On the Semantic Mapping of Schema-agnostic Queries: A Preliminary Study

André Freitas[1], João C. Pereira da Silva[2], Edward Curry[1]

[1]Insight Centre for Data Analytics, National University of Ireland, Galway
[2]Computer Science Department, Federal University of Rio de Janeiro

Abstract. The growing size, heterogeneity and complexity of databases demand the creation of strategies to facilitate users and systems to consume data. Ideally, query mechanisms should be *schema-agnostic* or *vocabulary-independent*, i.e. they should be able to match user queries in their own vocabulary and syntax to the data, abstracting data consumers from the representation of the data. Despite being a central requirement across natural language interfaces and entity search engines, there is a lack on the conceptual analysis of schema-agnosticism and on the associated semantic differences between queries and databases. This work aims at providing an initial conceptualization for schema-agnostic queries aiming at providing a fine-grained classification which can support the scoping, evaluation and development of semantic matching approaches for schema-agnostic queries.

Keywords: Schema-agnostic Queries, Semantic Matching, Natural Language Interfaces, Databases

1 Introduction

The growing data availability on Big Data environments demands the creation of strategies to facilitate the interaction between data consumers and databases. As the number of available data sources grows and schemas increase in size and complexity, the manual effort of building structured queries such as SPARQL and SQL becomes prohibitive. Ideally data consumers, being them humans or intelligent agents, should be able to be abstracted from the representation of the data by using a *schema-agnostic query mechanism*.

Schema-agnostic or *vocabulary-independent* queries can be defined as query approaches over structured databases which allow users satisfying complex information needs without a prior understanding of the representation (schema) of a structured database. Similarly, Tran et al. [3] defines it as '*search approaches, which do not require users to know the schema underlying the data*'. A mechanism which supports *schema-agnostic queries* is dependent on the support of a *semantic model* and of a *semantic mapping procedure*. The semantic differences between the query elements T and the database elements E (instances, attributes, etc), define the phenomenon of *query-database semantic heterogeneity*.

Despite being a central requirement across natural language interfaces, entity search engines and databases in general, there is a gap on the conceptualization of *schema-agnosticism* and of a more structured analysis of the semantic gap between queries and databases. This work aims at providing an initial conceptualization for schema-agnostic queries, extending the *semantic tractability* model, introduced by Popescu et al. [2] in the context of natural language interfaces over databases. The proposed model aims to provide a deeper and more fine-grained understanding of the semantic challenges involved in mapping schema-agnostic queries to databases, supporting a better scoping of new contributions and evaluation campaigns for schema-agnostic query mechanisms.

2 Dimensions of Query-Database Semantic Heterogeneity

Most of the analysis on semantic heterogeneity have been done in the context of data/schema integration, providing a comprehensive analysis of the dimensions involved in the semantic heterogeneity between two datasets. The problem of semantically matching a schema-agnostic query and dataset elements has commonalities to the problem of aligning elements between two datasets. The specificity of query-database alignments, however, lies on the asymmetry between the level of available contextual information and on the lack of a structured context from the query side. This section discusses and classifies the dimensions of semantic heterogeneity in the context of the gap between query and database, organizing them into a taxonomy of query-database *semantic differences*. The construction of the taxonomy of query-database differences was guided by the works of George[5] and Sheth & Kashyap[4]. The categories for the taxonomy of query-database lexico-semantic differences are described below.

1. *Synonym:* Different lexical expressions mapping to the same concept (e.g. customer vs. client).
2. *Lexical Differences:* Lexical expressions with the same morphological roots mapping to strongly related concepts.
3. *Conceptual Differences:* Distinct but related concepts under different lexical expressions in which the alignment satisfies the query information need.
   (a) *Taxonomical Differences:* Abstraction-level differences between the query and the database elements. *'PresidentsOfTheUnitedStates'* and *'AmericanPoliticians'* express two different sets where the former set is contained in the latter. In some cases the abstraction level expressed in the query may be different from the dataset and only a semantically approximate result can be returned. Two entities are *semantically similar* if they are under the same taxonomical structure.
   (b) *Non-taxonomical Differences:* A concept in the query and a concept in the database can represent distinct but strongly related concepts in the context of the query. For example the correspondence between *'married'* and *'spouse'*. Two entities are *semantically related* if they have a non-taxonomical and non-synonymic semantic relationship.

4. *Compositional/Predication Differences:* Information may be expressed as different compositions of different database elements or predicate structures. *'PresidentsOfTheUnitedStates'* can be expressed as a single predicate or as a composition of the binary predicate *'president'* and the instance *'UnitedStates'*.
5. *Functional Differences:* Aggregated information may be already conceptualized in the database or may need to be computed based on existing data. For the example query in Figure 1(1), the predicate *'numberOfKids'* could be expressed directly on the database or may need to be computed as an aggregation function over statements containing the predicate *'child'*. Superlatives are also examples of concepts which can be expressed either as predicates or through functions (e.g. *'highest'* mapping to *'elevation'*) in Figure 1(4).
6. *Convention Differences:* Consists of differences in the representation of the values and units used (RGB vs. HSV color scheme), dates (dd.mm.yy vs. mm.yyyy), numbers (first vs. 1st), dimension, units of measure and scale differences (units of measure, volume, weight, size, currency), unique identifiers (employer ID vs. employer SSN).
7. *Null Mappings:* Consists of a null mapping from a query term to a database element or vice-versa.
8. *Intensional Differences:* Consists of different intensional definitions expressed by the same term. The definitions for *'taxable revenue'*, *'age of majority'* and *'economically active population'* are concepts which are likely to vary between different regions, groups, etc. Although representing similar concepts they might be defined under different criteria.
9. *Contextual Differences:* Consists of scoping differences (e.g. temporal, spatial) in the context in which an alignment holds. The predicate *'most awarded actor'* can vary for different time spans and countries.

The classification above focuses on a mono-lingual and single data model query scenario. Schema-agnostic queries might include cross-language and cross-data models queries.

In order to address the vocabulary problem, schema-agnostic query approaches depend on the ability to match queries to database elements. The next sections formalize the problem of semantic matching using as a basis the concept of *semantic tractability* developed by Popescu et al. [2].

3 Semantic Tractability

3.1 Basic Concepts

Definition 1 (Data Model, Dataset, Dataset Lexicon). *A* ***data model*** $\mathcal{DM}$ *is a set* $\mathcal{T}_{\mathcal{DM}}$ *of data model types and relations* $\mathcal{R}_{\mathcal{DM}}$ *between these types. A* ***dataset*** DS *is a data collection which is represented under a data model* $\mathcal{DM}$*. The* ***dataset lexicon*** Lex_{DS} *of* DS *is a tuple of* $(t_0, \cdots, t_n)$ *where* $t_i \in \mathcal{T}_{\mathcal{DM}}$*.*

Definition 2 (Query). *A natural language question* q *can be represented by a* ***query*** Q *that is a tuple* $(Token_q, Att_q)$ *where* $Token_q$ *is the ordered set of tokens that form the question* q *and* $Att_q : Token_q \rightarrow Token_q$ *is the attachment function (syntactic relationship) between elements in* $Token_q$*.*

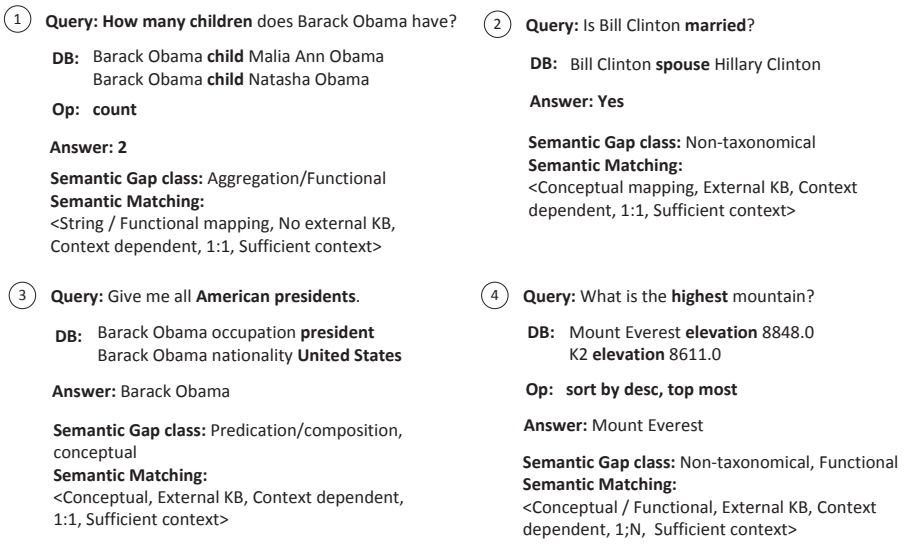

Fig. 1: Classification of existing queries according to the *lexico-semantic differences* and *semantic mappings*.

Definition 3 (Interpretation of a Query). *An interpretation of a query Q is a tuple $Q^{struct} = (E, R, L, Op, V)$, where E are a set of database elements mapped to the query, R is an ordered set of syntactic n-ary associations between elements in E, L is a set of logical operators, Op is a set of functional operators and V is a set of binding variables.*

Definition 4 (Syntactic Mapping). *Given a data model $\mathcal{DM}$ and a query Q with interpretation Q^{struct}, we can define a mapping function $m(Q, \mathcal{DM}) : Token_q \rightarrow E$ which defines the possible syntactic realizations of Q under $\mathcal{DM}$.*

The syntactic interpretation of a query Q, denoted by $I(Q, \mathcal{DM})$ are the possible realizations of Q under the data model $\mathcal{DM}$, such that $I(Q, \mathcal{DM})$ is semantically equivalent to Q.

3.2 Semantic Tractability

Popescu et al. [2] defines a framework to evaluate the reliability of a NLI, defining formally the properties of soundness and completeness and identifying a class of semantic tractable natural language queries. *Semantic tractability* essentially expresses that there should be a syntactic correspondence between the syntactic structure of the query and the syntactic structure of the database and a synonymic correspondence.

Definition 5 (Semantic tractability). *Given a query Q and a dataset DS with lexicon Lex_{DS} and data model $\mathcal{DM}$. If the query and dataset satisfies $m(Q, \mathcal{DM})$*

and there is a mapping $map_{(Q,DS)} : Token_q \rightarrow Lex_{DS}$ *such that* $Token_q$, Lex_{DS} *are considered synonyms whenever there is a mapping between* $Token_q$ *and* Lex_{DS}, *then the associated question* q *is considered semantically tractable.*

The concept of semantic tractability assumes that there is a *one-to-one synonym mapping* between the query and the database lexica which preserves the *dataset predicate-argument structure induced by the lexical categories of the query*, leaving the problem of *conceptual matching* and *more complex syntactic matching* out of the definition. This *unambiguous synonymic correspondence* which is the condition for semantic tractability cannot be guaranteed in a large schema/schema-less database query scenario, where the database lexicon is potentially very large, and the same terms can be used in different contexts with different meanings.

Additionally, with a large vocabulary variation, it is also not possible to guarantee a *syntactic correspondence between query and database*, rendering a significant part of the queries to the status of being not semantically tractable. Different conceptualizations induce structural differences in the dataset which correspond to different syntactic structures in the query.

In order to extend this classification, the concept of *semantic resolvability* is defined to cope with other category of semantic mappings.

4 Mapping Schema-Agnostic Query

4.1 Semantic Resolvability

In order to define a broader class of query-dataset mappings, the concept of a *semantic Knowledge Base (KB)* is introduced which, supports the $Token_q \rightarrow Lex_{DS}$ mapping.

Definition 6 (Semantic Knowledge Base (KB)). *A semantic knowledge base* $\mathcal{M}_\Sigma$ *with signature* $\Sigma = (\mathcal{R}, \mathcal{E})$ *is a collection of concepts constructed using two finite sets of symbols representing relations (and properties)* $r \in \mathcal{R}$ *and entities* $e \in \mathcal{E}$.

Definition 7 (Associated Semantic KB). *Given a semantic KB* $\mathcal{M}_\Sigma$ *with signature* $\Sigma = (\mathcal{R}, \mathcal{E})$ *and a lexicon Lex, we say that* $\mathcal{M}_{\Sigma,Lex} = (\mathcal{M}_\Sigma, f)$ *is the associated semantic KB wrt Lex whenever* f *is a mapping defined by*

$$f : Lex \rightarrow (\mathcal{R} \cup \mathcal{E})$$

We can define a mapping f_{cpt} from concepts in $\mathcal{M}_{\Sigma,Lex}$ to concepts in $\mathcal{M}_\Sigma$ using f as follows: $f_{cpt}(c(e_0, \cdots, e_n)) = f(c)(f(e_0), \cdots, f(e_n))$, where $f(c) \in \mathcal{R}$ and $f(e_0), \cdots, f(e_n) \in \mathcal{E}$.

Definition 8 (Semantic Reachability). *A concept* $r_n \in \mathcal{M}_\Sigma$ *is reachable from a concept* $r_0 \in \mathcal{M}_\Sigma$ *if there is an ordered sequence* $\langle r_0, r_1, \cdots, r_n \rangle$ *where for all* $i \in [0, n-1]$, *exist* $u \in [1, arity(r_i)]$ *and* $v \in [1, arity(r_{i+1})]$ *such that* $proj(r_i, u) = proj(r_{i+1}, v)$ *where* $arity(r)$ *means the arity of relation* r *and* $proj(x, y)$ *represents the* y*-ary argument of a relation* x.

A concept $c_n \in \mathcal{M}_{\Sigma,Lex}$ *is reachable from a concept* $c_0 \in \mathcal{M}_{\Sigma,Lex}$ *whenever* $f_{cpt}(c_n)$ *is reachable from* $f_{cpt}(c_0)$.

Definition 9 (Query-Dataset Semantic Mapping). *Given a query Q and a dataset DS with lexicon Lex_{DS}, a query-dataset semantic mapping wrt an associated semantic KB $\mathcal{M}_{\Sigma,Token_q}$ is a mapping*

$$map_{(Q,DS,\mathcal{M}_{\Sigma,Token_q})} : Token_q \rightarrow Lex_{DS}$$

such that $\forall c \in Token_q$, if $Dep_q(c) = d$ then $f_{cpt}(d)$ is reachable from $f_{cpt}(c)$.

Definition 10 (Semantic Resolvability). *A query Q is semantically resolvable to a dataset DS when $\forall t_i \in Token_q$ exists a semantic mapping $map_{(Q,DS,\mathcal{M}_{\Sigma,Token_q})}$ under a semantic KB $\mathcal{M}_\Sigma$ which satisfies the syntactic constraints in Dep_q and DS.*

Definition 11 (Resolved Schema-Agnostic Query). *A query Q over a dataset DS is a resolved schema-agnostic query if there is a semantic KB $\mathcal{M}_\Sigma$ in which Q is semantically resolvable to DS.*

4.2 Semantic Mapping Types

In the previous section the concept of *semantic mapping* was introduced without the analysis of the types and conditions involved in the semantic mappings supported by the semantic KB. However, under realistic scenarios, semantic mapping approaches need to cope with *inconsistent*, *incomplete* semantic KBs and *ambiguous*, *vague* queries and databases. This work builds upon the basis developed in the context of schema matching (in particular adapting the work of Kashyap & Sheth [4]) to provide a classification for types of *query-dataset mappings*.

Definition 12 (Semantic Mapping Type). *Given a query Q, a dataset DS with lexicon Lex_{DS} and a query-dataset semantic mapping $map_{(Q,DS,\mathcal{M}_{\Sigma,Token_q})}$, for all $t_i \in Token_q$, the semantic mapping type of (t_i, e_i), where $e_i = map_{(Q,DS,\mathcal{M}_{\Sigma,Token_q})}(t_i)$, is defined by the tuple $(\mathcal{AP}, \mathcal{PS}, \mathcal{M}, \mathcal{SE}, \mathcal{CT}, \mathcal{MC})$, where:*

Abstraction Process $\mathcal{AP}$: is defined as a mechanism used to map the concept associated with t_i to the concepts associated with the database elements e_i.

1. *Trivial:* A semantic mapping is *trivial* if the lexical expression of t_i is identical to the lexical expression of e_i and both t_i and e_i have a single word sense.
2. *Lexical:* A semantic mapping is *lexical* if t_i and e_i have a *common morphological root* r.
3. *Synonymic:* A semantic mapping is *synonymic* if t_i and e_i are synonyms and have the same lexical category.
4. *Generalization/Specialization:*
   (a) *Generalization:* A semantic map is a *generalization* if e_i is a superclass of t_i.
   (b) *Specialization:* A semantic map is a *specialization* if e_i is a subclass of t_i.
5. *Conceptual:* A semantic map is a *conceptual mapping* if t_i and e_i are non-taxonomically related and if there is a non-taxonomical inference process supporting the alignment between t_i and e_i.
6. *Functional/Aggregation:* A semantic mapping is *functional* if there is a functional operator op_j which maps dataset tuples to t_i.

Predicate Structure $\mathcal{PS}$: Maps to differences in the associated predicate structure from the projection of t_i into the data model $\mathcal{DM}$ and the predicate structure of e_i.

1. *Predication preserving:* If the predicate structure between t_i and e_i is preserved.
2. *Predication difference:* If the predicate structure between t_i and e_i is not preserved.

Semantic Knowledge Base $\mathcal{M}$: Consists of the type of a semantic knowledge base supporting the semantic mapping.

1. *Self-sufficient:* The semantic mapping does not depend on a knowledge base external to the dataset.
2. *Dependent on External Knowledge Base:* The semantic mapping depends on a knowledge base external to the dataset.

Semantic Evidence & Uncertainty $\mathcal{SE}$: Consists of the categorization of the mapping according to the supporting *semantic evidence* and *uncertainty* in the query, dataset, and in the semantic KB .

1. *Absolute:* A semantic mapping is *absolute* if for every possible context, t_i maps to e_i. An absolute mapping is independent of the context provided by the query and by the dataset.
2. *Context resolvable:* A semantic mapping is *context resolvable* if there is a mapping between t_i and e_i which is uniquely determined by a proper query and dataset contexts.

Context $\mathcal{CT}$: Consists of the query context $Q^{context} = \{t_i \mid t_i \in Token_q\}$ and the dataset context $DS^{Context} = \{e_i \mid e_i = map_{(Q,DS,\mathcal{M}_{\Sigma,Token_q})}(t_i)\}$

1. *Sufficient:* The context is *sufficient* to determine the query-dataset mapping given a context-resolvable semantic evidence scenario.
2. *Insufficient:* The context is *insufficient* to determine the query-dataset mapping given a context-resolvable semantic evidence scenario, leading to ambiguity or vagueness in the query-dataset semantic mapping.

Mapping cardinality $\mathcal{MC}$:

1. *Single mapping ($1:1$):* A semantic mapping is a *single mapping* if $map_{(Q,DS,\mathcal{M}_{\Sigma,Token_q})}$ is a one-to-one map.
2. *Data redundant ($1:N$):* A semantic mapping is *data redundant* if $map_{(Q,DS,\mathcal{M}_{\Sigma,Token_q})}$ is a multi-valued map.
3. *Query redundant ($N:1$):* A semantic mapping is *query redundant* if $map_{(Q,DS,\mathcal{M}_{\Sigma,Token_q})}$ is a many-to-one map between $Token_q$ and DS.
4. *Query-data redundant ($M:N$):* A semantic mapping is *query-data redundant* if $map_{(Q,DS,\mathcal{M}_{\Sigma,Token_q})}$ is a many-to-many relationship between $Token_q$ and DS.

The concept of *semantic tractability* corresponds to the tuple $(\mathcal{AP}, \mathcal{PS}, \mathcal{M}, \mathcal{SE}, \mathcal{CT}, \mathcal{MC}) = (\{*\}, \{\text{Predication Preserving}\}, \mathcal{M}, \{\text{Absolute, Context Resolvable}\}, \{\text{Sufficient}\}, *)$, which corresponds to a small subset of the possible mapping types.

The process of assigning a database associated interpretation $I_{DS}(Q)$ to a schema-agnostic query Q depends on coping with the semantic phenomena of term ambiguity, syntactic/structural ambiguity, vagueness and synonymy, given

the query Q, the dataset DS and the semantic KB $\mathcal{M}_{\Sigma}$. The interpretation is associated with mappings between four sets: (i) a *word set* W, which expresses the set of words used to describe the domain of discourse of the query tokens and the database lexicon, (ii) a *word sense set* WS, which describes the possible senses associated with the words within the semantic KB, (iii) a *composition set* S, to describe the possible (syntactically valid) compositions of words and (iv) a *concept set* C, to describe the set of concepts associated with the possible interpretation for all the compositions. The unambiguous *semantic interpretation* of a query $I(q)$ or database statement $I(s)$ is a concept c_i in the concept set.

5 Discussion

This work provides an initial framework for modeling the semantic differences and the semantic mappings types between schema-agnostic queries and structured databases. The semantic tractability framework proposed by Popescu et al. [2] was generalized in two directions: (i) proposing a model which is data model independent (in contrast with the focus on relational databases present in [2]) and (ii) deriving a more general set of categories for classifying query-database mappings. The concept of semantic tractability maps to a small subset of the possible query-database mapping conditions, leaving most of the queries out of the discussion. This work aims at providing a more comprehensive classification framework. We expect that this categorization can support a better scoping of the contributions of existing research and evaluation campaigns for natural language interfaces, entity search and schema-agnostic queries, providing the vocabulary to categorize existing semantic matching challenges.

Acknowledgment: This publication was supported in part by Science Foundation Ireland (SFI) (Grant Number SFI/12/RC/2289) and by the Irish Research Council.

References

1. Furnas, G. W. and Landauer, T. K. and Gomez, L. M. and Dumais, S. T., The Vocabulary Problem in Human-system Communication, Commun. ACM, 30, 11 pp. 964-971 1987.
2. Popescu, A. and Etzioni, O. and Kautz, H., Towards a Theory of Natural Language Interfaces to Databases, Proceedings of the 8th International Conference on Intelligent User Interfaces, IUI, 2003.
3. Tran, T. and Mathäß, T. and Haase, P., Usability of Keyword-driven Schema-agnostic Search: A Comparative Study of Keyword Search, Faceted Search, Query Completion and Result Completion, Proceedings of the 7th International Conference on The Semantic Web: Research and Applications, 2010.
4. Kashyap, V. and Sheth, A., Semantic and Schematic Similarities Between Database Objects: A Context-based Approach, The VLDB Journal, 1996.
5. George, D., Understanding Structural and Semantic Heterogeneity in the Context of Database Schema Integration, Technical Report, 2005.