Resolving Keyword Abbreviation Heterogeneity Using Gene Matching

Chris Fernandes
Northwestern Univ./Dept. of EECS
Evanston, Illinois, 60201, USA
Phone: +11-708-467-2299
Fax: +11-708-491-4455
E-mail: cfernand@eecs.nwu.edu

Wen-Syan Li †
Northwestern Univ./Dept. of EECS
Evanston, Illinois, 60201, USA
Phone: +11-708-467-2299
Fax: +11-708-491-4455
E-mail: acura@eecs.nwu.edu

1 Introduction

As information continues to proliferate in networks throughout the world, the need to access specific portions of data from disparate sources becomes increasingly more important. We see three types of information sources stored in digital libraries: media (audio and video), textual and HTML documents, and database-stored information. The methods to integrate these different types of information sources vary due to the nature of the information presented, which is usually heterogeneous. Media information is integrated via separate display windows. (e.g. MPEG player or SHOWPS). Textual documents can be integrated by converting to HTML formats or abstracts. Database-stored information can be integrated through heterogeneous database semantic integration techniques. It is within this latter domain that we intend to identify a problem and propose a possible solution.

2 The Problem

One problem inherent to integrating information from separate autonomous databases is that the information in one database may be represented in a vastly different form from another database, even if the same information is being represented. Data conflicts may manifest themselves in the form of naming, scaling differences, scope, or level of abstraction. The former is representative of a particular class of problems which we will be concentrating on: the problem of *naming*.

Synonyms and homonyms are examples of this problem. Synonyms occur when objects with different names represent the same concepts, and homonyms occur when different concepts have the same name.

We approach the task of the semantic integration of data with respect to another type of naming conflict: the problem of abbreviations. This

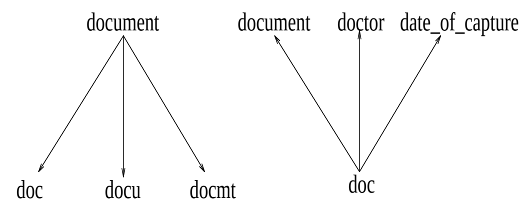

Figure 1: The Abbreviation Problem

type of conf lict occurs in database attribute naming and in traditional and digital libraries. The problem stems from the fact that humans often concoct their own abbreviations which may or may not be constructed from the letters in the original word. In addition, there will always be words for which there is no standard method of abbreviation, and the administrators who maintain the databases of the world are, more often than not, apt to choose differing abbreviations for the same word rather than construct one from a globally-defined technique. This is illustrated in Figure 1, where the word *document* may have many abbreviations, and, conversely, one of *document*'s possible abbreviations, *doc*, may actually stand for a number of different concepts.

3 Related Work

Most of the existing work focuses on building lexical dictionaries or further constructing ontological spaces to model the real world concepts. [2] argues that knowledge of the domain context is important in understanding the meaning of the terms. Once the domain is identified, it can also be used to resolve ambiguous terms, since terms are typically well defined within a domain. Understanding the meaning of a term can be much improved by greater use of standard ontologies, computing-intensive analysis, and AI-guided term matching.

The proposition by [1] can be used for the retrieval of closely related domains. It is composed of a *thesaurus* that contains representative

†This material is based upon work supported by the NSF under Grant No. CCR-9210704.

keywords which describe a facet (including synonyms) and a *conceptual graph* in which the leaves are terms, and the nodes are supertypes that denote general concepts relating two or more terms.

4 Technique

Our solution to the abbreviation problem involves deriving all plausible abbreviations for a given word as well as being able to reassemble an original word from a particular instance of abbreviation, as is suggested in Figure 2. The word *element* may have three "visually pleasing" methods of abbreviating it: *elem*, *elmt*, and *elt*. One would like to be able to derive these abbreviations from *element* as well as be able to construct the original word after seeing any subset of the above abbreviations. Also, if a synonym exists for the original word, such as *item* in Figure 2, one would also want to be able to obtain this connection from the original word *element*. In Figure 2, abbreviations for *element* could be deemed semantically equivalent to various abbreviations for *item*.

We applied a gene-matching algorithm to tackle this problem by comparing pairs of words for similarity in structure. Developed by [3] for a biological environment, this algorithm was originally designed to compare strings of genetic material. Since two strands of DNA need not be identical to perform the same function, the algorithm allowed mismatched nucleotides to be within a "small" distance away from its predetermined position without compromising a match. Thus, a random substitution, insertion, and/or deletion of a nucleotide from a particular genetic sequence could be smoothed out during the comparison.

In the same way, abbreviations could be thought of as being semantically equivalent to the original words they represent, having only been adjusted by certain deletions and substitutions of letters from the original word. By comparing the relative ordering of the letters within the abbreviation and the location of the mismatched letter within the original word, the distance between the abbreviated word's letter location and the original word's letter location can be quantified. Thus, while the algorithm was designed to be a nucleotide-by-nucleotide comparison of genes, we have adapted it into a letter-by-letter comparison of words. Figure 3 illustrates how we use this algorithm to derive three abbreviations of *element* which are similar to an abbreviation of *item*.

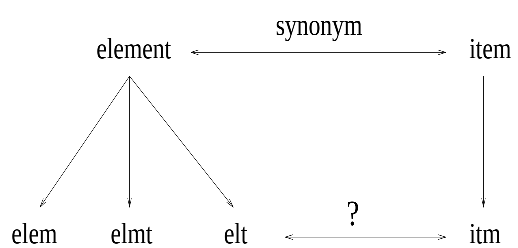

Figure 2: Possible Abbreviations of *element* and Related Concepts

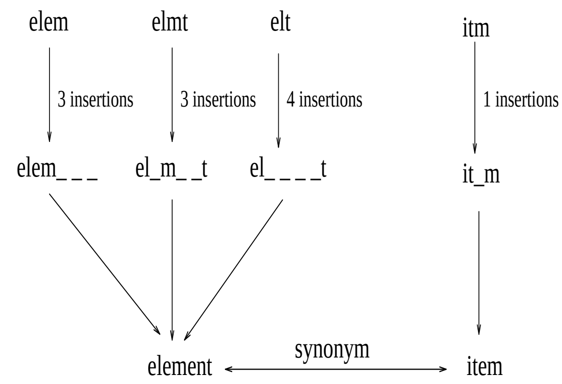

Figure 3: Resolving Keyword Abbreviation Heterogeneity Using Gene Matching

5 Current Status

Our implementation uses a theorem of distance based upon viewing the letters of a word as nodes in a *distance matrix* and then performing a path analysis to determine how far each letter is away from the others. The algorithm determines the probability of equivalence between two words by computing the cost of transforming one word into the other via a series of insertions, deletions, and substitutions. When comparing a word to its possible abbreviation, insertions and deletions cost less than substitutions since the former are more apt to be be involved in the relation between an abbreviation and its original word than the latter.

These operations are represented as lines in our distance matrix, and a traversal can reveal a minimal transformation cost. By performing this over a dictionary of possible abbreviations, a prime candidate can be found. Thus, under the assumption that abbreviations tend to use a subset of the same letters of the original word and in the same order as the original word, the system is able to determine the likelihood of one word being the abbreviation of another, and thus possibly being semantically equivalent.

References

[1] R. Prieto-Diaz. Classification of reusable modules. *IEEE Software*, 4(1):6–16, 1987.

[2] Gio Wiederhold. Digital libraries, value, and productivity. *CACM*, pp. 85–96, April 1995.

[3] E. Tyler, M. Horton, and P. Krause. A Review of Algorithms for Molecular Sequence Comparison. *Computers and Biomedical Research*, 24:72-96, May 1991.