Similarity Assessment Techniques

Michael Zang, Adam Gray, and Michael Hobbs

CDM Technologies, Inc
2975 McMillan Avenue, Suite 272
San Luís Obispo, CA 93401

Abstract

This paper serves as a layman's introduction to similarity assessment techniques, their distinction from contemporary computing paradigms, and their real-world applications. The following content derives from the experience gained by the authors in applying similarity assessment techniques to resolve the real-world problems of CDM customers, in particular those faced by the United States Transportation Command (USTRANSCOM).

Keywords

Agents, Case-Based Reasoning, Data Cleansing, Data Mapping, Search, Similarity Assessment

Introduction

Study of case-based reasoning (CBR) served as our initial introduction into similarity assessment strategies. CBR is a two-part reasoning process whereby (1) similarity assessment techniques are employed to compare a new, current case against a case base of previously stored cases to find the most similar case in the case base, which is (2) subsequently used to classify and resolve the current case. Case-based reasoning projects such as the Collaborative Agent-Based Control and Help system (COACH)[1] and the Navy Conversational Decision Aid Environment (NaCoDAE) (Breslow and Aha 1998) refactoring project[2] led us to appreciate the applicability of the first step of case-based reasoning (i.e., similarity assessment) to a variety of problems. Having identified the CBR-derived similarity assessment as a broadly applicable problem-solution paradigm, we applied it, with success, to a number of difficult real-world problems, thus initiating a compelling new growth platform for the company.

This paper first describes the paradigm by which most contemporary software-based problem solutions are derived—namely, Boolean logic, shown to be distinct from similarity assessment as a problem-solution paradigm. Similarity assessment is discussed within the context of its heritage, case-based reasoning. Examples of distinct similarity assessment techniques—word, trigram, numeric, vicinal, and mixed-initiative—and similarity assessment applications—search, mapping, and data cleansing—are provided to further illustrate the concept. The paper ends with a summarizing conclusion.

[1] COACH, a research project sponsored by the Office of Naval research from 1999 to 2001, employs case-based reasoning technology to provide analysis, evaluation, and the formulation of guidance for major equipment item repairs aboard US Navy ships.

[2] NaCoDAE refactoring was performed under a collaborative research agreement (CRADA) with the Navy Center for Applied Research in Artificial Intelligence.

Boolean Logic

Boolean logic is a complete system for operations in symbolic, mathematical logic named after its inventor, George Boole, the 19th-century mathematician, who uncovered the algebraic structure of deductive logic, the basis of computer hardware and software. In Boolean logic, elements each contain only two possible values, following various conventions, such as "true" and "false", "yes" and "no", "on" and "off", or "1" and "0". A Boolean expression, such as "X < Y And Y< Z" evaluates to true or false. Relational databases use Boolean logic to perform queries. For example in standard relational query languages a statement such as: "SELECT * FROM EMPLOYEES WHERE (Not LAST_NAME = 'Zang') And (FIRST_NAME = 'Mike' Or 'Michael')", is used to return records from a database. Search engines (e.g., Google) also employ Boolean logic. For example "Search term 1" "Search term 2" is equivalent to "Search term 1 Or Search term 2".

Special characters, such as truncation and wildcard, provide a higher level of abstraction than the Boolean logic which underlies the operations implementing them. Truncation, '*', allows for search using a shortened (i.e., truncated) form of a word. For example, "adolescen*" will return both "adolescent" and "adolescence". Wild card characters, '?', prove useful in allowing for alternate spellings and other quirks of the English language. For example, "wom?n" will return results for both "women" and "woman." These special characters provide a higher level abstraction than straight Boolean Logic. Note for example, "adolescen*" is not the same as the Boolean expression "adolescent And adolescence" as it may return other perhaps unforeseen values such as "adolescents."

Regular Expressions further the abstraction by providing a language of special characters for identifying strings of text of interest, such as particular characters, words, or patterns of characters. A relatively simple Regular Expression can, for instance, identify the word "car" in isolation or when preceded by the word "blue" or "red", while another can recognize a dollar sign immediately followed by one or more digits, followed, optionally, by a period and exactly two more digits, thus accounting for dollars and cents. While special characters and regular expressions can prove useful during focused searches, processing them across the entire database may consume excessive computer resources (Strickland and Henderson 2005).

Similarity Assessment

Similarity assessment techniques inherently concern objects that share a common set of features to varying degrees. Well-established techniques have been developed in the fields of machine learning—a branch of artificial intelligence—and statistical pattern recognition (Michie et al. 1994). Example approaches include neural networks, support vector machines, decision-tree induction, Bayesian, and case-based classification techniques.

Neural networks consist of interconnected, biologically inspired processors called neurodes that can learn to classify by pre-classified examples. Support vector machines use an optimization technique to find planes that best separate objects into distinguishing classes. Decision-tree induction algorithms typically employ a measure called information gain to select the most promising features, construct a decision tree, and use it to classify new objects. Bayesian classification techniques apply estimations of class-conditional probabilities to predict a label for

a new case or object. Case-based classification reuses the decisions from the closest matching pre-classified objects to label new objects.

The applicability and classification performances of these techniques depend on the representation language of the objects and the amount of available pre-classified data. For example, neural networks and support vector machines work well with many instances of numerical data. In contrast, case-based techniques are well-suited for a mixture of data types with relatively few examples.

Case-based techniques utilize similarity assessment to classify new objects. Classification refers to the task of assigning one or more predefined labels to a previously unlabelled object. Case-based classification works as follows. For a new object or a case to be labeled, similarity assessment techniques are utilized to retrieve the most closely matching previously labeled cases from a database of cases, called a case base, to assign the label from the retrieved cases as the label for the new object (Kibler and Aha 1988).

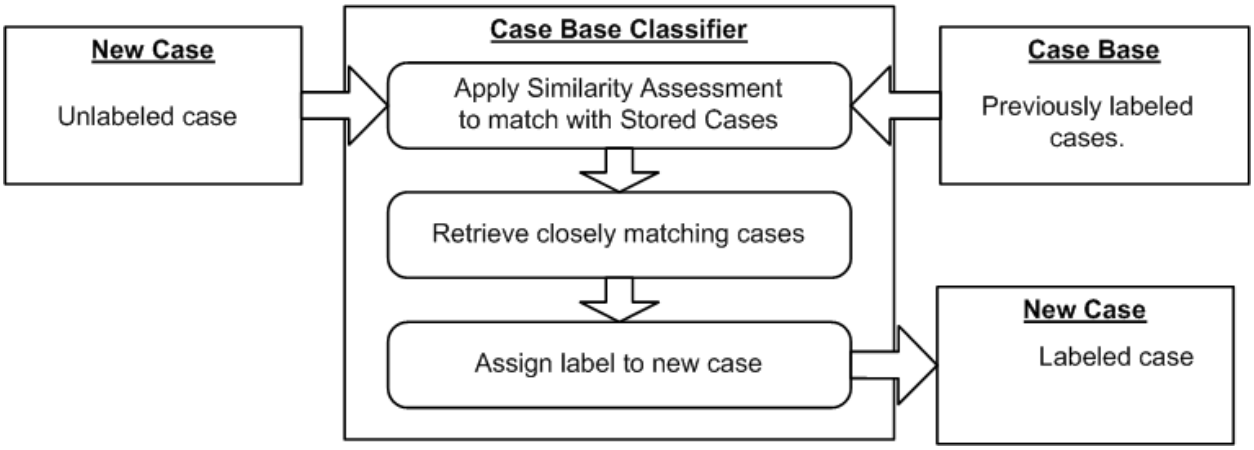

Figure 1: Case-Based Classification

Classification performance depends significantly on two design factors, the case representation and the similarity metric.

> ***Case Representation***: A case is a structured representation of the factors to be used for assessing similarity between two cases. The most common case representation is a list of attributes and values.
>
> ***Similarity Metric***: A similarity metric is an aggregation function that assigns a number between 0 and 1 as a measure of similarity between two cases. A similarity value of 1 implies that the two cases are identical while a similarity value of 0 implies that they are completely distinct.

Examples of similarity or distance metrics include the Euclidean metric, cosine metric, and Hamming distance. Such metrics often specify parameters such as attribute weights to improve classification performance. Nonetheless, when a large number of features are associated with a case base, some prove irrelevant and can reduce classification performance. While counteractive parameters can be set manually, automatic methods can potentially achieve the same result. For

this purpose, several attribute weighting and feature selection methods, such as information gain and rough-set theoretic methods (Gupta et al. 2005), are available.

Word Similarity

The principal problem when comparing the similarity of words is the breadth of naming differences for word-associated concepts. Causes for such differences are described by the taxonomy in Figure 2. Name variation can arise for syntactic or semantic causes. Syntactic distinctions engender commonly used vocabulary (e.g., "code" vs. "id"), conventions (e.g., "Airport_Code" vs. "AirportCode"), and abbreviations (e.g., "Airport" vs. "Arpt"). Semantic distinctions occur as differences in abstraction (e.g., "Ship" vs. "Vessel") and granularity (e.g., "Name" vs. "First Name" and "Last Name").

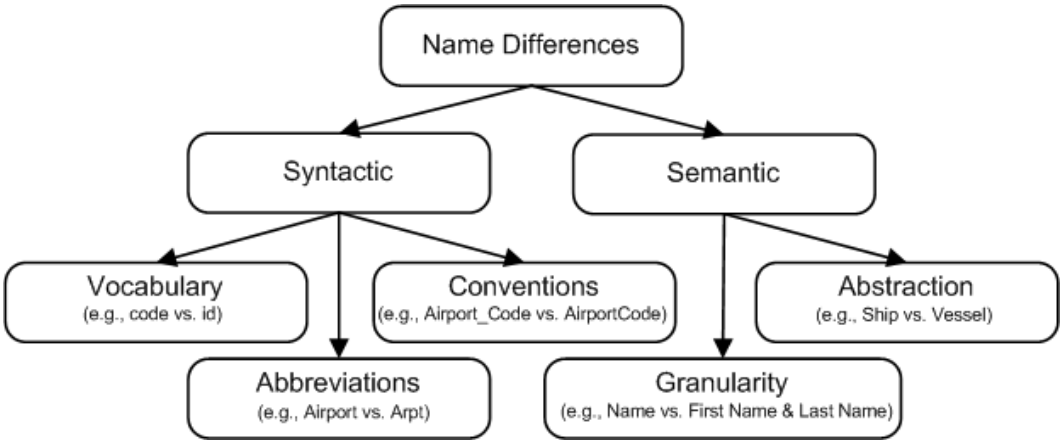

Figure 2: Taxonomy of Naming Differences

Many techniques have been developed to address such terminological variation. CDM has employed the following approaches to aid in detecting the conceptual equivalence of two distinct terms.

- ***Creation of a synonymous terms lexicon***, by exploiting the mapping that has already been performed. For example, if the field label "Ship Id" has been manually mapped to "Vessel Code" it can be interactively identified that "Ship" is synonymous with "Vessel" while "code" is synonymous with "identifier". Such a resource can be used directly for similarity assessment.
- ***Use of WordNet*** (Fellbaum 1998), a publicly available linguistic ontology, to identify occurrences of terminological variations due to conceptual abstraction. For example, the hyponym (is-a-type-of) relation between concepts (e.g., Ship is-a-type-of Vessel) may be exploited as part of the similarity assessment.
- ***Creation of an abbreviation resource***, by exploiting the correspondence of data model logical and physical names. The logical names that include non-abbreviated terms (e.g., Airport) and their corresponding physical names that include abbreviated terms (e.g., ARPT) can be used to automatically create a lexicon of abbreviations for use in similarity assessment.

- ***Employment of an order-based abbreviation detection algorithm***, which exploits the characteristic of abbreviations to preserve the relative ordering of the original word. For example, the letter "r" occurs after the "p" in the word "airport." A search utilizing this algorithm will, for instance, return "ARPT," a desired result, while excluding instances of "APRT" (i.e., apartment), an irrelevant, if similar abbreviation. Regularity in abbreviation conventions can be exploited to detect abbreviations dynamically when they are not available in the abbreviations lexicon.

Synonym and abbreviation resources may be employed with a technique commonly known as Bag-of-Words to assess the similarity between textual data. Here the number of primary words in common divided by the number of unique words provides a measure of similarity. Algorithms can exploit a synonym resource to maximize the number of common words. Synonyms may have an associated weighting factor denoting the semantic distance between the word pair. For example, the concept of the word "vessel" is more general than that of "ship," which is more general than that of "oil tanker." To account for variation in semantic abstraction, the synonym pair (vessel, ship) may have a weighting factor of 0.9 and the pair (vessel, oil tanker) a 0.8. The example in Figure 3 shows application of this technique to determine that the expression, "The vessel is arriving" is more similar to "The ship is coming" than "The airplane is coming".

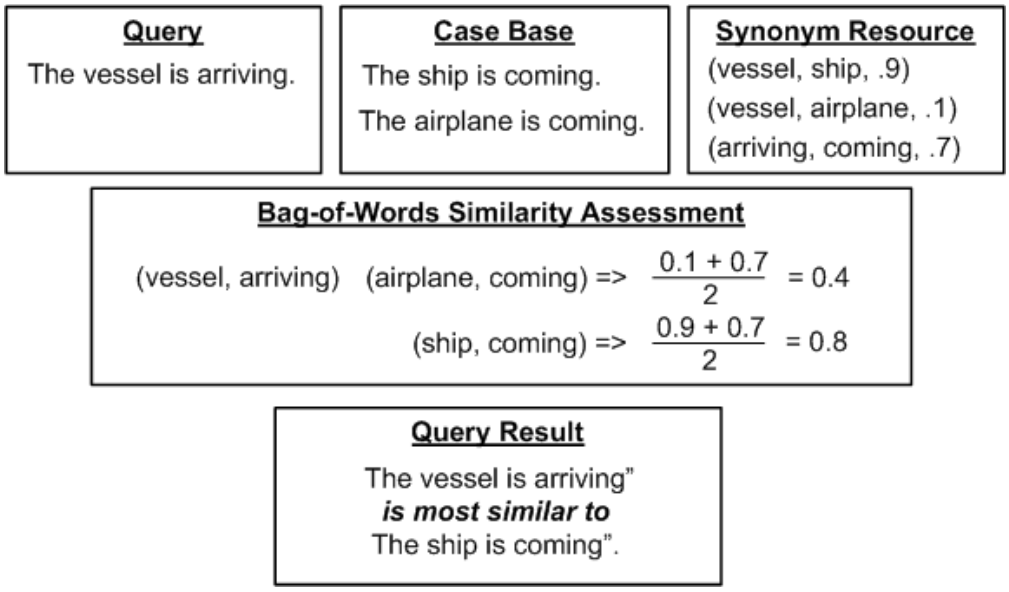

Figure 3: Word Similarity Query Example

Trigram Similarity

Trigrams represent a specific instance of the more general N-gram concept by which text is broken down into all-composing three letter contiguous sequences in order to compare for similarity with other text. N-grams, in turn, represent one of many techniques for producing similarity metrics for what is known in the industry as case-based classification. N-grams are particularly tolerant of spelling variations and misspellings and indeed serve as the underlying technology for contemporary spell-checkers, which present a list of correctly spelled words that

may correspond to a word flagged as incorrectly spelled (i.e., not found in the dictionary of words).

As an example, assume a mechanic wishes to query an intelligent parts database for a pressure gage. The semi-literate mechanic queries the system with the phrase "presure gage." This query generates the 11 unique trigrams shown in Figure 4. Note that the beginning and ends of words are padded with spaces to emphasize their importance within a word.

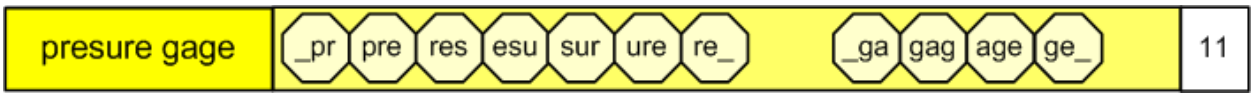

Figure 4: Query Phrase Trigrams

Further assume that the case base (i.e., the parts database employed in a similarity assessment context) contains a pressure gauge, garlic press, and box wrench. The trigrams for these items are shown in Figure 5. The number of occurrences of a particular trigram within the case base—1 or 2 in this example—is shown below each case base item trigram. Note that the trigrams "_pr," "_ga," "pre," and "res" occur in both "pressure gauge" and "garlic press" (i.e., twice as often as do any other trigrams). Statistical analysis of the number of occurrences of each trigram allows them to be weighted for rarity. For example, the twice-occurring trigrams may be weighted at .5 while the single occurring trigrams are weighted at 1. These weights are employed when summing like trigrams.

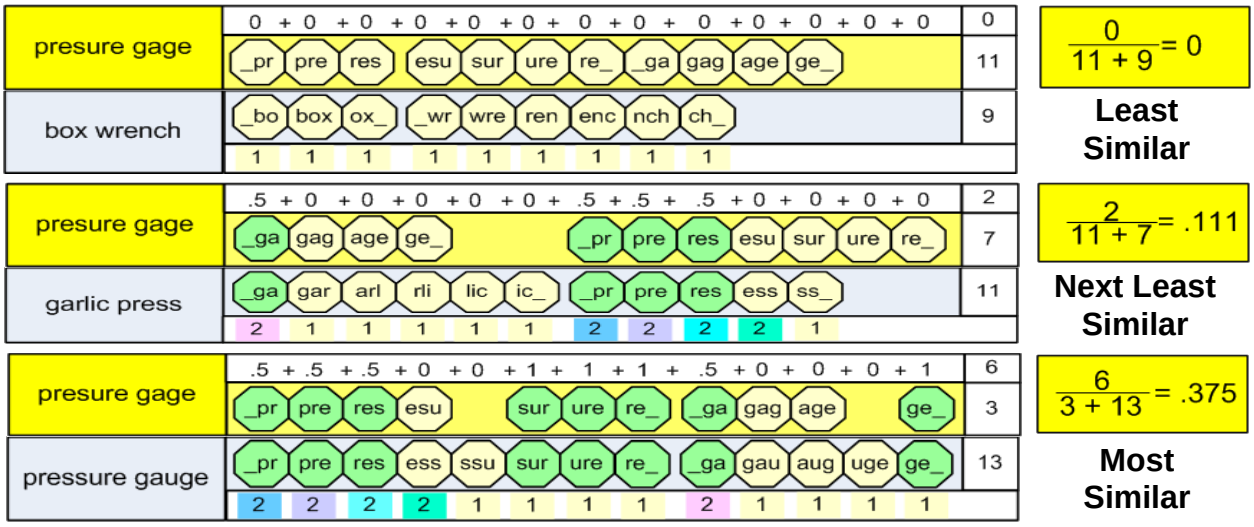

Figure 5: Trigram Similarity Query Example

In the example, the query text "presure gage" shares 0 trigrams with "box wrench. With "garlic press," "presure gage" shares 4 trigrams each weighted at .5 due to their commonality. This produces of score of 4 times 0.5 divided by 18—the total number of unique trigrams contained in both, which equals 0.111. Comparing "presure gage" to pressure gauge produces a score of 0.375. The individual comparison scores are only meaningful in relation to other scores. In this example the case base item "pressure gauge" is clearly the most similar to the query "presure gage" as its score of 0.375 has a 54% difference from the next highest score of .111 for "garlic press."

Numeric Similarity

The similarity assessment of numeric values, including quantitative values (e.g., weights and ages) and displacement values (e.g., heights and lengths), pose inherent difficulties for deriving appropriate similarity scores. Consider the comparison of the weight of a 20 pound dog to that of an 11 pound cat. When attempting to determine the appropriate similarity score, it becomes clear that additional information is needed. If the range of weights of all animals being compared runs from 5 pounds to 20 pounds, then the weights of the dog and cat are relatively dissimilar (i.e., their similarity score should be close to 0). However, if the list of animals includes a 100-ton blue whale, then, relatively speaking, the weights of the dog and cat are quite similar (i.e., their similarity score should approximate 1). Hence, the calculated similarity score should be different in these two cases.

For this reason it is critical that the variance of values across the entire comparison domain be considered when determining the similarity of any two elements. CDM's approach, specified by Equation 1, calculates the similarity between the two numeric values relative to the difference between the maximum and minimum values across the entire domain.

$$Similarity(X,Y) = \frac{((MAX_{val} - MIN_{val}) - |X_{val} - Y_{val}|)^{LOG\_MOD}}{(MAX_{val} - MIN_{val})}$$

Equation 1: Assessing Similarity of Quantitative Values

An additional difficulty arises when comparing values within a domain with a large variance. Taking our weight example, if the 100-ton whale is included in the comparison, a five pound difference in weight leads to only a .00002 difference in similarity score. This makes the calculated difference for a significant number of comparisons negligible (i.e., the difference in weight between the dog and the cat would have almost no effect on similarity score). To counter this issue, CDM uses a logarithmic modifier (LOG_MOD) to accentuate the differences at the lower end of the scale.

Vicinal Similarity

Another valuable technique, vicinal similarity comparison, targets the relative distance between locations (i.e., determines if two locations are in the same vicinity). This type of comparison faces an inherent difficulty in that geographic locations are often represented in multiple ways and include information from multiple fields (e.g., latitudinal and longitudinal values are combined). Hence, geographic information must be translated into a comparable format before similarity calculations can be attempted. CDM's approach is to take the provided format (e.g., latitude/longitude, geospatial, etc.) and convert it into vector [x,y,z] coordinates on the earth. The physical distance between two points is then calculated by determining the angle between the two vectors and scaling the results based on the radius of the earth.

Once a physical distance between two points has been calculated, the problem is essentially one of Numeric similarity, described in the previous section. The appropriate similarity score to assign the results again largely depends on the variance between the locations across the entire domain. For example, when comparing distances between locations in a single zip code, then the locations of two different cities in that zip code might be relatively dissimilar (i.e., their similarity score should be close to 0). However, if you are comparing locations across the entire

world, those same two cities' locations will be, relatively speaking, quite similar (i.e., Their similarity score should approximate 1). The approach taken by CDM, as when handling numeric comparisons, is to determine the similarity between two locations by comparing their physical distance relative to the largest physical distance between two locations across the entire comparison domain. Once again, a logarithmic modifier is again used to magnify the differences at the lower end of the scale when comparing domains with a large variance.

A related vicinal technique compares geographic areas, such as zip codes or countries, rather than specific points on the earth. CDM handles this type of comparison by assigning a specific point on the earth to represent the estimated center of each geographical area. Once this single point has been assigned for all areas within the case representation, similarity assessment can continue as described above.

Mixed-Initiative Assessments

Our experience with similarity assessment techniques, as with artificial intelligence paradigms in general, clearly indicates that specific techniques can perform exceptionally well with one data set and exceptionally poorly with others. All such techniques operate based on pre-established (i.e., built-in) assumptions, thus limiting the utility of any single approach as a solution to every case. The arbitrary or inconsistent performance of single assessment techniques can be counteracted by utilizing the weighted average of a number of distinct techniques to provide a single measure of similarity. Weighting factors allow techniques to be turned off (i.e., weight = 0), turned on (i.e., weight =1), or set to have less influence (i.e., 0 < weight < 1) depending on their domain performance individually and/or in conjunction with other techniques. This approach was shown to greatly improve the performance of the data mapping application (Gupta et. al. 2008) described later in this paper. Note, however, that the mapping performance gained by the mixed-initiative approach comes at the expense of computational performance. While computational performance is not as much of an issue for an application that can run in the background and post results when complete, it can present a problem for one which requires real-time user interaction, such as with an internet search or database query.

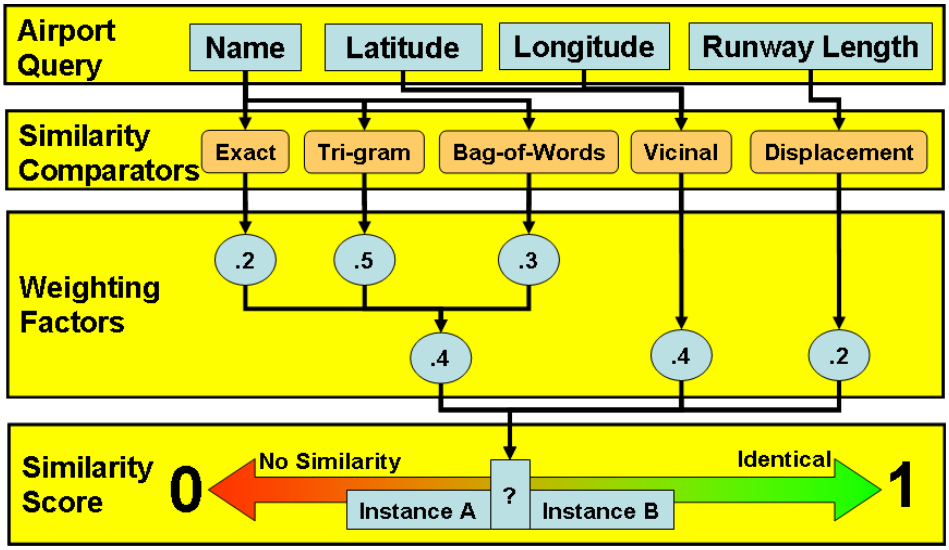

Figure 6: Mixed Initiative Assessment

Figure 6 depicts a mixed-initiative assessment used to correlate airport records contained in two distinct database tables, each standard to a specific domain, Commercial and Military. Both tables contain fields corresponding to name, latitude, longitude, and runway length. The naming conventions, descriptiveness, and data quality, however, vary both across and within the two databases. For example, LAX may be named Los Angeles International, LA Airport, or LAX CIV Runway 1. Geographic locations (i.e., Lat/Long) and runway lengths can be missing, dramatically erroneous, close, or precise. In the example, Names are compared using exact match weighted at .2, Trigram weighted at .5, and Bag-of-Words weighted at .3. The resulting assessment score is combined, with weight .4, with a vicinal comparison of the latitude and longitude, with weight .4, and a numeric comparison of runway length weighted at .2, to produce an overall similarity score.

Search Applications

The data quality and interoperability issues faced by USTRANSCOM are closely associated with reference data (RD) which captures the relatively static information that defines and categorizes enterprise-associated representational entities, thus specifying the universe of content that can be referenced by program-specific data to provide externally meaningful semantic context. As such, a substantial portion of the information exchanged between automated information systems (AISs) is comprised of RD codes which serve as unique identifiers for individual records within a specific reference data set. Contemporary practice results in a significant percentage of exchanged RD code values being rendered invalid or contextually improper, which results in interoperability issues among systems. Such issues introduce operational inefficiencies within the enterprise while degrading the quality of its composite information state, upon which critical business decisions are based. This problem is particularly prevalent in regards to very large and dynamic RD data sets such as National Stock Numbers (NSN), DoD Activity Address Codes (DODAACs), and Geographic Locations (GEOLOCs).

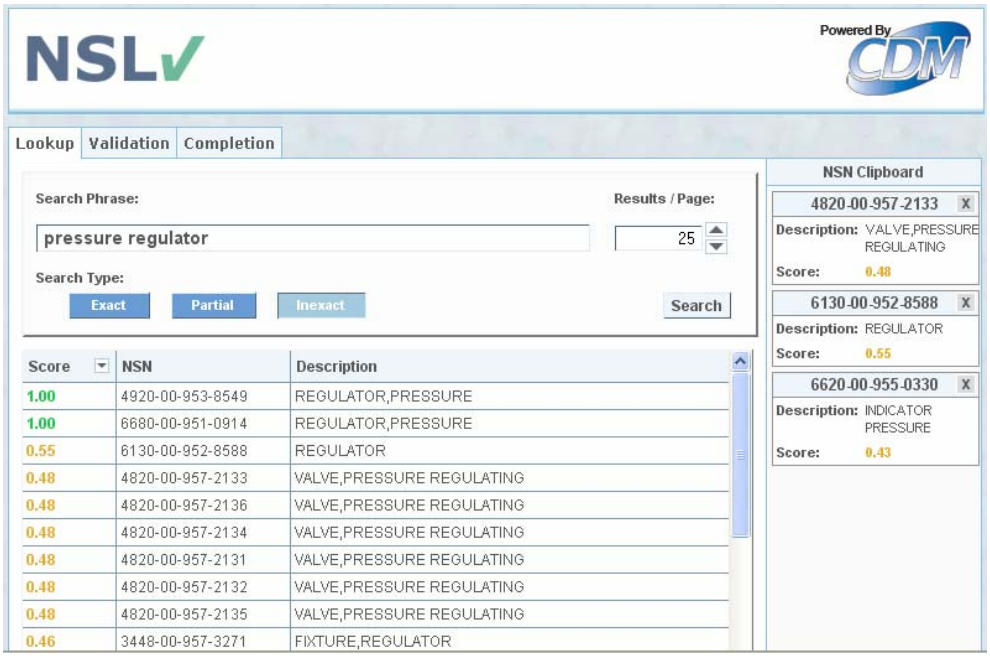

Figure 7: National Stock Number Lookup and Validation Tool

The users and agents of enterprise information systems require intelligent runtime access to RD sets in order to obtain the RD-identifying key codes or associated item characteristic data necessary to perform their system-specific IT tasks. Consider the problem of obtaining the National Stock Number—a 13-character code identifying one of the 8.6 million DoD supply items—to fill in a requisite field on a data form. Knowing what one has or wants does not necessarily provide the code, due to nomenclature variation, imprecision, and the limitations of contemporary database query technology. To confront this problem, CDM developed the National Stock Number Lookup and Validation Tool (NSLV)[3] to assist users in finding National Stock Numbers (NSNs) using free-form textual descriptions of desired items. NSLV employs the trigram similarity assessment technique described in the Figure 5 example. The NSLV screen shot (Figure 7) displays the ranked results of a user query for a pressure regulator.

Mapping Applications

The joint deployment and distribution responsibilities of USTRANSCOM require a substantial level of interoperability across the broad range of technically and functionally diverse automated information systems (AISs) utilized by USTRANSCOM and the individual service branches, as well as the associated commercial suppliers and shippers. This aspect of the USTRANSCOM mission prompted the development of an enterprise-wide data model, known as the Master Model (MM) and a formalized program for the identification, management, and distribution of enterprise reference data, known as the TRANSCOM Reference Data Management (TRDM) program.

The MM enables interoperability at the metadata (i.e., database schema) level while TRDM enables interoperability at the instance-data (i.e., database record) level. However, the human-intensive (thus costly and error prone) development and maintenance of semantic maps is required for these capital investments to provide for increased interoperability levels. At the metadata level, these semantic maps relate elements within AIS-specific data models and interface specifications to elements within the MM. At the instance-data level, they join equivalent or semantically related records (e.g., airport to geographic location records as correlated by latitude and longitude fields).

The Intelligent Mapping Toolkit (IMT)[4] is designed as a set of intelligent collaborative tools to support professional analysts performing labor- and knowledge-intensive semantic mapping tasks within a dynamically evolving information infrastructure. IMT employs a federation of matching agents for case-based similarity assessment and learning. IMT semi-automatically acquires domain-specific lexicons and thesauri to improve its mapping performance. It also provides an explanation capability for mixed-initiative mapping. IMT's primary goal is to suggest mappings to users for final verification and acceptance (Gupta, et. al. 08).

[3] NSLV was developed in the context of an analysis of the USTRANSCOM Reference Data Management program (TRDM) under contract to USTRANSCOM J6, 2006 – 2007.
[4] Sponsored by USTRANSCOM J6, 2005 - 2007

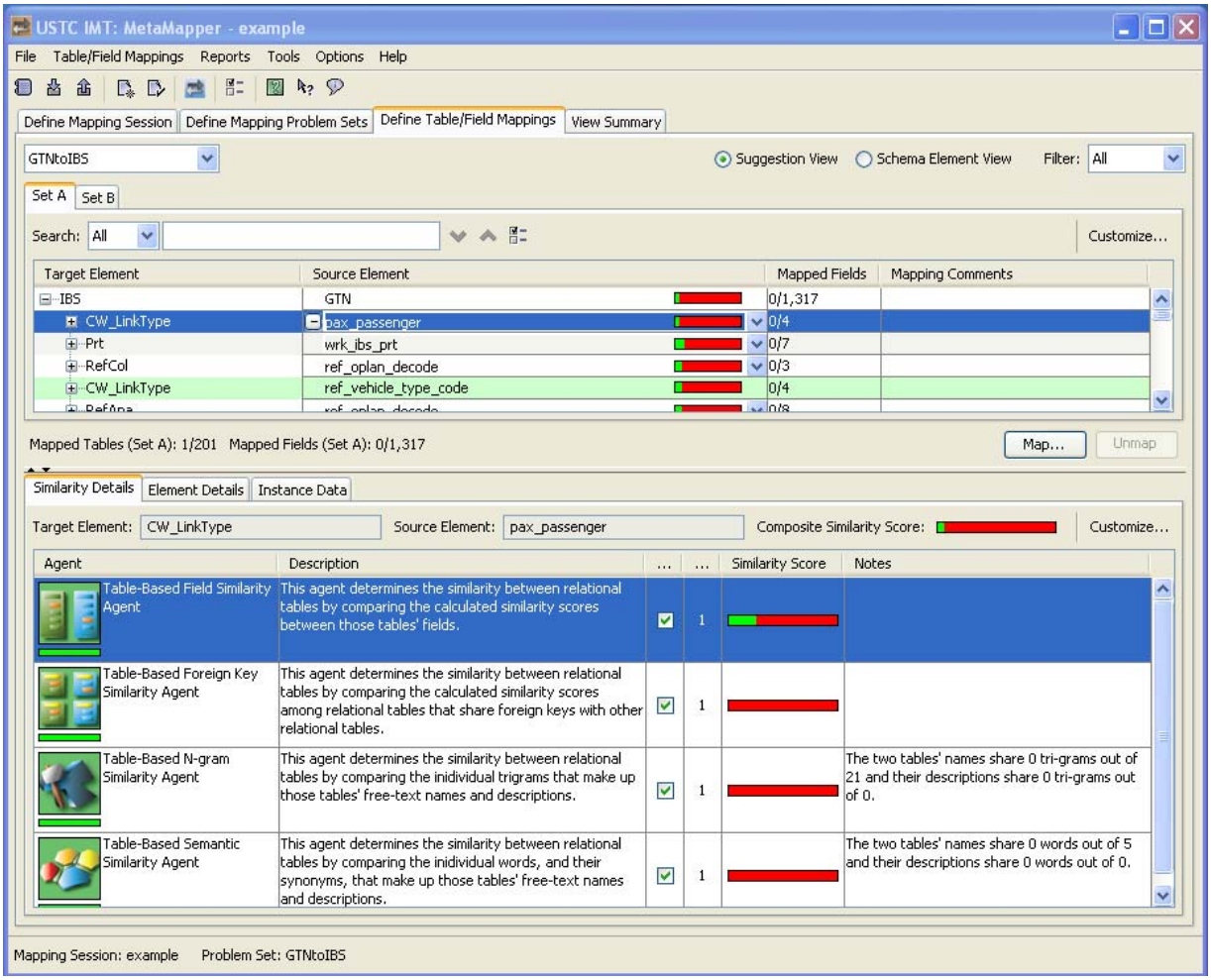

Figure 8: The Intelligent Mapping Toolkit

Data Cleansing Applications

The United States Central Command (USCENTCOM) Top 100 Analysis process provides visibility to the heaviest sustainment items (identified by National Stock Number [NSN]) from two continental United States aerial ports of embarkation—Charleston Air Force Base (AFB), South Carolina, and Dover AFB, Delaware—to the USCENTCOM area of responsibility (AOR) each month. This information is used to support shifting the mode of transportation of the heaviest, most often airlifted commodities to surface transportation, in order to realize significant transportation cost avoidance and greater efficiency for the Department of Defense. The current process is predominantly manual and requires excessive time to analyze the data, especially to overcome data quality issues. The most pressing of these data quality issues fall into four categories: missing item shipping weights, incorrect item shipping weights, missing NSNs, and missing item names.

The Intelligent Data Analysis Application (IDAA)[5] is designed to detect problematic data within imported queries of shipped cargo items and provide support to users in resolving them. Using a

[5] Sponsored by USTRANSCOM J6, 2007 - 2008

database of nearly three million approved cargo types, IDAA matches cargo items by National Stock Number (NSN), item name, weight, and cube. For those items that cannot be definitively matched, suggested cargo types are automatically generated using intelligent similarity-based data comparators.

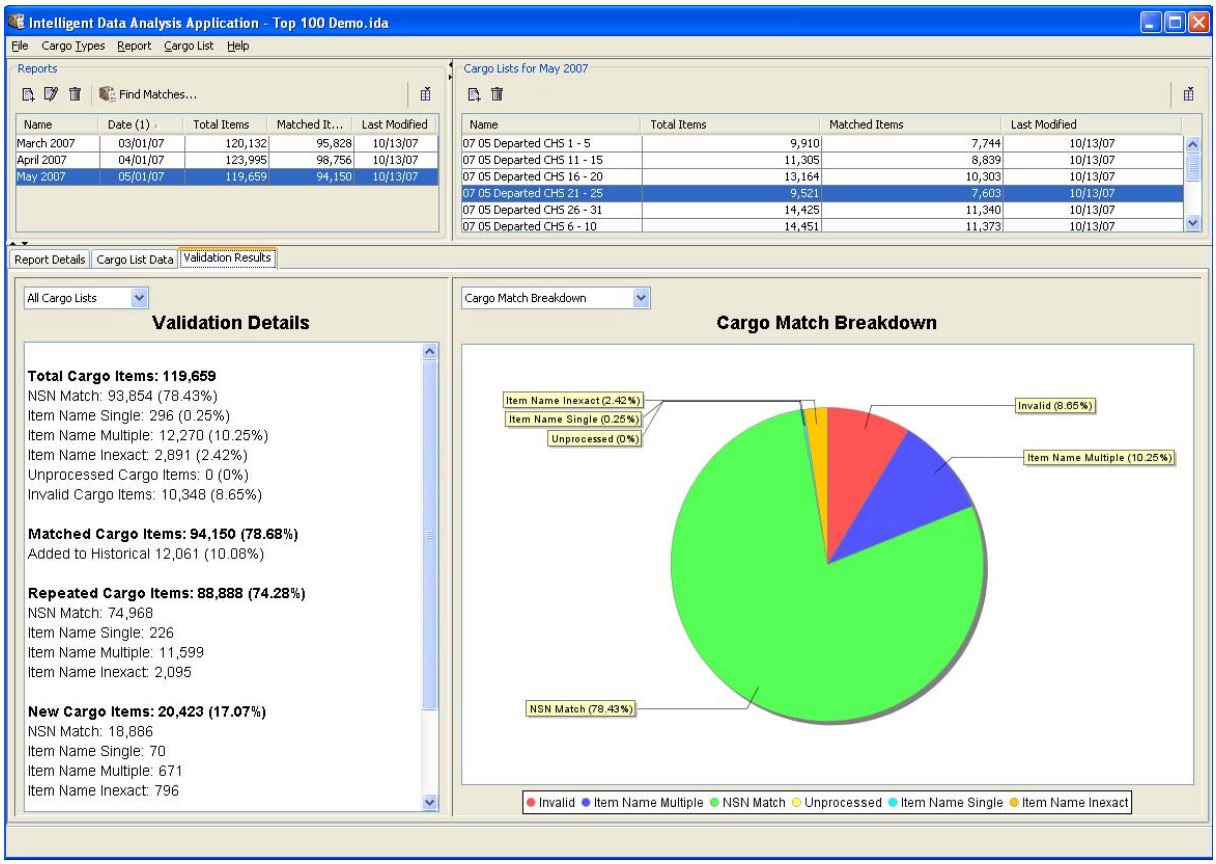

Figure 9: The Intelligent Data Analysis Tool

Conclusion

Similarity assessment techniques provide a unique solution approach applicable to a broad range of problems. They prove to be intuitive to both users and developers, perhaps due to similarities with human problem-solving approaches. Contemporary paradigms are based directly upon Boolean logic in which elements may either be True (1) or False (0). Similarity assessment paradigms provide an analog paradigm which allows elements to have values between 1 and 0 as well. Similarity assessment is derived from the field of case-base reasoning and is particularly effective when employed in conjunction with users to bring pertinent or desired items to their attention in ranked order. Various techniques are available for assessing similarity of textual and numeric data, although performance is strongly dependent on the application domain. This problem can be overcome by combining the results of multiple techniques to produce a single assessment result, given the acceptability in the corresponding loss of computational performance. Similarity assessment techniques have been successfully applied in a variety of search, mapping, and data cleansing applications previously irresolvable employing Boolean logic approaches alone.

References

Breslow, L.A. and Aha, D. W; NaCoDAE: Navy Conversational Decision Aids Environment (1998), Navy Center for Applied Research in Artificial Intelligence, Code 5510, Washington, D.C., December 1998

CDM Technologies, Inc. (2006A); Intelligent Mapping Tool [IMT] Design and Development Report, Task Area C: Intelligent Data Mapping Deliverable; USTRANSCOM J6 contract; 29 September 2006.

Fellbaum, C. (1998); WordNet: An Electronic Lexical Database. MIT Press.

Gupta, K.M.; Moore, P.G.; Aha, D.W.; and Pal, S.K. (2005); Rough-Set Feature Selection Methods for Case-Based Categorization of Text Documents, in S. K. Pal, S. Bandyopadhyay, & S. Biswas [Eds.]; Lecture Notes in Computer Science [LNCS 3776], (pp. 792-798); Heidelberg, Germany: Springer.

Gupta, K.M; Zang, M.; Gray, A.; Aha, D. W.; and Kriege, J. (2008); Enabling the Interoperability of Large-Scale Legacy Systems, Proceedings of the IAAI-08 conference, Emergent Application or Methodologies Papers; Menlo Park, CA. July 2008.

Kibler, D. & Aha, D.W. (1988); Case-Based Classification. E. Rissland & J.A. King [Eds.], Proceedings of AAAI Workshop on Case-Based Reasoning, (pp.62-67); St. Paul, MN.

Michie, D.; Spiegelhalter, D.J.; and Taylor, C.C. [Eds.] (1994); Machine Learning, Neural and Statistical Classification; New York, NY: Ellis Horwood.

Strickland, Jennifer and Henderson, John R. Web page maintained by: Library Webmaster, Ithaca College Library. http://www.ithaca.edu/library/course/expert.html, last modified: 10 October 2005.

USTRANSCOM Corporate Information-Centric Environment [CICE] (2003); USTRANSCOM Architecture and Technical Integration Division Report, prepared by CDM Technologies, Inc. USTRANSCOM TCJ6-A Scott Air Force Base, Illinois; October 2003.