Determining Relationships among Names in Heterogeneous Databases

Clement Yu and Biao Jia
Department of EECS
University of Illinois at Chicago
Chicago, IL 60680

Wei Sun
School of Computer Science
Florida International University
Miami, FL 33172

Son Dao
Research Lab.
Hughes Aircraft Company
Malibu, CA 90265

1 Introduction

Determining relationships among names has been identified as an essential task in schema integration in fully integrated heterogeneous database systems as well as in the interoperability of federated database systems [BrHu91, Demi89, Lars89, Shet89, Kim89, SiMa89]. The relationships of interest include IS-A relationships, aggregations, homonyms, and synonyms. This is done manually and co-operatively by many database administrators (DBAs).

There have been a few approaches to address the determination of attribute relationships semi-automatically for database schema integration [BrHu91, Demi89, Lars89, Shet89]. In [YSDK91], we have proposed that the semantics of names (attributes/entities) can be captured by a set of "common concepts", i.e. we make use of sets of concepts to represent the semantics of names. By comparing the concept sets of different names, we may establish relationships among these names.More specifically, our method uses concept hierarchies. Concept hierarchies are formed such that concepts on the top are either more general than concepts on the bottom of the hierarchy or are composed from lower level concepts. A similarity function is used to measure the semantic closeness of two names. Both [BrHu91]'s and [YSDK91]'s method assume that the mapping from names to term hierarchies or common concepts is done manually by DBAs. We describe briefly here an automatic mapping process, a process to recognize potential relationships among different names and a process to expand knowledge in our knowledge base so as to facilitate the first two processes.

2 Knowledge Base Description

The system which will be used to determine relationships among names of database objects (attributes and entities) has a knowledge base (KB) which contains concepts, relationships among concepts, synonyms, stoplist and rules for differentiating different meanings of certain keywords. The stoplist contains all words that are non-contents words.

Initially, some common concepts and part of their relationships can be established manually.The initial concepts and their relationships need not be complete to capture semantics of all database names. During the system execution, the system will find certain unknown concepts and possible relationships among them through interaction with the users. This makes the KB expand and eventually contain enough information to reveal most potential relationships among names of attributes or entities.

3 Relationship determination process

Our approach of determining relationships among names includes the following steps: (1) Map names to a set of common concepts by their descriptions automatically and at the same time, create new concepts whenever necessary; (2) Calculate the similarity of each pair of names and arrange the name pairs in descending order of similarity; (3) For each related name pair, ask the DBA to confirm the relationship. Depending on the outcome of the DBA's answer, certain specific relationships among concepts are considered likely by the system. These relationships are then presented to the DBA for verification. In this way, knowledge in the knowledge base will be expanded. The above 3 steps will be repeated until no modification is made to concepts and their relationships and no more possible related name pairs can be found.

(1) Map names to common concepts

For each word which appears in the description of a name, it can be an unimportant word if it appears in the stoplist, or a keyword if it exists in the sets of the describing words associated with the concepts stored in the KB or a potential concept, if it does not appear in the stoplist nor the sets given above. In the first case, the word will not be used to characterize the name. Examples of this type include words like "the","of" etc. In the second case, the keyword is mapped into the concept, which contains the keyword as one of its describing words. In the third case, the potential concept may be made to become a concept or may be added to the stoplist, depending on the outcome in the feedback process(3).

(2) Calculate similarity

Once all names have been mapped to common concepts in KB, we can determine relationships among them by comparing the relationships among their mapped concepts. To do this, we make use of some similarity functions [Salt89, YSDK91]. The degree of closeness between two names can be measured by the probability that two names are related by certain relationships given a set of concepts in common between two names, together with the set of concepts in one name which are related to the concepts in the other through is-a relationships and aggregate relationships.

(3) Gain knowledge by user feedback

After certain name pairs are determined by the system to have sufficiently high similarities, each such pair of names are suspected to be synonyms or related by a is-a relationship or an aggregate relationship, depending on the relationships among their mapping concepts. Such a pair is presented to the user to verify whether the relationship is correct. This is needed because the extraction of keywords from the description of a name may cause a loss of information and sometimes the description does not contain enough information to capture the semantics of the name. Furthermore, the mapping from a keyword to a concept may be inaccurate due to the fact that a keyword may have multiple meanings and its semantics may be influenced by the presence of other words.

If the determined relationship between the two names A_i and A_j is verified by the user to be correct, then some relationships can be expected to exist between the concepts in A_i but not in A_j, denoted by $C(A_i - A_j)$, and those in A_j but not A_i, denoted by $C(A_j - A_i)$. As an example, consider the situation that A_i and A_j are synonyms. The fact that $C(A_j - A_i)$ and $C(A_i - A_j)$ exist and A_i and A_j are synonyms imply that the following possibilities. (i) Some potential concepts in $C(A_j - A_i)$ and $C(A_i - A_j)$ are unimportant and should appear in the stoplist; (ii) certain concepts in $C(A_j - A_i)$ and $C(A_i)$ or in $C(A_i - A_j)$ and $C(A_j)$ are synonyms or have is-a relationships between them; (iii) or a combination of (i) and (ii). These situations will be identified and presented to the users for verification. The outcome is that additional knowledge will be stored in the KB.

If the determined relationship between the two names A_i and A_j is verified by the user to be incorrect, then it is possible that one or more keywords that exist in both descriptions of the two names A_i and A_j have different meanings in the context of other words. As an example, the keyword "number" in the description "Tape slot number" should be associated with the concept "IDENTIFICATION", while the same keyword in the description "Number of files on tape" should be associated with the concept "QUANTITY". A simple differentiation of the two meanings of the word is that the former name is probably a key or a part of a key in a database table while the latter is unlikely to be a key. This can be formalized into a rule and used to map a keyword into the correct concept.

4 Conclusion

We have presented a method of mapping names to sets of concepts automatically by their descriptions. By comparing the common parts of two names, the relationship between them, if exist, can be determined. The recognized relationship is presented to the user for verification. The user's feedback is used to add new knowledge to the knowledge base. This permits relationships among other names to be determined more precisely. This method will hopefully greatly reduce the work of the DBA who does the integration of different databases.

Our method is being applied to two NASA databases. More than 300 names appear in these databases.

References

[BrHu91] Bright, M.W. and Hurson, A.R., "Linguistic support for semantic identification and interpretation in multidatabases", *Proc. of the First Int'l Workshop on Interoperability in Multidatabase Systems*, Kyoto, Japan, April 1991, pp. 306-313.

[Demi89] Demichiel, L., "Resolving database incompatibility: an approach to perform relational operations over mismatched domains", *IEEE Trans. on Knowledge and Data Eng.*, Vol. 1, No. 4, Dec., 1989, pp. 485-493.

[Lars89] Larson, J., Navathe, S., and Elmasri, R., "A theory of attribute equivalence in databases with application to schema integration", *IEEE Trans. on Software Eng.*, Vol. 15, No. 4, April, 1989, pp. 449-463.

[Salt89] Salton, G., "Automatic text processing", Addison-Wesley Pub., 1989.

[Shet89] Sheth, A. and Gala, S., "Attribute relationships: an impediment in automating schema integration", NSF Heterogeneous Workshop, Ed. by Yu, C. and Scheuermann, P., Evanston, Illinois, Dec. 11-13, 1989.

[Kim89] Kim, W., "Research directions in heterogeneous databases", *Office Knowledge Engineering*, August, 1989, pp. 3-8.

[SiMa89] Siegal, M. and Madnick, S., "Maintaining valid scheme integration in evolving heterogeneous database systems", *Office Knowledge Engineering*, August, 1989, pp. 9-16.

[YSDK91] Yu, C., Sun, W., Dao, S., and Keirsey, D., "Determining relationships among attributes for interoperability of multidatabase systems", *Proc. of the First Int'l Workshop on Interoperability in Multidatabase Systems*, Kyoto, Japan, April 1991, pp. 251-257.