Evaluating Matching Algorithms: the Monotonicity Principle

Ateret Anaby-Tavor and Avigdor Gal
Technion – Israel Institute of Technology
{avigal@ie,ateret@tx}.technion.ac.il
Alberto Trombetta
Università dell'Insubria
alberto.trombetta@uninsubria.it

Abstract

In this paper we present the *monotonicity principle*, a sufficient condition to ensure that *exact mapping*, a mapping as would be performed by a human observer, is ranked close to the *best mapping*, as generated automatically by a matching algorithm. The research is motivated by the introduction of the semantic Web vision and the shift towards machine understandable Web resources. We support the importance of the monotonicity principle by empirical analysis of a matching algorithm, showing that algorithms that obey this principle rank the exact mapping close to the best mapping.
keywords: Ontology matching, Novel integration architectures

1 Introduction and motivation

The ambiguous interpretation of concepts, describing the meaning of data in heterogeneous data sources (*e.g.*, database schemata, XML DTDs, and HTML form tags) is commonly known as *semantic heterogeneity*. Semantic heterogeneity is a well-known obstacle to data source integration [5], a task that has become a common practice in automating Business-to-Business activities. Semantic heterogeneity is resolved through a process of *semantic reconciliation*, which matches concepts from heterogeneous data sources. Traditionally, semantic reconciliation was performed by a human observer (a designer or a DBA) [20; 15] due to its complexity [5]. However, manual reconciliation (with or without computer-aided tools) tends to be slow and inefficient in dynamic environments and does not scale for obvious reasons. Therefore, the introduction of the semantic Web vision [3] and the shift towards machine understandable Web resources has unearthed the importance of automatic semantic reconciliation. Consequently, new tools for automating the process, such as Cupid [17], GLUE [9], and OntoBuilder [19], were introduced.

Generally speaking, the process of semantic reconciliation is performed in two steps. First, given two attribute sets $\mathcal{A}$ and $\mathcal{A}'$ (denoted *schemata*) with n_1 and n_2 attributes, respectively,[1] a degree of similarity is computed **automatically** for all attribute pairs (one attribute from each schema),[2] using such methods as name matching, domain matching, structure (such as XML hierarchical representation) matching, and Machine Learning techniques (*e.g.*, [2; 8]). As a second step, a single mapping from $\mathcal{A}$ to $\mathcal{A}'$ is chosen to be the *best mapping*. Typically, the best mapping is the one that maximizes the sum (or average) of pair-wise weights of the selected attributes. We differentiate the best mapping from the *exact mapping*, which is the output of a matching process as would be performed by a human observer.

Automatic matching may carry with it a degree of uncertainty since "the syntactic representation of schemas and data do not completely convey the semantics of different databases" [18]. As an example, consider name matching, a common method in tools such as Cupid [17], OntoBuilder [12], Protégé [11], and Ariadne [16]. With name matching, one assumes that similar attributes have similar (or even identical) names. However, the occurrence of synonyms (*e.g.*, `remuneration` and `salary`) and homonyms (*e.g.*, `age` referring to either human age or wine age) may trap this method into erroneous mapping. As a consequence, there is no guarantee that the exact mapping is always the best mapping.

We present the *monotonicity principle*, a sufficient condition to ensure that exact mapping would be ranked sufficiently close to the best mapping. Roughly speaking, the monotonicity principle proclaims that by replacing a mapping with a better one, score wise, one gets a more accurate mapping (from a human observer point of view), even if by doing so, some of the attribute mappings are of less quality. The paper contribution is in demonstrating, through theoretical and empirical analysis, that for monotonic mappings that satisfy the monotonicity principle, one can safely interpret a high similarity measure as an indication that more attributes are mapped correctly. An immediate consequence of this result is the establishment of a corroboration for the quality of mapping algorithms, based on their capability to generate monotonic mappings. We have experimented with a matching algorithm and report on our experiences in Section 4. Our findings indicate that matching algorithms that generate monotonic mappings are well-suited for automatic semantic reconciliation. Another outcome of the monotonicity principle is that a

[1] The use of relational terms is in no way restrictive, and is used here to avoid the introduction of an extensive terminology that is of little benefit in this paper.

[2] Extensions to this basic model (*e.g.*, [18]) are beyond the scope of this paper.

good automatic semantic reconciliation algorithm would rank the exact mapping relatively close to the best mapping, thus enabling an efficient search of the exact mapping [1].

2 Preliminaries

We start by introducing the notions of *attribute similarity measure* and *mapping similarity measure*. The interested reader is referred to [13], where we have grounded these notions in a theoretical model of uncertainty based on fuzzy sets. To illustrate our model we shall use simplified schemata of two car rental reservation systems, as follows:

AvisRental(RentalNo, PickUpLocationCode, PickUpDate, PickUpHour, PickUpMinutes, ReturnDate, ReturnHour, ReturnMinutes, Price.)

AlamoRental(RentalNo, PickUpLocation, PickUp-Date, PickUpHour, PickUpMinutes, DropoffDate, DropoffHour, DropoffMinutes, Price.)

It is worth noting that this example was largely simplified, for the purpose of clarity. In Section 4 we provide our empirical analysis, which is based on data that was collected from multiple Web sites.

2.1 Attribute similarity

Given two attribute sets $\mathcal{A}$ and $\mathcal{A}'$, we associate a similarity measure, normalized as a similarity degree between 0 (total dissimilarity) and 1 (equivalence), with any mapping among attributes of $\mathcal{A}$ and $\mathcal{A}'$. Therefore, given two attributes $A \in \mathcal{A}$ and $A' \in \mathcal{A}'$, we say that A and A' are μ-*similar* to specify our belief in the mapping quality. The measure of similarity between A and A' is denoted by $\mu^{A,A'}$, or simply by μ, whenever the identity of the compared attributes is evident from the context. We assume that a manual matching is a perfect process with $\mu = 1$.[3] As for automatic matching, a hybrid of algorithms, such as presented in [8; 17; 19] or adaptation of relevant work in proximity queries (*e.g.*, [6]) and query rewriting over mismatched domains (*e.g.*, [7]) can determine the level of μ. For illustration purposes, consider a matching algorithm that computes μ, based on substring matching as follows. The similarity of two attributes A and A' is defined symmetrically as the maximum size of a matching substring in A and A' divided by the maximum number of characters in either A or A'. Consider next the schemata presented above, and let $A =$`PickUp-Date` and $A' =$`PickUpDate`. Then, $\mu = \frac{6 \text{ (for PickUp)}}{11 \text{ (for PickUp-Date)}} = 0.55$, due to the hyphen in A. However, by applying an IR (Information Retrieval) technique known as *dehyphanation*, `PickUp-Date` becomes `PickUpDate` and similarity increases dramatically to $\mu = 1$.

2.2 Mapping similarity

Given two attribute sets, $\mathcal{A}$ and $\mathcal{A}'$, a *mapping* F from $\mathcal{A}$ to $\mathcal{A}'$ is a set of pairs (A, A'), such that $A \in \mathcal{A} \cup \{null\}$, $A' \in \mathcal{A}' \cup \{null\}$, and $A' = F(A)$. A mapping with a null value represents no mapping. The *mapping similarity measure* μ^F is a function $\mu^F = h(\mu^{A,A'} | (A, A') \in F)$.

Attribute pair	μ
RentalNo,RentalNo	1
PickUpLocationCode,PickUpLocation	0.89
PickUpDate,PickUp-Date	1
PickUpHour,PickUpHour	1
PickUpMinutes,PickUpMinutes	0.95
ReturnDate,DropoffDate	0.68
ReturnHour,DropoffHour	0.68
ReturnMinutes,DropoffMinutes	0.7
Price,Price	1
	0.88

Table 1: Computing attribute-set similarity measure

Table 1 provides a mapping F, where each attribute pair is associated with an attribute similarity measure, computed using substring and domain matching. We shall demonstrate the μ computation using the (`AvisRental.PickUpMinutes`, `AlamoRental.PickUpMinutes`) pair. Using substring matching, the pair has a perfect match. However, we assume that the domains of the two attributes differ. While the domain of `AvisRental.PickUpMinutes` is {`0, 15, 30, 45`}, the domain of `AlamoRental.PickUpMinutes` is {`0, 10, 20, 30, 40, 50`}. Maximizing the pair-wise minimal Euclidean distance, one has that the domain similarity is 0.9, and by averaging the two similarity measures, one gets $\mu = 0.95$. Computing μ^F by averaging over $\mu^{A,A'}$ of all pairs (A, A') in F yields $\mu^F = 0.88$.

A few notes are in order at this point. First, we assume that pair-wise similarity measure is computed **automatically** by some matching algorithm, and does not involve human intervention. Second, the mapping F in Table 1, while being with most likelihood the exact mapping, is only one among many ($n!$ for $1:1$ matching). Finally, a mapping can be $1:1$ (in which case the mapping becomes a $1:1$ and onto function), $1:n$ (where an attribute from the scope can be mapped into multiple attributes in the domain), or $n:1$ (see [4] for more details). Methods for computing the best mapping depend on the type of mapping. For example, for a $1:1$ matching, algorithms for identifying the best mapping typically rely on weighted bipartite graph matching [14].

To simplify the discussion in the rest of this paper, we shall assume that $|\mathcal{A}| = |\mathcal{A}'| = n$. Extending the discussion to other cases is straightforward.

3 Monotonic mappings: measuring matching quality

In this section we aim at modeling the relationship between a choice of a mapping, based on mapping similarity, and a choice of a mapping, as performed by a human observer. As we empirically show in Section 4, the more correlated these mappings are, the more effective an automatic mapping process becomes. In order to compare the effectiveness of various choices of mappings and operators, we introduce the no-

[3] This is, obviously, not always the case. In the absence of sufficient background information, human observers are bound to err as well. However, since the monotonicity principle is based on comparing the best mapping with the exact mapping, and the latter is based on human interpretation, we keep this assumption.

tion of mapping *imprecision*, which follows common IR practice for retrieval effectiveness (*e.g.*, [10]). Assume first that among all possible mappings between two attribute sets of cardinality n, we choose one and term it the *exact mapping* (denoted $\bar{F}$). Intuitively, the exact mapping is the best possible mapping, as conceived by a human observer. Having selected the exact mapping between $\mathcal{A}$ and $\mathcal{A}'$, we measure the imprecision of any other mapping G simply by counting how many arguments of $\bar{F}$ and G do not coincide. We next present a formal definition of mapping imprecision.

Definition 1 *Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ and $\mathcal{A}' = \{A'_1, \ldots, A'_n\}$ be attribute sets of cardinality n. Also, let F and G be two mappings over $\mathcal{A}$ and $\mathcal{A}'$ and let $A_i \in \mathcal{A}$ be an attribute. F and G* differ *on A_i if $F(A_i) \neq G(A_i)$. $\mathcal{D}^{F,G}$ denotes the set of attributes of $\mathcal{A}$ on which F and G differ .*

Definition 2 (Imprecision) *Let $\bar{F}$ be an exact mapping over $\mathcal{A}$ and $\mathcal{A}'$ and let G be some mapping over $\mathcal{A}$ and $\mathcal{A}'$ such that there are $m \leq n$ attributes in $\mathcal{A}$ on which $\bar{F}$ and G differ. Then G is* m-imprecise *(with respect to $\bar{F}$). We denote by i_G the imprecision of G.*

The exact mapping between `AvisRental` and `AlamoRental` is given in Table 1. A possible 2-imprecise mapping is a mapping that varies from the one presented in Table 1 by associating `PickUpDate` with `DropoffDate` and `ReturnDate` with `Pickup-Date`.

Definition 3 (Similarity preservation) *Let F and G be mappings over attribute sets $\mathcal{A}$ and $\mathcal{A}'$. F and G are* similarity preserving *on an attribute $A \in \mathcal{A}$ if $i_F < i_G$ implies $\mu^{A,F(A)} > \mu^{A,G(A)}$. $\mathcal{M}^{F,G}$ denotes the set of attributes of $\mathcal{A}$ on which F and G are similarity preserving.*

Example 1 *Consider the 2-imprecise mapping given above. Associating `PickUpDate` with `DropoffDate` yields an attribute similarity measure of $\mu = 0.68$, and associating `ReturnDate` with `Pickup-Date` yields $\mu = 0.7$. Referring to this mapping as G and to the mapping of Table 1 as $\bar{F}$, $\bar{F}$ and G are similarity preserving on attribute `Pickup-Date`, since $0 = i_{\bar{F}} < i_G = 2$ and $1 = \mu^{A,\bar{F}(A)} > \mu^{A,G(A)} = 0.68$. However, $\bar{F}$ and G are not similarity preserving on attribute `ReturnDate`, since $0.68 = \mu^{A,\bar{F}(A)} \not> \mu^{A,G(A)} = 0.7$.* □

Definition 4 (Benefit and cost) *Let F and G be mappings over attribute sets $\mathcal{A}$ and $\mathcal{A}'$, such that $i_F < i_G$. Let $\bar{\varpi} = (\varpi_1, ..., \varpi_n)$ be a weight vector that sums to unity, associating with each attribute $A_i \in \mathcal{A}$ a weight ϖ_i. The* benefit *of switching from G to F is defined to be*

$$Benefit(F,G) = \sum_{A_k \in \mathcal{D}^{F,G} \cap \mathcal{M}^{F,G}} \left(\varpi_k \left(\mu^{A_k, F(A_k)} - \mu^{A_k, G(A_k)} \right) \right)$$

The cost *of switching from G to F is defined to be*

$$Cost(F,G) = \sum_{A_k \in \mathcal{D}^{F,G} \setminus \mathcal{M}^{F,G}} \left(\varpi_k \left(\mu^{A_k, G(A_k)} - \mu^{A_k, F(A_k)} \right) \right)$$

$Benefit(F,G)$ represents the benefit of switching from G to F. $\mathcal{D}^{F,G} \cap \mathcal{M}^{F,G}$ represents those attributes over which F and G differ, yet are similarity preserving. $Cost(F,G)$ represents the loss involved in switching from G to F. $\mathcal{D}^{F,G} \setminus \mathcal{M}^{F,G}$ represents those attributes over which F and G differ, and that are not similarity preserving.

Definition 5 (Monotonicity Principle) *Let $\mathcal{F} = \{F_1, F_2, ..., F_m\}$ be a set of mappings over attribute sets $\mathcal{A}$ and $\mathcal{A}'$. $\mathcal{F}$ is* monotonic *if the following inequality holds for any pair $\{F_i, F_j\} \subseteq \mathcal{F}$ such that $i_{F_i} < i_{F_j}$:*

$$Benefit(F_i, F_j) > Cost(F_i, F_j) \quad (1)$$

Each term in $Benefit(F_i, F_j)$ adds to the overall similarity, yet the attributes that participate in computing $Cost(F_i, F_j)$ lower the overall similarity by switching from F_j to F_i. If the benefit of switching from F_j to F_i surpasses the cost for all pairs $\{F_i, F_j\} \subseteq \mathcal{F}$ such that $i_{F_i} < i_{F_j}$, we consider the set to be monotonic. If the exact mapping is chosen among monotonic mappings, then the following holds: if $\bar{F} \in \mathcal{F}$ and $\mathcal{F}$ is monotonic then $\bar{F}$'s overall similarity measure is greater than the overall similarity degrees of i-imprecise mappings in $\mathcal{F}$, even if such mappings yield better similarity degrees on some attribute pairs.

In [13] we have used the monotonicity principle to show a practical result that holds for a monotonic set of mappings. We show that, under certain conditions on h (the aggregation function that is used for computing μ^F for a mapping F), one can formally show that $i_F < i_G$ entails $\mu^F > \mu^G$.

Example 2 (Monotonic mappings) *The set of possible mappings between `AvisRental` and `AlamoRental` is not monotonic. For example, consider the 3-imprecise mapping, in which `RentalNo` is mapped into `PickUpHour`, `PickUpHour` is mapped into `Price`, and `Price` is mapped into `RentalNo`. The similarity measure of this mapping is 0.54. Consider now a 4-imprecise mapping, in which `PickUpLocationCode` is mapped into `PickUp-Date`, `PickUpDate` is mapped into `PickUpMinutes`, `PickUpMinutes` is mapped into `PickUpHour`, and `PickUpHour` is mapped into `PickUpLocation`. The similarity measure of this mapping is 0.55, slightly higher than a 3-imprecise mapping.* □

Example 2 demonstrates how difficult it is to achieve monotonicity even in a toy example. The inherent uncertainty of the matching process generates variability that may cause imprecision sets to overlap in their similarity measures. Indeed, in all our experiments we have never come across such a set of monotonic mappings.

If all one wishes to obtain is the ability to identify the exact mapping through the use of similarity, one needs a weaker notion of monotonicity, as defined next.

Definition 6 *Let $\mathcal{F} = \{F_1, F_2, ..., F_m\}$ be the set of all possible mappings over attribute sets $\mathcal{A}$ and $\mathcal{A}'$. $\mathcal{F}$ is* monotonic with respect to the exact mapping $\bar{F} \in \mathcal{F}$ *if the following inequality holds for any $F_i \in \mathcal{F}$:*

$$Benefit(\bar{F}, F_i) > Cost(\bar{F}, F_i) \quad (2)$$

The set of all possible mappings of the case study, while not being monotonic (see Example 2) is monotonic with respect to the exact mapping. Finally, While one cannot always achieve monotonicity, there may be a general monotonic trend, as formally presented below using statistical terms.

Definition 7 *Let $\mathcal{F} = \{F_1, F_2, ..., F_m\}$ be a set of mappings over attribute sets $\mathcal{A}$ and $\mathcal{A}'$ of cardinality n, and let*

$\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n$ be subsets of $\mathcal{F}$ such that for all $1 \leq i \leq n$, $F \in \mathcal{F}_i$ iff F is i-imprecise. We define M_i to be a random variable, representing the similarity measure of a randomly chosen i-imprecise mapping. $\mathcal{F}$ is statistically monotonic *if the following inequality holds for any $1 \leq i \leq j \leq n$:*

$$E(M_i) > E(M_j) \tag{3}$$

where $E(M)$ stands for the expected value of M.

For statistical monotonicity to hold, one should hypothesize that the similarity measure of a mapping is sensitive to the number of attributes on which the two schemata differ. To evaluate this hypothesis, one needs to examine how similarity measure varies with imprecision level. To do so, a linear regression analysis can be performed, focusing on the variability of the residual values around the regression line. Of special interest are the R^2 and X variable coefficient (the regression line gradient) statistics. The R^2 measure indicates the fraction of the total variability that is explained by the imprecision level. Plainly put, a high R^2 measure means that by separating the set of similarity measures into groups of imprecision levels, different groups have distinguished similarity measures.[4] A positive X variable coefficient is an indication of a positive correlation between imprecision level and similarity measure, while a negative X variable coefficient indicates negative correlation. Combined together, a negative X variable coefficient and a high R^2 measure indicate that imprecision is a major factor in determining the level of μ and that there is an inverse relation between the two. Such an indication is sufficient for ensuring statistical monotonicity.

Figure 1(top) illustrates a linear regression analysis of mapping schemata of two Web sites, namely "Absolute Agency" and "Adult Singles," from the dating and matchmaking domain. For each mapping, the horizontal axis shows the imprecision level of a mapping, while the vertical axis provides the mapping similarity measure. The figure shows strong negative correlation between imprecision level and similarity measure. This conclusion is supported by the R^2 and X variable coefficient of the regression analysis. For this pair of sites, $R^2 = 0.97$, *i.e.*, imprecision level explains, in this case, 97% of the original variability. X variable coefficient is -0.06. Therefore, we have sufficient evidence to claim that statistical monotonicity holds in this case. Figure 1(bottom), illustrates the linear regression analysis of matching "hotels.com" with "usahotelguid.com(holidayinn)," with $R^2 = 0.44$. In this figure, similarity measures in each imprecision level are scattered, rather than being concentrated around the regression line. Therefore, similarity measures of various imprecision levels are interleaved and differentiating the various similarity levels becomes much more difficult.

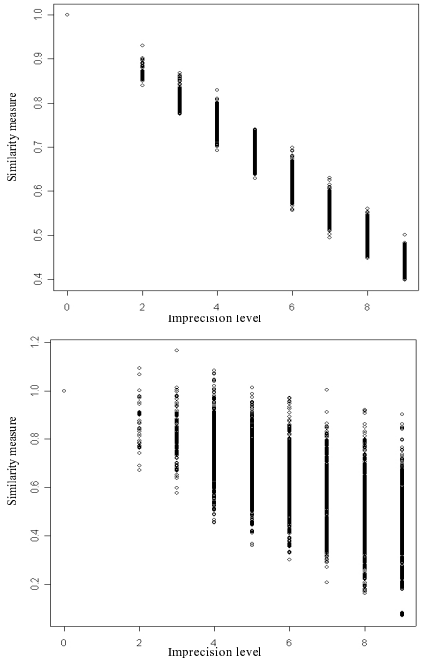

Figure 1: Linear regression graphs

4 Empirical analysis

This section presents initial empirical results, evaluating a matching algorithm using the monotonicity principle. The analysis we propose is aimed at verifying empirically the correlation between a similarity measure (generated by a given algorithm) on the one hand and monotonicity on the other hand, using imprecision level as the experimentation tool. All data sets were collected from real-world Web forms. Due to space constraints, we will refrain from detailing our experiments, readily available in [13], and focus on highlighting the empirical main results.

All experiments were conducted using an inhouse tool named *OntoBuilder*,[5] which supports an array of matching and filtering algorithms. We have selected 36 Web forms, from four different domains, namely flight reservation, hotel reservation, dating and matchmaking, and newspaper search engines. For each Web form, we have automatically extracted a schema, with number of attributes ranging from ten to thirty attributes. Web forms were paired, and for each pair (18 all-in-all) we have applied a *combined algorithm*, combining string matching with domain matching and two structural algorithms. Full discussion of these algorithms is given in [12]. For each Web form pair, we computed all attribute pairwise mappings $\mu^{A,A'}$ and determined the exact mapping $\bar{F}$. We partitioned all possible permutations ($n!$) into imprecision levels with respect to $\bar{F}$. For each permutation we computed $\mu^F = h(\mu^{A,A'} | (A, A') \in F)$, where h is taken to be the $average$ function.

Table 2 summarizes our regression analysis. We have distinguished between high R^2 value (above 0.75), medium R^2 value (0.5-0.75) and low R^2 value (below 0.5). Low values of R^2 indicate that imprecision level explains less than half of the variance in similarity measures. The table shows that

[4] For large data sets, the normal distribution is assumed. R^2 is an indicator to how "close" the data is to the median at each imprecision level. For normal distributions, the median and the mean (the unbiased estimate of the expected value) are the same.

[5] http://www.cs.msstate.edu/~gmodica/Education/OntoBuilder/

R^2	Number of pairs	
0.75-1	8	
0.5-0.75	7	
$<$0.5	3	

Table 2: R^2 distribution

in the vast majority of our experiments the algorithm yielded either medium or high R^2 values. This indicates that the algorithm generates statistically monotonic mappings. That is, the lower the imprecision levels becomes, the further away would a mapping similarity measure be from the exact mapping.

Rank	Number of pairs	
0	13	
1-5	3	
6-99	2	
$>$100	0	
Average rank	4.88	

Table 3: Exact mapping positioning with respect to the best mapping

We next look into the positioning of the exact mapping within an ordered list of all possible mappings. Table 3 summarizes our findings. A rank of 0 means that the algorithm was successful in identifying the exact mapping as the best mapping. Other ranks show the positioning within all possible mappings. We observe that even if an algorithm fails to identify the exact mapping as the best mapping, high ranking of the exact mapping can assist in identifying it within a small number of trials (see [1] for efficient algorithms to identify top-K mappings). However, if one needs to iterate over all possible permutations, searching the search space becomes intractable. Practically speaking, a good algorithm for automatic semantic reconciliation should take into account the inherent uncertainty of the process. Therefore, it should aim at minimizing the number of iterations required for finding an exact mapping, acknowledging that it is probably impossible to identify an algorithm that would always rank the exact mapping first.

For the majority of the experiments the algorithm has positioned the exact mapping with a rank of 0. On the average, the combined algorithm positions the exact mapping between the fourth and fifth positions.

Finally, we analyze the relationship between statistical monotonicity and monotonicity with respect to the exact mapping. A-priori, one may assume that the latter is indifferent to the behavior of permutations, as long as their similarity measure do not exceed that of the exact mapping. In particular, one should not be concerned whether lower imprecision levels demonstrate monotonic behavior. To dispute this assumption we shall hypothesize that there should be no correlation between statistical monotonicity and monotonicity with respect to the exact mapping, and show that our hypothesis is invalid. As a measurement for the former we utilize the R^2 statistic. As for the latter, we count the number of permutations that their similarity measure is "sufficiently close" to the exact mapping (P_c). In our experiments, we have defined this notion of closeness to include all of those permutations whose similarity measure is greater than a threshold (which was chosen to be 0.89μ in our case, where μ is the similarity measure of the exact mapping).

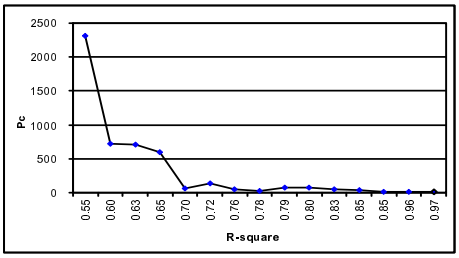

Figure 2: R^2 vs. P_c

Figure 2 provides P_c as a function of R^2 for the combined algorithm. The clear negative trend is testimonial to the invalidity of our hypothesis. Therefore, there is a correlation between statistical monotonicity and monotonicity with respect to the exact mapping. Moreover, from Table 2 it is clear that the combined algorithm typically generates mappings with high R^2, thus the combined algorithm is likely to rank the exact mapping in a top position among all permutations.

5 Conclusion and future work

We have presented the monotonicity principle, a sufficient condition to ensure that exact mapping would be ranked close to the best mapping. We believe this approach offers a useful tool for identifying semantically strong algorithms. From the theoretical and empirical analysis of the model it becomes evident that for monotonic mappings, one may correlate similarity measure with precision, as conceived by a human observer. While monotonicity is a strong notion, weaker notions suffice for practical purposes. Therefore, matching algorithms that generate monotonic mappings (in any form) are well suited for automatic semantic reconciliation.

One possible justification for this correlation has to do with the variance of similarity measures around imprecision levels. The higher R^2 is, the lower the variance becomes. Therefore, less permutations are likely to supersede the exact mapping. This reasoning leads one to believe that the regression line gradient can impact the quality of a matching algorithm, which is something we plan to explore next.

Monotonicity is not defined in "operational" terms, since it is compared to an initially unknown exact mapping. In fact, such an operational definition may not be generally developed, since algorithms may perform well only on some schema pairs. Therefore, a task for future research involves possible classification of application types on which certain algorithms would work better than others. Best mappings may also be subjective at times (less so in the type of applications we were exploring, though). It is not clear at this time how an operational definition can be developed in such cases without personalizing the algorithms to specific human observers. Taken to the extreme, an adaptive algorithm would rank erroneous mappings higher, simply because of a human observer presumptions. This line of research is also left for future investigation.

The recent steps taken in the direction of automating semantic reconciliation highlight the critical need of this research. As the automation of the process has already begun to take shape, often without the benefits of thorough research, the study is timely. We envision multitude of applications of automatic schema matching to the semantic Web. For example, we are currently designing smart agents that negotiate over information goods using schema information and can combat schema heterogeneity.

Acknowledgments

The work of Gal was partially supported by Technion V.P.R. Fund - New York Metropolitan Research Fund, the Ministry of Science, Culture, and Sport in Israel and by the CNR in Italy, and the IBM Faculty Award for 2002/2003 on "Self-Configuration in Autonomic Computing using Knowledge Management." This work is part of an ongoing collaboration with Dr. Danilo Montesi from Università di Camerino, Italy. We thank Adi Luboshitz, Ido Peled, and the class of "Information Systems and Knowledge Engineering Seminar," Fall Semester, 2002, for their assistance in collecting and analyzing the data.

References

[1] A. Anaby-Tavor, A. Gal, and A. Moss. Efficient algorithms for top-k matchings. Submitted for publication. Available upon request from avigal@ie.technion.ac.il, 2003.

[2] J. Berlin and A. Motro. Autoplex: Automated discovery of content for virtual databases. In C. Batini, F. Giunchiglia, P. Giorgini, and M. Mecella, editors, *Cooperative Information Systems, 9th International Conference, CoopIS 2001, Trento, Italy, September 5-7, 2001, Proceedings*, volume 2172 of *Lecture Notes in Computer Science*, pages 108–122. Springer, 2001.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[4] P.A. Bernstein. Generic model management. In C. Batini, F. Giunchiglia, P. Giorgini, and M. Mecella, editors, *Cooperative Information Systems, 9th International Conference, CoopIS 2001, Trento, Italy, September 5-7, 2001, Proceedings*, volume 2172 of *Lecture Notes in Computer Science*, pages 1–6. Springer, 2001.

[5] B. Convent. Unsolvable problems related to the view integration approach. In *Proceedings of the International Conference on Database Theory (ICDT)*, Rome, Italy, September 1986. In *Computer Science'* Vol. 243, G. Goos and J. Hartmanis, Eds. Springer-Verlag, New York, pp. 141-156.

[6] L.S. Davis and N. Roussopoulos. Approximate pattern matching in a pattern database system. *Information systems*, 5(2):107–119, 1980.

[7] L. G. DeMichiel. Resolving database incompatibility: An approach to performing relational operations over mismatched domains. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 1(4):485–493, 1989.

[8] A. Doan, P. Domingos, and A.Y. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In Walid G. Aref, editor, *Proceedings of the ACM-SIGMOD conference on Management of Data (SIGMOD)*, Santa Barbara, California, May 2001. ACM Press.

[9] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proceedings of the eleventh international conference on World Wide Web*, pages 662–673. ACM Press, 2002.

[10] W.B. Frakes and R. Baeza-Yates, editors. *Information Retrieval: Data Structures & Algorithms*. Prentice Hall, Englewood Cliffs, NJ 07632, 1992.

[11] N. Fridman Noy and M.A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 450–455, Austin, TX, 2000.

[12] A. Gal, G. Modica, and H.M. Jamil. Improving web search with automatic ontology matching. Submitted for publication. Available upon request from avigal@ie.technion.ac.il, 2003.

[13] A. Gal, A. Trombetta, A. Anaby-Tavor, and D. Montesi. A framework for evaluating similarity-based schema matching. Submitted for publication. Available upon request from avigal@ie.technion.ac.il, 2003.

[14] Z. Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*, 18(1):23–38, March 1986.

[15] R. Hull. Managing semantic heterogeneity in databases: A theoretical perspective. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 51–61. ACM Press, 1997.

[16] C.A. Knoblock, S. Minton, J.L. Ambite, N. Ashish, I. Muslea, A. Philpot, and S. Tejada. The Ariadne approach to web-based information integration. *International Journal of Cooperative Information Systems (IJCIS)*, 10(1-2):145–169, 2001.

[17] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proceedings of the International conference on very Large Data Bases (VLDB)*, pages 49–58, Rome, Italy, September 2001.

[18] R.J. Miller, L.M. Haas, and M.A. Hernández. Schema mapping as query discovery. In A. El Abbadi, M.L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, and K.-Y. Whang, editors, *Proceedings of the International conference on very Large Data Bases (VLDB)*, pages 77–88. Morgan Kaufmann, 2000.

[19] G. Modica, A. Gal, and H. Jamil. The use of machine-generated ontologies in dynamic information seeking. In C. Batini, F. Giunchiglia, P. Giorgini, and M. Mecella, editors, *Cooperative Information Systems, 9th International Conference, CoopIS 2001, Trento, Italy, September 5-7, 2001, Proceedings*, volume 2172 of *Lecture Notes in Computer Science*, pages 433–448. Springer, 2001.

[20] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.