-04 International Workshop on Supporting Imprecision and Uncertainty in Flexible Databases, Zaragoza, Spain, September 3, 2004*, DEXA Workshops. IEEE Computer Society, September 2004.

[ELLS01] Thomas Eiter, James J. Lu, Thomas Lukasiewicz, and V. S. Subrahmanian. Probabilistic object bases. *ACM Trans. Database Syst.*, 26(3):264–312, 2001.

[FKL97] Daniela Florescu, Daphne Koller, and Alon Y. Levy. Using probabilistic information in data integration. In Matthias Jarke, Michael J. Carey, Klaus R. Dittrich, Frederick H. Lochovsky, Pericles Loucopoulos, and Manfred A. Jeusfeld, editors, *VLDB'97, Proceedings of 23rd International Con-*

ference on Very Large Data Bases, August 25-29, 1997, Athens, Greece, pages 216–225. Morgan Kaufmann, 1997.

[HGS03] Edward Hung, Lise Getoor, and V. S. Subrahmanian. Pxml: A probabilistic semistructured data model and algebra. In Umeshwar Dayal, Krithi Ramamritham, and T. M. Vijayaraman, editors, *Proceedings of the 19th International Conference on Data Engineering, March 5-8, 2003, Bangalore, India*, pages 467–. IEEE Computer Society, 2003.

[Lev99] Alon Y. Levy. Combining artificial intelligence and databases for data integration. In *Artificial Intelligence Today*, pages 249–268. 1999.

[MBR01] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proc. 27th Int.Conf.on Very Large Databases*, pages 49–58, 2001.

[MZ98] Tova Milo and Sagit Zohar. Using schema matching to simplify heterogeneous data translation. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, pages 122–133. Morgan Kaufmann, 1998.

[NJ02] Andrew Nierman and H. V. Jagadish. ProTDB: Probabilistic data in XML. In *Proceedings of the 28th VLDB Conference*, 2002.

[RB01] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, 10(4):334–350, December 2001.

[SD05] Dan Suciu and Nilesh N. Dalvi. Foundations of probabilistic answers to queries. In *SIGMOD Conference*, page 963, 2005. Bibliographic notes to this tutorial: http://www.cs.washington.edu/homes/suciu/tutorial-sigmod2005-bib.pdf.

[vKdKA05] M. van Keulen, A. de Keijzer, and W. Alink. A probabilistic XML approach to data integration. In *Proceedings of the International Conference on Data Engineering, ICDE 2005*, 2005.

URLs

[AMG] All Movie Guide. http://www.allmovie.com.

[IMDb] The Internet Movie Database (IMDb). http://www.imdb.com.

[SS] Simply Scripts - Free Movie Scripts and Screenplays. http://www.simplyscripts.com.

[Yahoo] Yahoo! Movies. http://movies.yahoo.com.