Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2007)

Computational Intelligence Support for Smart Queries and Adaptive Data

Trevor Martin, Ben Azvine and Yun Shen

***Abstract*— The UK initiative in "Network Enabled Capability" (NEC) can be loosely summarised as "right information, right place, right time - and not too much". It is closely related to the US concept of "network-centric operations". A core requirement for NEC is "active intelligence" where information streams from multiple sources can be dynamically blended and fused into the relevant form for a decision maker. We have identified "smart queries" and "adaptive data" (SQuAD) as the key components of active intelligence, and these form the basis of a current research programme. In this paper we outline three existing research streams that can contribute to the SQuAD project by using soft computing and computational intelligence to merge semi-structured data and taxonomic categories.**

I. Introduction

The UK initiative in "Network Enabled Capability" (NEC) can be loosely summarised as "right information, right place, right time - and not too much". It is closely related to the US concept of "network-centric operations". A core requirement for NEC is "active intelligence" where information streams from multiple sources can be dynamically blended and fused into the relevant form for a decision maker. We have identified "smart queries" and "adaptive data" as the key components of active intelligence, and these form the basis of a current research programme. The main aim of the SQuAD (Smart Queries and Adaptive Data) project is to investigate new methods in soft computing, machine learning and knowledge representation to implement a universal data interface suitable for sources offering little or no formal mark-up. This will be able to integrate multiple sources such as documents, databases and sources which are hidden behind a query / retrieval system. Adaptive data is an encapsulated data source augmented by meta-data giving it

- a self-describing capability
- an ability to configure itself to any format required (including taxonomic re-organisation)

The aim is to enable simpler conversion between different data sources, leading to a more flexible and effective method for integrated information access. Smart query objects interact with adaptive data, and have a capacity to re-write and expand themselves (within specific boundaries) so that they can adapt to new data sources, requests and user feedback. They have an awareness of relevant sources and the resources needed to find answers to a specified degree of accuracy. A smart query can

- automatically fuse data or messages from multiple adaptive sources
- communicate with other smart query objects, estimate their reliability and performance,
- provide "anytime" answers – in general, more accurate answers given more processing time

For example, given multiple information sources such as phone calls, emails, meetings, credit card information, hotel and airline reservations, a smart query could identify individuals and interact with the different sources to extract relevant information and then integrate it to give an overall answer at any point in time. A smart query could detect unusual patterns, raise alarms based on classification of activities and offer explanations based on automatic learning techniques for why a certain activity is placed in a particular class such as "Safe", "Suspicious", "Dangerous" etc.

In this paper, we outline three existing computational intelligence tools [1-4] using soft computing to merge semi-structured data, categorised taxonomically. These tools will be extended and used as the basis for the SQuAD project.

II. Background

Ontologies and sound logical inference rules have been proposed as tools to aid in retrieving, manipulating and combining data from diverse sources. The semantic web is the most high profile example (e.g. see [5]), but there are many other applications in data mining, e-commerce and "mediator" systems which aim to answer a query by combining responses from multiple sources. In cases where the problem domain is sufficiently well-structured and adequate meta-data is available, we would not argue against this position. However, there is a large volume of data (on the surface and deep web, or in stand-alone databases and documents) where either no explicit meta-data is present, or only low-level mark-up is available (e.g. simple XML). Our work is targeted at these sources, where lack of manpower or lack of agreed mark-up should not exclude productive use of data - instead, we need to adopt an alternative approach using computational intelligence to infer the "semantics" of the data as needed.

The SQuAD project is funded by the Defence Technology Centre for Data and Information Fusion. SOFT, iPHI and fuzzy fragment grammars were developed under a Senior BT Fellowship, and applications of iPHI were part of the FP6 ePerSpace project.

Trevor Martin (corresponding author), Artificial Intelligence Group, University of Bristol BS8 1TR UK, trevor.martin@ bris.ac.uk

Ben Azvine, Computational Intelligence Group, Intelligent Systems Laboratory, BT Research and Venturing, Ipswich IP5 3RE UK, ben.azvine@bt.com .

Yun Shen, Artificial Intelligence Group, University of Bristol BS8 1TR UK, yun.shen@ bris.ac.uk.

Semantics is not an exclusive property of formal, logic-based representations. Almost all data stored in computers has "semantics" in the sense that it can be interpreted meaningfully by a human. We aim to use computational intelligence to extract (or at least approximate) aspects of the human-understandable semantic content. We argue that the use of soft computing techniques is vital bridging the gap between human and machine understandable meta-data, as well as in resolving the inherent uncertainty involved in processing knowledge from heterogeneous sources.

The efficient sharing of information requires a common understanding of terms. Natural language is able to convey a large amount of information relatively compactly, without the participants necessarily sharing precisely the same definition of terms. For example, a weather report may state that *"there is a strong chance of heavy snowfall in the south-west of the country"* using several terms that are not precisely defined; nevertheless, there may be sufficient information to decide whether or not to make a journey. Defining terms too precisely can easily lead to confusion, as borderline cases are excluded or included without any indication that they are borderline - to illustrate, let us take "*heavy snowfall*" as meaning "*4 or more inches of snow falling at a specified location within a single day*". Many situations are excluded by this definition - 3 inches of snow on each of two successive days, 3.9 inches in one day, etc. Conversely all cases included by the definition are equivalent, so that 4 inches falling in 24 hours satisfies the definition just as well as 6 inches falling within an hour. It is at this level that we argue for the use of fuzzy sets – to model human understanding of concepts using nested and graded sets. We highlight as a key feature the need to incorporate a representation of uncertainty at fundamental level, in a portable and interpretable manner – to handle inconsistency, absence, reliability, incompleteness.

Depending on the degree of human interpretation and judgment required, we can define a "meta-data spectrum" from informal "understanding-free" approaches involving simple syntactic operations (e.g. keyword matching) to more formal "understanding-rich" approaches, needing considerable human expertise (see also [6, 7]).

Librarians have provided standardised solutions for many years, by analysing and categorising books within carefully designed taxonomies. Clearly this is a very understanding-rich approach, as it requires a human to read or view the material in order to judge exactly where it fits within the taxonomy and how it should be indexed for later retrieval. A user needs knowledge of the subject and the categorisation scheme in order to find relevant sources within a reasonable time. It can be difficult to agree standards for meta-data of this sort, and there may be even less agreement on how the meta-data should describe the content. Even a straightforward schema such as the Dublin Core allows for free text in parts of the description, and hence there is a degree of subjectivity. Although this can be restricted to some degree by use of controlled vocabularies, free text almost inevitably leads to a problem in matching user queries with a data source, or in combining data from more than one source.

Most research is focused at the formal end of this spectrum – because that is where the elegant theorems are to be proved – but most available meta-data is at the informal end. Commercial interests dictate that there is advantage in using XML, as it enables electronic commerce; there is no comparable driving force for creating mark-up in a form suitable for the semantic web.

At the "understanding-free" end of the spectrum, XML enables use of an agreed vocabulary within a community; however, the semantics of this vocabulary is largely dependent on the interpretation of a programmer or user. Different but overlapping vocabularies may differ in their interpretations of terms (e.g. *shipmentDate* could either refer to the time when goods leave a factory or to the time when they arrive at their destination), and may use different terms to refer to the same thing. To take a simple illustration,

www.eccma.org

and

www.eclass-online.com/point

are different "standard" classifications of goods for e-commerce applications. An illustration of the difficulties involved with categorisation can be seen in the following hierarchies, which represent a simple marker pen:

Communication technology, office > Office supplies > writing material > marker pen (office)

Office Equipment and Accessories and Supplies > Office supplies > Writing instruments > markers

where the symbol > represents a superclass-subclass relation. Clearly these hierarchies are similar but not identical - to communicate with both requires a way of mapping one to the other (and preferably mapping both into the classes most convenient for a user). Combining meta-data (and the associated data) is not possible without knowing / learning the mappings between the taxonomies. Such mappings are likely to be approximate – different sources arise from different designers with different world views.

In summary, we can expect meta-data from different sources to have some common elements and some unique elements; also, different sources may use different structure to refer to essentially the same information. Where these differences follow set-theoretic relations (equivalence, subsethood), pure logic is adequate for reasoning. However, in many cases there is no strict set-theoretic relation between classes from different sources and approximate relations must be considered.

III. Overview of Three CI Applications

Most information systems are built (explicitly or implicitly) using crisply defined categories and rely on the underlying theory of databases, which requires a unique identifier for every individual entity.

Within a single database or information source, this can

work adequately (although there are still problems if an object is assigned more than one "unique" identifier). The problem becomes more serious when attempting to combine information from multiple sources - we face the problem of not knowing whether the different sources are referring to the same object. Additionally, data may be in slightly different formats, increasing the problem of matching. This question - when are two entities in an information system the same - is the basis of the "record linkage" problem identified by [8] and formalised by [9]. It remains a problem for information systems [10] as well as the semantic web [11]. "Instance-matching" is the process of determining that objects from different sources are the same - for example, to deduce with a reasonable degree of certainty that an author known in one database as "Lewis Carroll" represents the same individual as the author known in a second database as "C L Dodgson".

The classification structure and attributes (properties) of the objects can be used to guide searching and integration of multiple sources. Even if different hierarchies use different categories, there is likely to be a degree of correspondence, and objects placed within similar categories are likely to have similar properties. For example, a digital library and an online bookseller refer to the same (structured) objects but may differ in categorisation and details stored about each book. This leads to the "ontology alignment" or "schema matching" problem, when different sources classify the same set of objects according to two different hierarchies. Rahm and Bernstein[12] provide a useful survey of approaches to the automation of schema matching, including both database and semantic web problems. They present a taxonomy covering many existing approaches based on the split between metadata matching and content (instance) matching. They also note the problem of *structure matching* where an attribute such as *address* in one source could map to several attributes such as *street*, *town* and *postcode* in another.

We treat information fusion as the integration of material from two or more sources into a single consistent answer. There are three key aspects of this problem - how can we detect additional structure within unstructured or semi-structured data, how can we tell when two sources refer to the same entity, and how can we compare or customise the classifications of that entity? To solve the first problem, we have used evolutionary fuzzy grammars [13] to tag small text fragments. We have proposed the iPHI representation (intelligent Personal Hierarchies of Information) to personalise taxonomic structures, and *SOFT* - Structured Object Fusion Toolkit [14] to determine that objects from different sources are the same (instance-matching). More recent work has shown how *SOFT* can be extended to compare and/or predict the hierarchical classification [15].

In the following sections, we outline these three approaches and give simple applications to illustrate their usefulness. The aim is not to present thorough analysis or extensive results, but to give an overview and show how the components interact.

IV. SOFT INSTANCE MATCHING

We assume two sets of objects (also referred to as instances) $A = \{a_1 \ldots a_n\}$ and $B = \{b_1 \ldots b_m\}$, where we wish to establish an approximate relation

$$h : A \rightarrow B$$

The SOFT method [14] determines which instances are equivalent by comparing their attributes. For example, if sets A and B refer to films, attributes could be *title, director, year* etc.

Let the objects in A and B have attribute values taken from $C_1, C_2, \ldots D_1, D_2, \ldots$ with relations defined as

$$R_i : A \in C_i \quad i=1\ldots n_A$$

$$S_j : B \in D_j \quad j=1\ldots n_B$$

We do not assume that the information about A and B in relations R_i, S_j is identical or completely consistent, but we do assume that some of these relations reflect similar or identical properties of the objects in A and B. Thus for some choices of pairs of codomains (C_i, D_j) we assume an exact or approximate matching function h_{ij} which for each element of C_i returns a (possibly fuzzy) subset of D_j. As shown in [16, 17], this can be converted to a mass assignment giving an estimate of the probability that the element corresponding to some C_i lies in a subset $\{d_1 \ldots d_k\} \in D_j$.

If $R_i(a_k) = C_{ik}$

and $h_{ij}(C_{ik}) = \tilde{D}_{jk}$

where $\tilde{D}_{jk}$ denotes a fuzzy subset of D_j

and $S_j(\tilde{B}_k) = \tilde{D}_{jk}$ using the (inverse) extension principle

then

$$h(a_k) = \tilde{B}_k$$

i.e. a_k corresponds to a fuzzy subset $\tilde{B}_k$. We consider each h_{ij} to give a different "observation" (or sample) of the true value[1], and seek the fuzzy set which is most likely to give these observations.

Let M_n be the mass assignment on B that makes the observed values most likely after n observations, i.e. choose the masses to maximize

$$Pr(M_n | o_1, o_2, \ldots, o_n)$$

This gives a way of updating M after each observation. Using a naive bayes assumption

$$Pr(M_n | o_1, o_2, \ldots, o_n) = \frac{Pr(o_1, o_2, \ldots, o_n | M_n) \times Pr(M_n)}{Pr(o_1, o_2, \ldots, o_n)}$$

$$Pr(o_1, o_2, \ldots, o_n | M_n) = Pr(o_1 | M_n) \times Pr(o_2 | M_n) \times \ldots \times Pr(o_n | M_n)$$

Assuming each possible mass assignment M_n is equally likely,

$$M_n(B_k) = \frac{N_n(B_k)}{\sum_{X \subseteq B} N_n(X)}$$

where $N_n(X)$ is number of times the subset X has been observed.

We need to determine the order in which to apply the "observations" i.e. which of the approximate mappings between attributes is the "best". We need only consider possible pairings Ci∈ Dj where the apparent data types are related; in the work reported here, the user has selected possible pairings although an automated approach is feasible. For each possible pairing, we calculate

$$\frac{\sum_k MAXLPD\left(h_{ij}\left(C_{ik}\right)\right)}{|C_i|}$$

where *MAXLPD(F)* is the maximum probability in the least prejudiced distribution corresponding to the fuzzy set *F*. This measure focuses on the most specific mappings, i.e. those which map elements of C_i onto singleton or small subsets of D_j

A. An Example

The SOFT algorithm has been applied to a number of different semi-structured matching problems, including identification of news stories concerning the same topic [15] and integration of multiple online sources for movie information [2] and classified directories [14]. We focus on the latter problem to illustrate the work in this paper.

We have used two xml datasets describing restaurants (denoted *dbZ* and *dbY* below) to test the method. These are derived from a classified directory and an online source, and have the following structure:

dbZ	Name, TelNo, Addr, FoodType, Meal, TextLine (optional)	295 entries
dbY	name, phone, heading, textline (optional, may be multiple lines)	426 entries

Manual comparison suggests there are about 220 common entries .e.g. from the first source

dbZ43	
Name	pizza hut uk ltd
TelNo	01473 604770
Addr	upper brook st ipswich suffolk ip4 1du
FoodType	pizza european
Meal	lunch dinner
TextLine	pizza

and from the second source

dbY51	
name	pizza hut
phone	216922
heading	food - delivered
textline	pizza hut uk ltd 45-49 upper brook st ipswich

Note that in this case the phone number does not match but these clearly represent the same establishment. To illustrate the problem,

dbY117	
name	pizza hut
phone	214214
heading	food - delivered
textline	pizza hut uk ltd 49 norwich rd, ipswich, ip1 2ep

is almost as similar to *dbZ43* but this time does not correspond to the same establishment. This is obvious to the human observer who is able to extract the embedded address information in the *textline* field. We return to this problem later. The rules selected to match attribute values are straightforward :

Phone ∈ TelNo
EITHER
<*area code*> <*number*> matches <*area code*> <*number*>
OR
<*area code*> <*number*> matches <*number*> and vice versa
OR
<*number1*> partially matches <*number2*> if they differ by permutation of two digits
where degree of match = proportion of digits in the "correct" position

Text strings
EITHER
String *STR1* is an approximate substring of *STR2* if *STR1* is shorter than *STR2* and most words in *STR1* are also in *STR2*
OR
STR1 is an approximate permutation of *STR2* if they have a high proportion of common words
where degree of match = proportion of common words, which must be at least two

Both ignore "stop" words such as *the*, *and*, etc.
This gives reasonable results on well-matched attributes as shown in the next section

B. Results

Mapping from *dbZ* to *dbY*, the average maximum matches between domains are shown in table 1.

On the basis of telephone number matching, 171 entries out of the 220 true matches are found (the true matches are from a manually established ground truth). There are also 20

incorrect matches. When the *name* attributes are also taken into account, these figures improve to 185 correct and 15 incorrect.

Table 1 - best four attribute pairs

dbZ attribute	dbY attribute	average match using max in LPD
TelNo	phone	65%
Name	name	60%
Name	textline	55%
Addr	textline	39%

Matching *Name* with *textline* and *Addr* with *textline* actually decreases the number of correct matches and simultaneously increases the number of incorrect matches. A look at some of the data reveals that the *textline* label is often used to store address and name data as in the following case:

dbX128	
Name	moon & mushroom inn
TelNo	01473 785320
Addr	high rd swilland ipswich suffolk ip6 9lr
FoodType	pubs european english
Meal	dinner breakfast lunch
TextLine	english pub

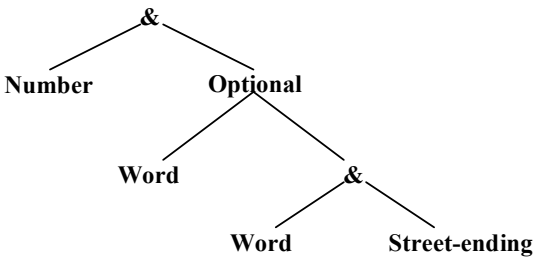
dbY171	
name	half moon inn
phone	785320
textline	half moon inn high rd swilland witnesham
heading	public houses

V. FUZZY GRAMMAR FRAGMENTS

From analysis of this dataset and others involving structured data such as email addresses, URLs, time and date information within unstructured text, it is clear that improved matching performance could be obtained by extracting and tagging the free text sections, then repeating the matching process with the newly discovered "attributes". There are a number of methods for representing string structure. In order to achieve the representation of string structure which had the desired properties of i) possibility of attaching supports to branches on the trees and ii) learnability from sample sets, we have used a simple tree grammar in which a given path from root to leaf represents a possible sentence in the grammar. Tokens in the grammar are simple sets such as *word*, *number* or *alphanumeric;* or known related word sets e.g. *town_names* or *country_names.*

The use of evolutionary methods to find a grammar (or more specifically, a population of grammars) is not new. For example, Smith and Witten [18] defined the usual grammatical classes (noun phrases, verb phrases, etc.) and created AND/OR trees to represent simple grammars. They measured fitness by a combination of the grammar's ability to parse test strings and the inverse of its size. Initial work only included copying and mutation, although this was later extended to include crossover.

We follow a similar approach but use binary trees in which the non-leaf nodes are labelled either as "and" nodes, meaning that both branches must be present, *or* nodes meaning that only one of the branches is needed or "*optional* nodes, meaning that the left branch need not be present. A null terminal is introduced to enable the final symbol to be optional. For example, this tree represents a simple grammar for the first line of an address which could parse *25 acacia avenue* or *101 tyndall park avenue.*



(see also [18])

A. Partial Parsing

Because we are dealing with data that is not always well-formatted and may be noisy, we allow partial parsing of strings. A partial parse is defined as a sequence of tokens that "nearly" matches a grammar - more specifically, for a grammar

$$Gi = T_{1i}\ T_{i2}\ \ldots\ T_{in}$$

and a string

$$S = T_1\ T_2\ \ldots\ T_m$$

where T represents a terminal symbol of the grammar we define the degree of parse as

$$\mu_{Gi}(S) = 1 \mid \min\left|1, \frac{4 \mid L(G_i,S)}{len(G_i)}\right|$$

where $len(G_i)$ is the number of terminal symbols in G_i and $L(G_i,S)$ is the levenshtein distance between G_i and S.

This is a somewhat arbitrary measure which appears to work well in practice, although a full justification is beyond the scope of this paper. Essentially it allows up to a quarter of the grammar to be "wrong" as defined by a substitution, deletion or insertion in calculating the L distance. The use of edit distance allows some tolerance for strings which vary a little from the sample set, but not by much. This allows for some generalization - for example, if the sample set has all its examples in the form

... Suffolk, IP5 3RE

i.e. the postcode comes (correctly) after the county and a sample of the form

…IP5 3RE, Suffolk

is matched to it, we will get a non-zero match.

Work reported in [13] shows an improvement in performance when the generated grammars are used to label the free text (textline) and the matching process is restricted to the tagged address (for matching with *Addr*) or the unlabelled component (for matching with *Name*).

VI. COMBINING MULTIPLE TAXONOMIES - IPHI

The idea of an intelligent Personal Hierarchy for Information (iPHI) is to configure access to multiple sources of information based on personal categories. Frequently, data sources are organised explicitly or implicitly according to an internal taxonomy. Large AI projects in knowledge representation e.g. [19] have shown that it is impossible to create a single unified taxonomy. "Mediator" systems which aim to answer a query by combining responses from multiple sources form an important area of current research. Several tools have been proposed to aid in the automation of this problem, and were surveyed from various perspectives by Rahm and Bernstein [12] They present a taxonomy covering many existing approaches based on the split between meta-data matching and content (instance) matching. The need for uncertainty has also been noted by others. For example, Chang and García-Molina [20] developed an approach for the precise translation of Boolean queries across different information sources. In a subsequent paper [21] they presented a real-world case study (combining book searches from four web sites), and found that it was only possible to make exact mappings in 30% of the rules, while 70% required approximation.

To take a very simple example, the set of tracks or albums classified in one online music store as

music > rock > classic rock > 70's classics

may correspond to another's

music > rock&pop oldies

Music and film genres are an easily-understood and readily accessible application for soft category mapping - Aucouturier. and Pachet [22] state that

> *music genre is an ill-defined notion, that is not founded on any intrinsic property of the music, but rather depends on cultural extrinsic habits.*

They also comment that the intensional and extensional definitions of music genres frequently do not coincide, and review various approaches to the representation and assignment of musical genre. They divide approaches into manual and automatic, with the latter further split into approaches which use various features of the music, typically extracted from signal processing, to decide class membership, and approaches which are based on measuring similarity to other tracks in a class. Similarly [23] looked at the automatic creation of document genre in text libraries, also on the basis of measurable properties of the document. We take the view that genre (and in our view, most other useful hierarchical classifications) are subjective; at the same time, they are a very useful tool for organising and retrieving instances in a collection. Although it is probably impossible to rigorously derive genre on the basis of simple features, the work reported here shows that a useful approximate classification method can be created.

Here, we use the instance matching method to find correspondences between hierarchies, when instances are classified according to different categorisations. We use the equivalence of instances from different sources to learn a soft mapping between categories in these hierarchies, allowing us to compare the hierarchical classification of instances as well as their attributes. Such correspondences may in turn be used to improve the identification of equivalent instances.

In general, we consider two sets of instances *A* and *B* with corresponding sets of labels *LA* and *LB* each of which has a hierarchical structure i.e. there is a partial order defined on the labels. Note that this does not imply that a hierarchy induces a partition on the instances - it may be that "orthogonal" attributes such as "date" and "type of storytelling" are represented within the same hierarchy.

Each label $li \in$ LA denotes a subset of A i.e. we have a denotation function

$$den : L_A \rightarrow A$$

such that

$$l_i > l_j \subseteq den(l_j) \subseteq den(l_i)$$

(and similarly for B)

For example, if A and B are sets of films then LA and LB could be genres such as western, action, thriller, romance, etc. To illustrate, let

$A = \{a1, a2, a3, a4, a5, a6\}$

and

$B = \{b1, b2, b3, b4, b5\}$

with genres defined as

$den(Horror_A) = \{a1, a2, a3\}$

$den(Animation_A) = \{a5, a6\}$

$den(Suspense_B) = \{b1, b2, b3\}$

$den(Childrens_B) = \{b4, b5\}$

Note that these can easily be fuzzy rather than crisp. We use the SOFT method to derive a soft mapping on the sets of entities *A* and *B*

$$h : A \rightarrow \tilde{P}(B)$$

(where $\tilde{P}(B)$ is the set of all fuzzy subsets of *B*)

In our illustrative example, assume *h* is

$h(a1) = \{b1/1, b2/0.2\}$

$h(a2) = \{b1/1\}$

$h(a3) = \{b2/1, b3/0.5\}$

$h(a4) = \{b1/1, b2/0.9, b3/1\}$

$h(a5) = \{b4/1, b5/0.8\}$

$h(a6) = \{b1/0.3, b4/0.9, b5/1\}$

where the notation *element / membership* is used in fuzzy sets. This relation states that the film *a1*in source *A* almost certainly corresponds to the film *b1* in source *B* or (with much lower possibility) to film *b2* in source *B*. It is used to determine a (soft) correspondence between any pair of labels *li* and *lj* from the label sets *LA* and *LB*

$$g_c : L_A \to L_B$$

Given a label $li \in LA$ we consider its denotation *den(li)* under the mapping *h* and compare it to the denotation of $lj \in LB$. In the ideal case if the two labels are equivalent,

$$h\left(den\left(l_i\right)\right) = den\left(l_j\right)$$

Given that *h* is approximate and that the correspondence between labels may not be exact, we use semantic unification to compare the sets.

$$\Pr\left(l_i \to l_j\right) = \Pr\left(h\left(den\left(l_i\right)\right) = den\left(l_j\right)\right)$$

This gives an interval-valued conditional probability which expresses the relation between a pair of labels; we then extract the most likely pair to give a crisp relation

$$g_c : L_A \to L_B$$

In the illustration,

$$\begin{aligned}&\Pr\left(Horror_A \to Suspense_B\right)\\ &= \Pr\left(h\left(\{a1,a2,a3\}\right) \mid \{b1,b2,b3\}\right)\\ &= \Pr\left(\{b1/1,b2/1,b3/0.5\} \mid \{b1/1,b2/1,b3/1\}\right)\\ &= \left[0.5,1\right]\end{aligned}$$

This is straightforward to calculate. Clearly a "real" case involves many more instances than this, but the example suffices to show the principles involved.

1) Application to Film Databases

The two film websites "rotten tomatoes" and the internet movie data-base (IMDb) are "user-maintained" datasets which aim to catalogue movie information . The databases denoted dbT and dbI below are derived from these sites, respectively containing 94,500 and 94,176 film records, and were used in experiments. Since dbT and dbI are produced by two different movie web sites, there is inevitable "noise" existing in the film data; i.e. different tag sets, different genre names and missing elements. A typical example of instances that correspond to each other yet are not syntactically identical is as follows :

dbI	
Title	Gentleman B.
Year	2000
Directed_by	Jordan Alan
Genre	Thriller
Aka	Gentleman Bandit, The (2000) (USA: MIFED title)
Country	USA

dbT	
Title	Gentleman Bandit
Year	2000
Director	Jordan Alan
Genre	Dramas, Film Noir, Blackmail
MPAA_rating	NOT Rated
Cast	Ed Lauter, Peter Greene, Justine Miceli, Ryan O'Neal,

In order to match attributes, the same simple string matching functions as were used as in the previous section, ignoring "stop" words such as *the*, *and*, etc. We note also that it is possible to obtain better results for people's names (attributes such as cast, director, etc) by using fragment grammars to extract first name and surname and then matching individuals on that basis.

The average matches between domains are given in Table 2 - this is used in the SOFT method to determine which attribute pairs to use in identifying equivalent instances.

Table 2. Average degree of match between attributes in *dbI* and *dbT*

dbI attributes	dbT attributes	average match
Year	Year	100%
Title	Title	41%
Directed_by	Director	27%
Aka	Title	21%

On the basis of the three best attributes, the system identified movies from dbI dated 1976-1990 which were also in dbT, and compared the genre classification. The similarity threshold between two film records was set to 0.5 giving a total of 14,124 movies which are found to be identical.

The similarity between two genres is relatively hard to decide from simple string matching. For example, *"animation"* is not similar to *"children"* from the point of view of text matching, but the extension of the sets of films in these two categories shows considerable overlap. Some examples of intuitively reasonable genre mappings are listed in Table 3.

Table 3. Examples of matching genre pairs determined by the system

dbT genre	dbI genre
Animation	Children
Comedy	Drama
Horror	Suspense
Sci-fi	Fantasy

2) Results on Unseen Data

The attribute and genre mappings were applied to a new set of 24,839 entries from *dbI* (calendar years 2000-05), trying to find matches in *dbT*. For comparison, a manually produced ground truth established 1274 true matches – this

figure is low due to the relatively large number of entries for TV series, "foreign" movies etc in *dbI* which are not included in *dbT*. Using the SOFT algorithm without genre mapping, we find 861 pairs of matching film entries when the similarity threshold between two films is set to 0.44. With the presence of the ground truth, 261 film matching pairs out of 382 film pairs in 2000 are missing, 102 out of 364 in 2001 are missing, 87 out of 330 in 2002 are missing, 60 out of 142 in 2003 are missing, and 3 out of 8 in 2004 are missing. This represents a recall of 67 % and a precision of 100%. Incorporating the genre mapping as well produces a much better (100%) recall, at the expense of loss in precision.

VII. SUMMARY

We have outlined three components for integrating data sources by finding additional structure, identifying different representations of the same instance, and using taxonomic categories to enhance that identification. and given simple applications to illustrate their usefulness. The aim is not to present thorough analysis or extensive results, but to give an overview and show how the components can contribute to the goal of creating adaptive data and smart queries.

REFERENCES

[1] Martin, T. P. and B. Azvine, "Acquisition of Soft Taxonomies for Intelligent Personal Hierarchies and the Soft Semantic Web," *BT Technology Journal*, vol. 21, pp. 113-122, 2003.

[2] Martin, T. P. and Y. Shen, "Improving access to multimedia using multi-source hierarchical meta-data," in *Adaptive Multimedia Retrieval: User, Context, and Feedback*, vol. LNCS vol 3877, *LNCS*: Springer, 2006, pp. 266 - 278.

[3] Martin, T. P., "Fuzzy sets in the fight against digital obesity," *Fuzzy Sets and Systems*, vol. 156, pp. 411-417, 2005.

[4] Martin, T. P. and B. Azvine, "Evolution of Fuzzy Grammars to aid Instance Matching," presented at 2006 IEEE International Symposium on Evolving Fuzzy Systems, Ambleside, UK, 2006.

[5] Shadbolt, N. R., N. Gibbins, H. Glaser, S. Harris, and M. C. Schraefel, "Walking through CS AKTive Space: a demonstration of an integrated Semantic Web application," *Web Semantics*, vol. 1, pp. 415-419, 2004.

[6] Sheth, A., C. Ramakrishanan, and C. Thomas, "Semantics for the Semantic Web: The Implicit, the Formal and the Powerful," *Int Journal on Semantic Web and Information Systems*, vol. 1, pp. 1 - 18, 2005.

[7] Uschold, M., "Where Are the Semantics in the Semantic Web?," *Ai Magazine*, vol. 24, pp. 25-36, 2003.

[8] Newcombe, H. B., J. M. Kennedy, S. J. Axford, and A. P. James, "Automatic Linkage of Vital Records," *Science*, vol. 130, pp. 954-959, 1959.

[9] Fellegi, I. P. and A. B. Sunter, "A Theory for Record Linkage," *J. American Statistical Assoc*, vol. 64, pp. 1183-1210, 1969.

[10] Dey, D., S. Sarkar, and P. De, "A Distance-Based Approach to Entity Reconciliation in Heterogeneous Databases," *Ieee Transactions on Knowledge and Data Engineering*, vol. 14, pp. 567-582, 2002.

[11] Dou, D., D. McDermott, and P. Qi, "Ontology Translation on the Semantic Web," *Lecture Notes in Computer Science*, pp. 952-969, 2003.

[12] Rahm, E. and P. A. Bernstein, "A Survey of Approaches to Automatic Schema Matching," *The VLDB Journal*, vol. 10, pp. 334-350, 2001.

[13] Martin, T. P. and B. Azvine, "Evolution of Fuzzy Grammars to aid Instance Matching," presented at EFS-06, Lancaster, UK, 2006.

[14] Martin, T. P. and B. Azvine, "Soft Integration of Information with Semantic Gaps," in *Fuzzy Logic and the Semantic Web*, E. Sanchez, Ed.: Elsevier, 2005.

[15] Martin, T. P. and Y. Shen, "Soft Mapping between Hierarchical Classifications " presented at IPMU-06, Paris, France, 2006.

[16] Baldwin, J. F., "The Management of Fuzzy and Probabilistic Uncertainties for Knowledge Based Systems," in *Encyclopedia of AI*, S. A. Shapiro, Ed., 2nd ed: John Wiley, 1992, pp. 528-537.

[17] Baldwin, J. F., T. P. Martin, and B. W. Pilsworth, *FRIL - Fuzzy and Evidential Reasoning in AI*. U.K.: Research Studies Press (John Wiley), 1995.

[18] Smith, T. C. and I. H. Witten, "Learning language using genetic algorithms," *Lecture Notes in Computer Science*, vol. 1040, pp. 132-145, 1996.

[19] Lenat, D. B., "CYC: A Large-Scale Investment in Knowledge Infrastructure," *Communications- Acm*, vol. 38, pp. 32, 1995.

[20] Chang, C. C. K. and H. Garcia-Molina, "Approximate Query Translation Across Heterogeneous Information Sources (Research)," presented at Very large data bases, Cairo, 2000.

[21] Chang, C. C. K. and H. Garcia-Molina, "Approximate Query Mapping: Accounting for Translation Closeness. ," *VLDB Journal*, vol. 10, pp. 155 - 181, 2001.

[22] Aucouturier, J. J. and F. Pachet, "Representing Musical Genre: A State of the Art," *Journal of New Music Research*, vol. 32, pp. 83-94, 2003.

[23] Rauber, A. and A. Mueller-Koegler, "Integrating Automatic Genre Analysis into Digital Libraries," presented at Digital libraries, Roanoke, VA, 2001.