35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Mapping Query Terms to Data and Schema Using Content Based Similarity Search in Clinical Information Systems

Leila Safari, Jon D. Patrick

Health Language Laboratories, School of Information Technology
The University of Sydney, NSW, Australia
lsaf7301@uni.sydney.edu.au, jonpat@it.usyd.edu.au

***Abstract*—This paper reports on the issues in mapping the terms of a query to the field names of the schema of an Entity Relationship (ER) model or to the data part of the Entity Attribute Value (EAV) model using similarity based Top-K algorithm in clinical information system together with an extension of EAV mapping for medication names. In addition, the details of the mapping algorithm and the required pre-processing including NLP (Natural Language Processing) tasks to prepare resources for mapping are explained. The experimental results on an example clinical information system demonstrate more than 84 per cent of accuracy in mapping. The results will be integrated into our proposed Clinical Data Analytics Language (CliniDAL) to automate mapping process in CliniDAL.**

I. Introduction

Extracting knowledge from data is essential in clinical research, decision making and hypothesis testing. Our ultimate objective is to introduce a special purpose query language for Clinical Data Analytics (CliniDAL), in which a user can express and can compute answers to any question that is answerable from a Clinical Information System (CIS). The restricted natural language form of a user query matches with one of the question and answer categories in CliniDAL which include point-of-care retrieval queries, descriptive statistics, statistical hypothesis testing, scientific studies requiring expression of complex hypotheses, and semantic concept recognition and retrieval.

One issue in designing CliniDAL as a generic language is to be able to install it on any CIS and do analytics on the extracted information from that CIS. However CISs can be designed around a variety of data models dominantly from three kinds of designs, namely: (1) Entity-Relationship (ER) model which is usually implemented as ordinary relational database or star schema; (2) Entity Attribute Value (EAV) model which is mostly implemented as binary relationship in ordinary relational databases and (3) Document (Form) design model which is implemented as XML forms. In addition, much of the useful clinical information is buried in the free text fields of those data models. Hence, these data models constitute the application context in which CliniDAL has to operate if it is to extract the requisite information.

In addition, as textual information in the clinical domain is highly noisy so to facilitate the information extraction we have integrated two different methods in CliniDAL. Firstly, it is necessary to use NLP (Natural Language Processing) techniques to do clinical concept matching. Secondly, similar to keyword based searching in the World Wide Web we want to apply a search on the free text fields of the structural data models. The challenge here is how to match the query expressions at the user level of CliniDAL with the underlying data model terminology. Such functionality in EAV and ER data models is our focus of interest in this paper which will be extended to the XML data model and integrated into CliniDAL.

II. Background

Keyword based search has attracted a number of research projects with the aim of integrating improvements in the information retrieval (IR) area into relational data bases (RDB). The key reason was to bring flexible means for users to query information in RDBs without any need to know either underlying schema or SQL like convenient keyword searching on the Web. The main principles and required data structures of a variety of keyword based search approaches have been studied in a survey [1] which are classified as schema-based and graph-based approaches. Khine et al. [2] have added the class of tuple-unit based methods to these two main classes.

In DBXplorer [3], given a set of keywords it firstly finds all tables which contain at least one of the keywords in their data part. Then a set of sub-graphs are created using a schema graph in which each node of a graph is a relation and each edge is a foreign-key relationship, hence the sub-graphs represent a joining of relations. As a result, all rows from one table or combination of tables which contain all of the query keywords are revealed. They focused on matching keywords across an entire attribute's value set and tried to extend their work to support token or substring matches. BANKS[4] and DISCOVER [5] are two other systems which shared a similar approach.

In [6] a general architecture and a prototype system called EKSO is proposed which supports keyword based search by crawling the content of the connected DB in advance and indexing texts in the virtual documents which are generated from text objects. The virtual documents are generated from interconnected tuples based on joining relational tables from the connected RDB. Their work was similar to Verity [7] in terms of indexing all data in an

offline paradigm, but they prevent transferring data outside the database. In [8] an effective search of text information in relational databases was proposed by introducing a novel ranking strategy and using the Top-K algorithm. The notion of a virtual document in their work is similar to that in [6] but is different from that as it is query specific and dynamic.

In most of the above works, the aim was to find tuples or combinations of tuples which contain all query keywords. However, in our work the key problem is to match query terms to one or a set of attribute ids in an EAV model. Once the appropriate attribute ids are identified they have to be matched to the data fields of the tables in the schema which stores the related data values to the extracted attribute ids. In addition, matching of the query terms to the data fields of the CIS with an ER model is addressed as well. Moreover, all of the above works were conducted in the general domain while to apply similar approaches on data to the clinical domain we need to use clinical ontologies like SNOMED CT (Systematized Nomenclature Of Medicine Clinical Terms), to unify diverse and idiosyncratic query terms created by the users. The results of the current work lead to automating the mapping process for CliniDAL[9].

III. METHODS

A. *Data collection*

The data which are used in our mapping algorithm come from the ICIP information system. ICIP is a commercial system developed by Philips Corporation for use in the Intensive Care Unit (ICU) in the Royal Prince Alfred Hospital (RPAH), Sydney, Australia. The data model of ICIP is a hybrid model of EAV and ER. The EAV model contains several dimension and fact tables. The dimension tables contain the descriptions of the attributes and their internal identifiers which are needed to find their actual instances in the fact tables. The fact tables contain the raw patient data charted in the ICIP system such as arterial blood pressure (ABP) values together with other properties as foreign keys to point to the dimension tables. The dimension tables provide details around who, what, where and how of the data. For example, a dimension table exists which contains the patient name, sex and date of birth. On the other hand, an ER model is used to include patient demographics, care providers details and hospital locations information. So, to extract information about medical attributes we first need to find their internal ids and then the actual data fields in the EAV implementation while to extract information on patient demographics we get them directly from matched fields in the ER implementation. Finally, the foreign key relationships among dimension and fact tables have been used to connect the extracted medical attributes to different dimensions such as patient demographic and care providers.

The important information which can be found in some of the ICIP dimension tables are SNOMED CT concept labels and codes which are not reliable to be used directly in the mapping algorithm as they are not available for all concepts in ICIP. In addition, some concepts are not correctly matched with SNOMED CT due to user faults, and finally, concepts in ICIP matched at various levels of the SNOMED CT concept hierarchy indicating unequal granularity of the ontological levels and therefore irregular use of generalization. For instance among 544 variations which are defined in ICIP for "Paracetamol", 500 ids mapped to "Administration of medication (procedure)" class of SNOMED CT hierarchy, 25 ids mapped to "Non-formulary pharmaceuticals (product)" , 16 ids mapped to "Text value (qualifier value)" and so on. Furthermore, as a generic data analytics tool CliniDAL aims to work with CISs which may not have SNOMED CT nor any other medical ontologies. Fig. 1 illustrates the main steps for mapping process in CliniDAL which the details will be explained in next sections.

B. *Ranked retrieval models*

Matching query terms to free text with dictionaries is not easy since they can appear in the text in different forms to a standard format as found in a dictionary or gazetteer [10]. Similarity methods or ranked retrieval models can be useful to identify the best matches regardless of differences in word order, length or word morphologies in dictionaries with respect to a query. To rank documents, IR systems assign a score for each document as an estimation of the document relevance to a given query. So, the accuracy of results is directly related to the ranking algorithm. A widely used model to compute such scores is the vector space model [11]. Variations of the Top-K algorithm are used in ranked retrieval to return the top k best matches where k is the number of desired matched results. The Top-K algorithm of our work has been used in [10]. The algorithm looks for k records of a dictionary that have the largest similarity matches with a query string x, provided that each similarity is more than a threshold [10].

In this work we have applied a ranked retrieval model on the ICIP CIS with a relational database and EAV model. However, in a real implementation the ICIP system is a hybrid system of EAV and ER models. The main reasons which motivate using ranked retrieval as part of CliniDAL are: firstly, we need to make CliniDAL a generic analytic tool to be able to install on any CIS to extract and analyze information. To provide a general solution we cannot use the schema elements such as the table and field names directly. Rather we need to access that information through the metadata content management structures of the RDMS as that will provide a general solution to search across a schema for terms matching the query expressions. Secondly, most of the information in the clinical domain is noisy, so to increase the reliability of the extracted information it is better to extract several ranked results instead of one unique answer. Thirdly, in the clinical domain the data is very dynamic. Based on our experiments with ICIP, clinical staffs create several similar concepts. For research purposes it is important to extract all of the variations of one concept.

C. *Pre-processing*

The Top-K algorithm is used to find a unique ICIP id (intervention-id) for each query term. User validation is necessary to get one or more ids as a final match since there may be more than one match for a query term in the CIS.

We can categorize query terms based on CliniDAL's grammar which supports patient demographics, medical, hospital location, care provider and temporal constraints.

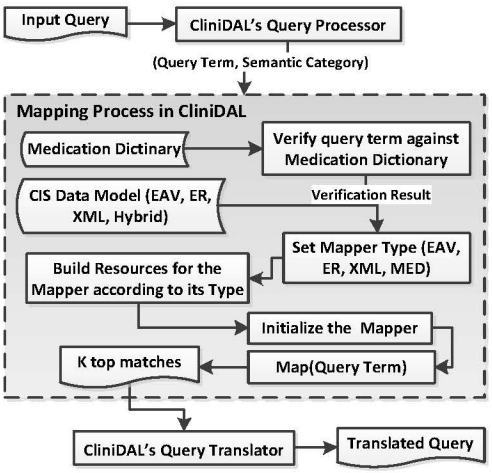

Figure 1: Mapping in CliniDAL

The most important query terms are terms related to the medical category which should be mapped to more than 200,000 concept ids from 13 dimension tables by similarity search. Using the metadata of the CIS we merged all text fields of these dimension tables to create a collection as a basic resource for the Top-K algorithm. Processing the data in these tables revealed that the unique id of each query term for the medical category of CliniDAL constraints is an intervention-id which is stored in the D_Intervention table. So, instead of using all 13 dimension tables with more than 200,000 records, we used the D_Intervention table with more than 58000 records. Finally, to increase the efficiency and accuracy of mapping we filtered out about 8000 ids which are defined but not used in the system. To accomplish this work we made a dictionary of ids with their frequency of usage.

D. *Building Resources*

Before using the Top-K algorithm to map CliniDAL's query terms to ICIP ids, we need to produce the required resources. Two dictionaries were made, an Inverted Index and Terms Collection. Terms Collection was created as a table of about 58000 records of merged text fields from the ICIP D_Intervention table of which the key of the dictionary is the combination of an ICIP intervention id with a table name (separated by $) and the value of that key is a list of tokens in each record. The second dictionary, an Inverted Index, is broadly defined in the Top-K algorithm. Given a token (word) as a key, ids of all records (or documents as in IR) which contain that term together with the similarity score (in decreasing order) of that term in the records are available via the dictionary. For example "heart rate" as a query expression has two keys of "heart" and "rate" in this dictionary. The list [("997$d_intervention", 0.22), ("663$d_intervention", 0.17) …] shows the entry for the term "heart" in this dictionary and we have a similar list for the term "rate". The similarity scores of these lists are used in ranking criteria for the query expression "heart rate" to find records which contain both of these tokens. The number of entries in the Inverted Index is the number of unique tokens in Terms Collection. In the current implementation we have around 9,000 keys. One point here is that the Terms Collection is not completely a static dictionary as clinical staff may add new ids to the system. But the frequency of changes is very low so monthly updating of the collection and consequently the related inverted index is necessary.

To support mapping to the ER model, we have made an ER-Collection and Inverted Index as well. Using metadata of the CIS we extracted the names of all fields of the tables. Then created a collection in which each record is a key of "tablename.fieldname" and the list of tokens includes the table name together with field name. The compound field names are broken down to the atomic tokens. For instance the field name "dateOfBirth" was converted to three tokens "date", "of" and "birth" so in a similarity match it can be matched to "birth date" as well. The ER-Inverted Index is created from the ER-Collection in similar way to the EAV model. The number of keys of the inverted index in ER-mapping implementation was 412 which were used to index 596 records in the ER-collection.

E. *Natural Language Pre-processing Tasks*

Tokenization is used to create the list of tokens as the smallest element of the text from each record that is of interest. The simplest method to tokenize a text in NLP is segmentation based on white space. However, in medical text there are several concepts like scores and measurements which are not just simple strings of characters but use punctuation symbols like ',', '.', '/' to group words which need to be separated for processing. We have used a tokenizer both for tokenizing texts of each record extracted from the ICIP tables and for query terms to run the similarity based mapping algorithm. During tokenization we need to standardize tokens to a lemmatized form so we can match morphological variants of query terms in the same way.

In addition, the large presence of abbreviations, acronyms and incomplete words in queries and records necessitates the use of abbreviation and acronym expansion and verification components as pre-processing steps. The more common tokens between query terms and a record from Terms Collections mean greater similarity between query terms and the CIS record and consequently a better match.

F. *Similarity measure*

Many variations of query-document matching scores are used in the literature. We have used the basic form of TF-IDF similarity measure which contains two components Term Frequency (TF) and Inverse Document Frequency (IDF). After applying pre-processing steps and making the Collection and Inverted Index, the mapping algorithm first finds common tokens between the query and each record in the Terms Collection dictionary. Then it extracts the similarity score of each of the common tokens based on the TF-IDF similarity measure which is previously stored in the Inverted Index and computes the sum of the similarity scores

as final score. In addition, we applied a Completeness Factor[8] to the final ranking score which gives priority when a record matches more query terms and improved our mapping results for more than 8 percent. Fig. 2 shows the pseudo code for computing the completeness factor in our mapping algorithm. ***P*** is the tuning parameter which can be set to a value from 1 to ∞ [8]. In our experiments we set ***P*** to 2.

IV. RESULTS

Table 1 shows a list of query terms which have been used to evaluate the Top-K algorithm on mapping query terms to ICIP ids. Among 221 query terms which have been used for validation of our mapping approach, 187 terms mapped correctly to the ICIP data model which is more than 84 percent correct. Although we used k=5 in the Top-K algorithm to return the 5 best matches to the query term, for 183 concepts the correct match occurred in first position or in the first 2 or 3 positions if we have more than one defined id for a concept, and only for 4 concepts including "abdominal pressure", "icu outcome", "intra abdomen pressure" and "ph" the correct match occurred in 5th, 2nd, 4th and 2nd position respectively. Among the 34 query terms which were not mapped correctly, 12 did not exist in the collection and the remaining 22 terms (including "pain", "bleeding", "drug", "dressing", etc.) were too general to find an exact match. So, we need to attach more query keywords to them to make them more specific to be correctly matched. Table 2 summarizes the results of mapping in ICIP system.

A. *Creating a template for Medications*

Using several variations of similarity function and applying frequency of used ids did not help in correctly matching medications as several ids are defined for some medications. For instance we found 544 variations for Paracetamol and 112 variations for Atorvastatin in ICIP system. So, to get the correct match in mapping medication names to underlying data of a CIS, in addition to their name we need to specify more features like dose, route and frequency to specify all the different values for each feature to make a query term more specific. Consequently, a template has been created for specifying medications including features like Drug Name, Dose, Route (oral, intravenous, etc.) and Frequency (hourly, BD, etc.).

Completeness Factor: Requires query, document, p
1- Compute max of TF(w,document) and max of IDF(w) for each w in query and set tf_max and idf_max
2- Compute normalized term frequency(ntf) for each w in query using: ntf= (TF(w,document)/tf_max) * (IDF(w)/idf_max)
3- compute sum = $\sum_w (1-ntf)^p$
4- return 1- (sum/len(query)) $^{1/p}$

Figure2. Algorithm for Completeness Factor

TABLE 1. A LIST OF QUERY TERMS MAPPED TO ICIP IDS

ALP	IBW	Abdominal pressure	MV Spontaneous
ALT	LCW	Attending Physician	Nasogastric Aspirate
BGL	PEEP	Bilirubin Total	Respiratory Arrest
BSA	PLT	Body height	Urine Sodium Random

TABLE 2. MAPPING RESULTS

	# Matched	# Not Matched	Total	Per cent
EAV	165	32	197	83.8
ER	22	2	24	91.7
Total	187	34	221	84.6

However, in data analytics the name of the medication is of most interest for a query from a CIS. For instance, to study the effect of administration of Paracetamol in reducing temperature in patients, we prefer to use "Paracetamol" as a query term rather than specifying any route or frequency for it. So, we have created a conversion of the EAV mapping model which aims to extract all medications with a specified name in the query term rather than top k similar ones. Hence, there is no need to compute a similarity measure for medication items. As a result a simplified form of the Inverted Index of the EAV mapping model is created in which the similarity score is removed. So, we can extract all ids for query term by using it as the key of the Inverted Index effectively. Then the extracted ids have to be validated by the user and one or a set of these ids are chosen to answer the query. Finally, to accomplish the medication mapping we have created a dictionary of terminologies used for medication names based on the SNOMED CT ontology. Finding a match for the query term in this dictionary (Fig. 1) means that mapping have to be done by Medication mapping model rather than EAV mapping.

REFERENCES

[1] J. X. Yu, L. Qin, & L. Chang, "Keyword search in relational databases: A survey". *IEEE Data Eng. Bull.* vol. 33, pp. 67-78, 2010.

[2] P. T. T. Khine, H. P. P. Win, & K. N. N. Tun, "Indexing Relational Databases for Efficient Keyword Search". vol. 2, 2011.

[3] S. Agrawal, S. Chaudhuri, & G. Das, "DBXplorer: A system for keyword-based search over relational databases", in *Proc. the 18th International Conference on Data Engineering* IEEE, 2002, pp. 5-16.

[4] B. Aditya, G. Bhalotia, S. Chakrabarti, A. Hulgeri, C. Nakhe, P. Parag, & S. Sudarshan, "Banks: Browsing and keyword searching in relational databases", in *Proc. the 28th international conference on Very Large Data Bases*, VLDB Endowment, 2002, pp. 1083-1086.

[5] V. Hristidis, & Y. Papakonstantinou, "Discover: Keyword search in relational databases", in *Proc. the 28th international conference on Very Large Data Bases*, VLDB Endowment, 2002, pp. 670-681.

[6] Q. Su, & J. Widom, "Indexing relational database content offline for efficient keyword-based search", in *Proc. Database Engineering and Application Symposium, 2005. IDEAS 2005. 9th International*, IEEE, 2005, pp. 297-306.

[7] P. Raghavan, "Structured and unstructured search in enterprises". *IEEE Data Engineering Bulletin*. vol. 24, pp. 15-18, 2001.

[8] Y. Luo, X. Lin, W. Wang, & X. Zhou, "Spark: top-k keyword query in relational databases", in *Proc. International Conference on Management of Data: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007, pp. 115-126.

[9] J. D. Patrick, L. Safari, & Y. Cheng, "Knowledge Discovery and Knowledge Reuse in Clinical Information Systems", in *Proc. The 10th IASTED International Conference on Biomedical Engineering (BioMed 2013)*, Innsbruck, Austria, 2013,

[10] S. M. Sabbagh-Jafari, "Text Mining in Clinical Notes", Phd Thesis. School of Information Technologies Sydney: University of Sydney. 2012.

[11] F. Liu, C. Yu, W. Meng, & A. Chowdhury, "Effective keyword search in relational databases", in *Proc. Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, ACM, 2006, pp. 563-574.