Quality Measures in Uncertain Data Management

Ander de Keijzer and Maurice van Keulen

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

Abstract. Many applications deal with data that is uncertain. Some examples are applications dealing with sensor information, data integration applications and healthcare applications. Instead of these applications having to deal with the uncertainty, it should be the responsibility of the DBMS to manage all data including uncertain data. Several projects do research on this topic. In this paper, we introduce four measures to be used to assess and compare important characteristics of data and systems: uncertainty density, answer decisiveness and adapted precision and recall measures.

1 Introduction

Many applications somehow depend on uncertain data. Currently, most of these applications handle this uncertainty themselves, or just ignore the uncertainty associated with the data. Since the uncertainty is associated with the data, the database would be the logical system to store and handle this uncertainty.

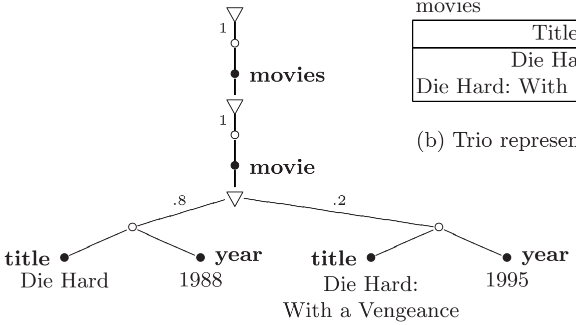
In recent years, the interest in management of uncertain data has increased greatly. Several projects on the subject have been initiated. A few examples in the relational setting are Trio [7], MystiQ [3] and ORION [4] and in the semistructured setting PXML [5] and IMPrECISE [6].

Since the topic *management of uncertain data* is relatively new to the database area, there is currently in our opinion a lack of means to assess and compare important characteristics of data and systems.

The contribution of this paper is the introduction of four measures for uncertain data and data management systems: uncertainty density, answer decisiveness, and specifically adapted notions of precision and recall measures to assess answer quality. We have tried to define the measures in a generic way to enable comparison between relational and XML systems.

The paper is organized as follows. Section 2 gives a short introduction into uncertain data and uncertain data management. In this paper we will use our own system IMPrECISE as a reference system for uncertain XML data and Trio as a reference system for uncertain relational data. We subsequently introduce the four measures for uncertain data in Section 3. The experiments in Section 4 are geared towards evaluating the behavior of the measures to validate their usefulness. Sections 5 and 6 contain conclusions and directions for future research.

H. Prade and V.S. Subrahmanian (Eds.): SUM 2007, LNAI 4772, pp. 104–115, 2007.



(a) IMPrECISE representation

movies

Title	Year	conf
Die Hard	1988	.8
Die Hard: With a Vengeance	1995	.2

(b) Trio representation

Fig. 1. Example movie database in IMPrECISE and Trio

2 Uncertain Data

Although all of the mentioned projects have their own unique details, they do have one aspect in common. In every project, the central theme is uncertain data. If we consider data to represent, or describe objects in the real world, then uncertain databases describe possible appearances of these objects.

As an example, we show a movie database with one movie that is the result of an integration between two different movie databases. Figure 1 shows this integrated database for two representative systems: IMPrECISE as a representative of an XML-based system (Figure 1(a)) and Trio as an representation of a relational system (Figure 1(b)). The databases show a movie database containing the title and year. Both databases hold information on one movie, but both the title and the year of the movie are uncertain. The title of the movie is either "Die Hard" or "Die Hard: With a Vengeance" and the year of the movie is either 1988 or 1995 respectively. Note that in Trio this single movie is captured in one *x-tuple* containing two alternative representations, or simply *alternatives.*

Another central concept in uncertain databases, is that of possible worlds. A possible world is a possible representation of the real world and is constructed by taking one possibility or alternative for each of the real world objects. In the previous example, there are 2 possible worlds, since only 1 real world object with 2 possible representations is captured in the database. Note that the alternatives for title and year are dependent here. Independent alternatives resulting in 4 possible worlds can of course also be represented in both systems. For the case where a real world object possibly doesn't exist, indicated by an empty possibility node in IMPrECISE or a question mark in Trio, this inexistence is also a possible appearance of the real world object when constructing the possible worlds.

Querying uncertain data results in uncertain answers. If probabilities are associated with the data, these are accessible in the query result as well. Query languages for uncertain data closely resemble query languages for *normal* data. In Trio the query language is called TriQL and is a superset of SQL. In

IMPrECISE the query language is a probabilistic version of XQuery. Although the syntax of the languages are (almost) equal to their normal counterparts, the semantics of course differs. Instead of returning answers to the questions, the system returns *possible answers*. The possible answers can be obtained by evaluating the query for each possible world. Of course this is the semantics behind query evaluation and in neither of the systems it is the actual execution plan.

2.1 IMPrECISE

We use the IMPrECISE system for the experiments in Section 4, so we give some more detail on this system here. The IMPrECISE system uses XML as a data model. The advantage of XML is that it more naturally and generically captures uncertainty. because it closely resembles a decision tree. The expressiveness is, because of the tree structure, high. We introduced two new kinds of nodes, probability nodes ($\bigtriangledown$) and possibility nodes ($\circ$). The root node is always a probability node, child nodes of probability nodes are possibility nodes, child nodes of possibility nodes are regular XML nodes and these, in turn, have probability nodes as child nodes.

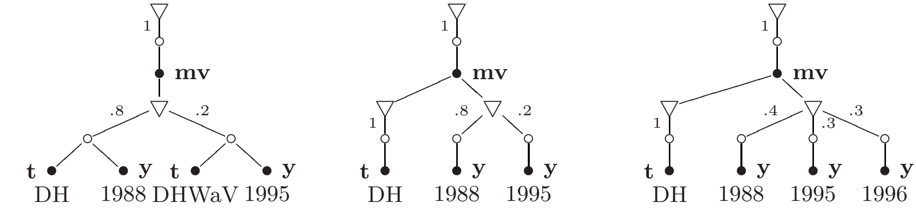
Probability nodes indicate *choice points*. Sibling child nodes are mutually exclusive, which introduces possibilities. Each possibility has an associated probability. Probabilities of sibling possibility nodes sum up to at most 1. More details on this model can be found in [6].

IMPrECISE is developed as an XQuery module for the MonetDB/XQuery DBMS [2]. In this way, it demonstrates the power of this XML DBMS and the XQuery language as well.

3 Measures

The measures we introduce in this section can be used for all data models, as long as local possibilities or alternatives can be identified. In IMPrECISE probabilities are always local, because the probability associated with a possibility node expresses the likelihood of the subtree of that particular possibility node to hold the correct information about the real world. In Trio, probabilities are associated with alternatives, which indicate the likelihood of an alternative being correct in the real world. This type of probability is also local. The number of choice points in IMPrECISE is equal to the number of probability nodes, since at each of these nodes a choice for one of the possibility nodes has to be made. In Trio the choice points are determined by the number of *x-tuples* in the relation. For each x-tuple one alternative has to be chosen.

We first define some notation. Let N_{cp} be the number of choice points in the data (i.e., probability nodes in IMPrECISE), $N_{poss,cp}$ the number of possibilities or alternatives of choice point cp, and let P_{cp}^{max} be the probability of the most likely possibility of choice point cp.



$N_{cp} = 2,$
$N_{poss,1} = 1, N_{poss,2} = 2$
$P_1^{max} = 1, P_2^{max} = .8$

Dens$= 1 - \frac{1}{2}(\frac{1}{1} + \frac{1}{2})$
$= \frac{1}{4} = .25$
Dec$= \frac{1}{2}(1 + \frac{.8}{1.2})$
$= \frac{5}{6} = .83$

(a) Example A

$N_{cp} = 3,$
$N_{poss,1} = N_{poss,2} = 1,$
$N_{poss,3} = 2, P_3^{max} = .8$
$P_1^{max} = P_2^{max} = 1,$

Dens$= 1 - \frac{1}{3}(\frac{1}{1} + \frac{1}{1} + \frac{1}{2})$
$= \frac{1}{6} = .17$
Dec$= \frac{1}{3}(1 + 1 + \frac{.8}{1.2})$
$= \frac{8}{9} = .89$

(b) Example B

$N_{cp} = 3,$
$N_{poss,1} = N_{poss,2} = 1,$
$N_{poss,3} = 3, P_3^{max} = .4$
$P_1^{max} = P_2^{max} = 1,$

Dens$= 1 - \frac{1}{3}(\frac{1}{1} + \frac{1}{1} + \frac{1}{3})$
$= \frac{2}{9} = .22$
Dec$= \frac{1}{3}(1 + 1 + \frac{.4}{1.6 \times \log_2 3})$
$= \frac{1}{3}(2 + \frac{.4}{2.536}) = .72$

(c) Example C

Fig. 2. Examples of uncertainty density and decisiveness

3.1 Uncertainty Density

An often used measure for the amount of uncertainty in a database is the number of possible worlds it represents. This measure, however, exaggerates the perceived amount of uncertainty, because it grows exponentially with linearly growing independent possibilities. Furthermore, we would like all measures to be numbers between 0 and 1. We therefore propose the *uncertainty density* as a measure for the amount of uncertainty in a database. It is based on the average number of alternatives per choice point:

$$\text{Dens} = 1 - \frac{1}{N_{cp}} \sum_{j=1}^{N_{cp}} \frac{1}{N_{poss,j}}$$

Dens is 0 for a databases that contains no uncertainty. Dens decreases if there is more certain data in the database for the same amount of uncertain data (compare Figures 2(a) and 2(b)). Dens rises if a choice point contains more alternatives (compare Figures 2(b) and 2(c)). If all choice points contain n alternatives, Dens is $(1-\frac{1}{n})$, which approaches 1 with growing n. The uncertainty density is independent of the probabilities in the database. It can be used, for example, to relate query execution times to, because query execution times most probabily depend on the number of alternatives to consider.

3.2 Answer Decisiveness

Even if there is much uncertainty, if one possible world has a very high probability, then any query posed to this uncertain database will have one, easy to distinguish, most probable answer. We say that this database has a high *answer decisiveness*. In contrast, if there is much uncertainty and the probabilities are rather evenly distributed over the possible worlds, then possible answers to queries will be likely to have similar probabilities. We have defined the answer decisiveness as

$$\text{Dec} = \frac{1}{N_{cp}} \sum_{j=1}^{N_{cp}} \frac{P_j^{max}}{(2 - P_j^{max}) \times \log_2(max(2, N_{poss,j}))}$$

Dec is 1 for a database that contains no uncertainty, because each term in the sum becomes $\frac{1}{(2-1)\times\log_2 2} = 1$. If at each choice point j with two alternatives, there is one with a probability close to one (i.e., P_j^{max} is close 1), then all terms for j are also close to 1 and Dec is still almost 1. When P_j^{max} drops for some j, then Dec drops as well. Dec also drops when choice points occur with growing numbers of alternatives. This is accomplished by the $\log_2(\max(2, N_{poss,j}))$ factor (compare Figures 2(b) and 2(c)). We have taken the logarithm to make it decrease gradually.

3.3 Answer Quality

Querying uncertain data results in answers containing uncertainty. Therefore, an answer is not correct or incorrect in the traditional sense of a database query. We need a more subtle notion of answer quality.

In the possible world approach, an uncertain answer represents a set of possible answers each with an associated probability. In Trio, it is possible to work with alternatives without probabilities, but these can be considered as equally likely, hence with uniformly distributed probabilities. The set of possible answers ranked according to probability has much in common with the result of an information retrieval query. We therefore base our answer quality measure on precision and recall [1]. We adapt these notions, however, by taking into account the probabilities of the possible answers. Correct answers with high probability are better than correct answers with a low probability. Analogously, incorrect answers with a high probability are worse than incorrect answers with a low probability.

XQuery answers are always sequences. The possible answers to an XQuery on an uncertain document, however, largely contain the same elements. Therefore, we construct an amalgamated answer by merging and ranking the elements of all possible answers. This can be accomplished in XQuery with the function in Figure 3. The effectiveness of this approach to querying a probabilistic database can be illustrated with an example. Suppose we query a probabilistic movie database asking for horror movies: `//movie[.//genre="Horror"]/title`. Even though the integrated document may contain thousands of possible worlds, the amalgamated answer is restricted to the available movie titles considered to be possibly belonging to a horror movie, which will be few in number.

```
declare function rank_results($pws as element(world)*)
   as element(answer)*
{
   for $v in distinct-values($pws/descendant::text())
   let $ws := $pws[./descendant::text()[.=$v]]
      ,$rank := sum($ws/@prob)
   order by $rank descending
   return <answer rank="{$rank}">{$v}</answer>
};
```

Fig. 3. XQuery function for ranking query results

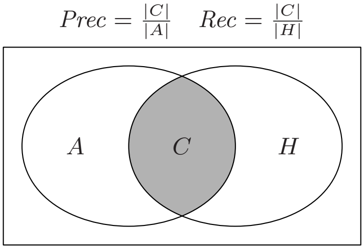


Fig. 4. Precision and recall

Precision and recall are traditionally computed by looking at the presence of correct and incorrect answers. Let H be the set of correct answers to a query (as determined by a human), A the set of answers (the elements of the amalgamated query answer), and C the intersection of the two, i.e., the set of correct answers produced by the system (see Figure 4).

We adapt the precision and recall measures by taking into account the probabilities: An answer a is only present in the amount prescribed by its probability $P(a)$. This reasoning gives us the following definitions for precision and recall.

$$\text{Prec} = \frac{\sum_{a \in C} P(a)}{|C| + \sum_{a \in (A-C)} P(a)} \qquad \text{Rec} = \frac{\sum_{a \in C} P(a)}{|H|}$$

For example, say the answer to the query "Give me all horror movies" is "Jaws" and "Jaws 2". If the system returns this answer, but with a confidence of 90% for both movies, then precision and recall are both 90%. If, however, it also gives some other (incorrect) movie with a confidence of 20%, then precision drops to 82% and recall stays 90%.

4 Experiments

4.1 Set Up

The contributions of this paper are the uncertainty density, decisiveness, and answer quality measures. The purpose of the experiments hence is not to validate or compare systems or techniques, but an evaluation of the behavior of the measures to validate their usefulness.

As application of uncertainty in data, we selected data integration. In our research on IMPrECISE, we attempt to develop data management functionality for uncertain data to be used for this application area. When data sources contain data overlap, i.e., they contain data items referring to the same real world

name	repr.	#pws	#nodes
2x2	tree	16	469
4x4	tree	2,944	7,207
6x6	tree	33,856	25,201
6x9	tree	2,258,368	334,616
2x2 +rule	tree	4	328
4x4 +rule	tree	64	2,792
6x6 +rule	tree	256	8,328
6x9 +rule	tree	768	21,608
6x15 +rule	tree	3,456	87,960
2x2	dag	16	372
4x4	dag	2,944	1,189
6x6	dag	33,856	2,196
6x9	dag	2,258,368	13,208
2x2 +rule	dag	4	280
4x4 +rule	dag	64	761
6x6 +rule	dag	256	1,243
6x9 +rule	dag	768	1,954
6x15 +rule	dag	3,456	4,737

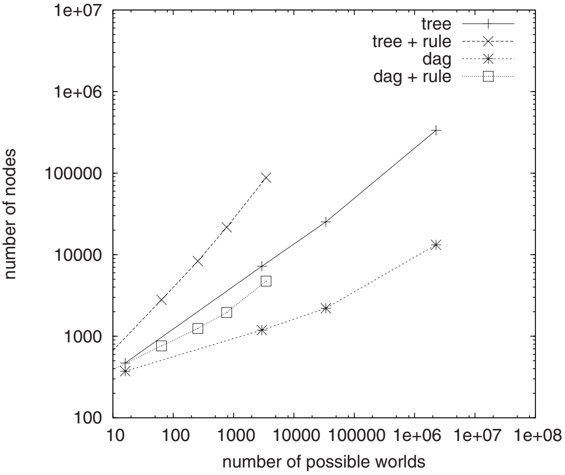
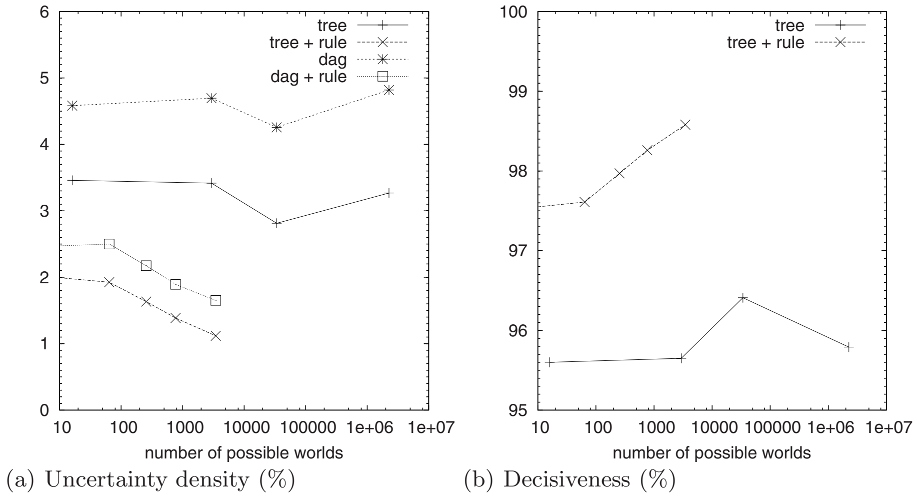


Fig. 5. Data sets (pws = possible worlds)

objects, they may conflict and it is not certain which of the sources holds the correct information. Moreover, without human involvement, it is usually not possible for a data integration system to establish with certainty which data items refer to the same real world objects. To allow for unattended data integration, it is imperative that the data integration system can handle this uncertainty and that the resulting (uncertain) integrated source can be used in a meaningful way.

The data set we selected concerns movie data: Data set 'IMDB' is obtained from the Internet Movie DataBase from which we converted title, year, genre and director data to XML. Data set 'Peggy' is obtained from an MPEG-7 data source of unknown but definitely independent origin. We selected those movies from these sources that create a lot confusion: sequels, documentaries, etc. of 'Jaws', 'Die Hard', and 'Mission Impossible'. Since the titles of these data items look alike, the data integration system often needs to consider the possibility of those data items referring to the same real-world objects, thus creating much uncertainty in the integration result. The integrated result is an XML document according to the aforementioned probabilistic tree technique [6].

To create integrated data sets of different sizes and different amounts of uncertainty, we integrated 2 with 2 movies selected from the sources, 4 with 4, 6 with 6, and 6 with 15 movies. We furthermore performed this integration with (indicated as '+rule') and without a specific additional rule that enables the integration system to much better distinguish data about different movies. This results in data sets with different characteristics. To be able to investigate uncertainty density, we additionally experiment with the data represented as tree as well as DAG. Although our implementation of the DAG representation does not produce the most optimally compact DAG yet, it suffices to experiment with its effect on uncertainty density. See Figure 5 for details of the data sets and an indication of the compactness of the representation.



(a) Uncertainty density (%)

(b) Decisiveness (%)

Fig. 6. Uncertainty density and decisiveness

4.2 Uncertainty Density

Figure 6(a) shows the uncertainty density for our data sets. There is a number of things to observe.

- Density values are generally rather low. This is due to the fact that integration produces uncertain data with mostly choice points with only one alternative (certain data) and relatively few with two alternatives (uncertain data). For example, the '6x9 tree' case has 74191 choice points with one alternative and 5187 choice points with two alternatives.
- When comparing the lines for 'tree' with 'dag', and 'tree + rule' with 'dag + rule', we observe that the dag-versions have a considerable higher uncertainty density. This can be explained by the fact that the DAG representation shares common subtrees. Most commonality appears for certain data that occurs in all possible worlds. Hence, *relatively* more nodes are devoted to uncertainty in the DAG representation. The uncertainty density measure correctly exhibits this behavior.
- When comparing the lines for 'tree' with 'tree + rule', and 'dag' with 'dag + rule', we observe that the additional rule not only reduces the number of possible worlds, but also reduces the uncertainty density. The knowledge of the rule reduces uncertainty, but the amount of certain information stays the same. Therefore, it is logical that the uncertainty density goes down.
- The '+ rule' lines drop with growing database size, while the other two do not. Database growth in this experiment means additional movies in both data sources. The specific rule we used in this experiment helps the integration system to determine which pairs of data items from both sources cannot possibly refer to the same real world object. The density measure

correctly shows that the additional movies cause relatively more confusion without the rule than with it.

In general, we can say that important characteristics concerning the amount of uncertainty in the database can be assessed successfully with the uncertainty density measure. Moreover, it does not suffer from the disadvantage of exaggeration that the number of possible worlds has.

4.3 Answer Decisiveness

Figure 6(b) shows the answer decisiveness for our data sets. This experiment focuses on the tree representation only, because the answers produced by a query is independent of the representation, hence the answer decisiveness does not depend on the representation. There are a number of things to observe.

- Decisiveness values are generally rather high. This has the same reason as why density is generally low: there are mostly choice points with only one alternative and few with two alternative, hence in most cases it is easy to make a choice for an answer because there is only one to choose from.
- Similar patterns in the lines for decisiveness can be observed when comparing with uncertainty density. Both measures are related, because the more alternatives per choice point on average, the higher the uncertainty density, but also the lower the decisiveness. Decisiveness only starts to deviate from density if the associated probabilities ensure that it is easy to choose the most likely possible answer. The probability assignment logic in our system, however, is still in its infancy and is apparently not capable of giving good decisiveness despite high uncertainty density.

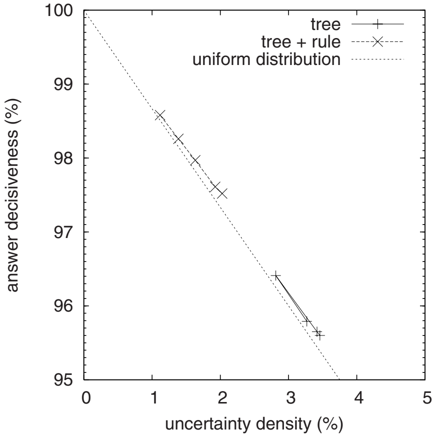


Fig. 7. Density vs. Decisiveness

The relationship between the density and decisiveness measures is illustrated by Figure 7. The straight line marked 'uniform distribution' is drawn for the situation where the probabilities are always uniformly distributed and, for simplicity, where there are only choice points with at most two alternatives (which is the case for our test data and which makes the line straight). In this situation, uncertainty density fully determines answer decisiveness. The fact that the lines are not on the straight line shows that the probability assignment logic of our system has some impact on decisiveness despite the uncertainty density, but the impact is (as expected) rather limited. We expect that an integration system with better probability assignment logic will produce points much higher

Table 1. Answer quality (‘X’ marks an incorrect answer)

(a) Query 1: `//movie[.//genre="Horror"]/title` (All horror movies)

Poll.	P(a)	Answer	Prec	Rec
2	79.4%	“Jaws”	79.4%	79.4%
	79.4%	“Jaws 2”		
5	77.4%	“Jaws”	69.5%	77.4%
	77.4%	“Jaws 2”		
	22.6%	X “Ma@ing of Steven Spielberg’s ’Jaws’, The”		
10	85.4%	“Jaws”	74.5%	85.4%
	85.4%	“Jaws 2”		
	29.2%	X “Ma@ing of Steven Spielberg’s ’Jaws’, The”		
20	85.4%	“Jaws”	74.5%	42.7%
	14.6%	X “Ma@ing of Steven Spielberg’s ’Jaws’, The”		

(b) Query 2: `//movie[./year="1995"]/title` (All movies produced in 1995)

Poll.	P(a)	Answer	Prec	Rec
2	100.0%	“Die Hard: With a Vengeance”	100.0%	100.0%
	100.0%	“Behind the Scenes: Die Hard - With a Vengeance”		
	100.0%	“Making of Steven Spielberg’s ’Jaws’, The”		
5	79.4%	“Die Hard: With a Vengeance”	56.3%	64.3%
	58.8%	“Behind the Scenes: Die Hard - With a Vengeance”		
	54.8%	“Making of Steven Spielberg’s ’Jaws’, The”		
	20.6%	X “Behind th@ Scenes: Die Hard - With a Vengeance”		
	11.3%	X “Ma@ing of Steven Spielberg’s ’Jaws’, The”		
	5.6%	X “Jaws”		
	5.6%	X “Jaws 2”		
10	85.4%	“Die Hard: With a Vengeance”	47.1%	56.3%
	41.7%	“Behind the Scenes: Die Hard - With a Vengeance”		
	41.7%	“Making of Steven Spielberg’s ’Jaws’, The”		
	21.9%	X “Behind th@ Scenes: Die Hard - With a Vengeance”		
	14.6%	X “Ma@ing of Steven Spielberg’s ’Jaws’, The”		
	7.3%	X “Jaws”		
	7.3%	X “Jaws 2”		
	7.3%	X “Die Hard 2”		
20	78.1%	“Die Hard: With a Vengeance”	52.6%	53.8%
	41.7%	“Behind the Scenes: Die Hard - With a Vengeance”		
	41.7%	“Making of Steven Spielberg’s ’Jaws’, The”		
	7.3%	X “Behind th@ Scenes: Die Hard - With a Vengeance”		

(c) Query 3: `//movie[./title="Jaws 2"]/year` (When has Jaws 2 been produced?)

Poll.	P(a)	Answer	Prec	Rec
2	69.1%	“1978”	62.6%	69.1%
	10.3%	X “1975”		
5	66.1%	“1978”	59.4%	66.1%
	5.6%	X “1975”		
	5.6%	X “1995”		
10	78.1%	“1978”	72.8%	78.1%
	7.3%	X “1995”		
20	78.1%	X “197@”	0.0%	0.0%
	7.3%	X “@995”		

in the graph. Most importantly, the decisiveness measure can be effectively used to measure the quality of the probability assignment logic.

4.4 Answer Quality

To obtain test data suitable for evaluating our answer quality measure, we took one of the data sources: an IMDB document with 9 movies. We made two copies of it, randomly polluted them by corrupting text nodes, and then integrated them. We made sure we didn't pollute the same text nodes, so 'the truth' is still available in the combined data of both sources and an ideal integration system would be able to reconstruct it. We furthermore took three queries and posed them to the data integration result of data sources with increasing pollution. A pollution of 2 means that 2 randomly chosen text nodes in both source have been corrupted by changing a randomly chosen character to '@'. This pollution not only affects the data integration, also in some of the answers we see these modified strings appear. Although they are seemingly almost correct, we classified these answers as incorrect.

Table 1 shows the answer quality measurements for the three queries. Even though our system produces the correct answers in most cases, the confidence scores the system produces are rather modest. This is due to the naive probability assignment explained earlier. Our adapted precision and recall measures effectively reflect this aspect of reduced answer quality. Missing answers (as in Query 1 / Pollution 20, and Query 3 / Pollution 20) is of course worse than just modest confidence scores; indeed radically lower recall is given to these cases.

5 Conclusions

In this paper we introduced several new measures for assessment and comparison of important characteristics of uncertain data and uncertain data management systems: uncertainty density, decisiveness, and modifications of two existing answer quality measures, precision and recall.

In contrast with the number of possible worlds as a measure for the amount of uncertainty present in the database, the uncertainty density measure doesn't exaggerate this uncertainty. The uncertainty density is based on the average number of alternatives per choice point, hence it also takes into account the amount of certain data.

The answer decisiveness is an indication how well in general a most likely answer can be distinguished among a set of possible answers. Even in the presence of much uncertainty, if one possible world has very high probability, then any query posed to this uncertain database will have one easily distinguishable most likely answer. The decisiveness is an indication of how well the confidence scores in the document were assigned. The ratio between decisiveness and density also shows this fact. The ratio can be used to evaluate how much the probabilities deviate from uniform distribution, i.e., how much the system tends to confidently give a high probability to one answer, hence aiding the user or application in selecting the most probable answer.

High decisiveness does of course not mean that the answers the system so adamantly claims to be the most probable ones, are indeed the correct answers. Therefore, we introduced adapted precision and recall measures to evaluate answer quality which takes into account the probabilities assigned to the answers.

6 Future Research

As a next step of this research, we plan to improve IMPrECISE and validate the improvements using the quality measures. For this purpose, a central component in the system which assigns the probabilities, called "The Oracle", has to be improved. "The Oracle" determines, at integration time, how likely it is that two elements refer to the same real world object. An improved "Oracle" will give a increased values for decisiveness, precision and recall.

The current DAG implementation does not produce the most compact representation of uncertain data possible. We have identified some patterns that can be used to improve the current implementation.

One of the reasons for inefficiency in querying at the moment, is confidence computation. In order to speed up this process we plan to investigate if provenance, or lineage as used in Trio is suitable for our model.

Another item on our agenda is to release IMPrECISE as a module of MonetDB/XQuery. Before we can do this, the probabilistic query functionality has to be extended and some operators and functions dealing with the confidences associated with possibility nodes, have to be made available.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, Reading (1999)
2. Boncz, P.A., Grust, T., van Keulen, M., Manegold, S., Rittinger, J., Teubner, J.: MonetDB/XQuery: a fast XQuery processor powered by a relational engine. In: Proc. of SIGMOD, Chicago, IL, USA, pp. 479–490 (2006)
3. Boulos, J., Dalvi, N.N., Mandhani, B., Mathur, S., Re, C., Suciu, D.: MYSTIQ: a system for finding more answers by using probabilities. In: Proc. of SIGMOD, Baltimore, Maryland, USA, pp. 891–893 (2005)
4. Cheng, R., Singh, S., Prabhakar, S.: U-DBMS: A database system for managing constantly-evolving data. In: Proc. of VLDB, Trondheim, Norway, pp. 1271–1274 (2005)
5. Hung, E., Getoor, L., Subrahmanian, V.S.: PXML: A probabilistic semistructured data model and algebra. In: Proc. of ICDE, Bangalore, India, pp. 467–478 (2003)
6. van Keulen, M., de Keijzer, A., Alink, W.: A probabilistic xml approach to data integration. In: Proc. of ICDE, Tokyo, Japan, pp. 459–470 (2005)
7. Mutsuzaki, M., Theobald, M., de Keijzer, A., Widom, J., Agrawal, P., Benjelloun, O., Sarma, A.D., Murthy, R., Sugihara, T.: Trio-One: Layering uncertainty and lineage on a conventional DBMS. In: Proc. of CIDR, Monterey, USA, pp. 269–274 (2007)