SEMANTIC QUERY LANGUAGE FOR TEMPORAL GENEALOGICAL TREES

Innopolis University

Thesis submitted to The Innopolis University in conformity with the requirements for the degree of Bachelor of Science.

presented by

Evgeniy Gryaznov

supervised by

Manuel Mazzara

Date

SEMANTIC QUERY LANGUAGE FOR TEMPORAL GENEALOGICAL TREES

To my parents and close relatives. This work would't be possible without them.

Contents

List of Tables

List of Figures

Abstract

abstract ...

Chapter 1

Introduction

1.1 Spacing & Type

This is a section. This is a citation without brackets **?**. and this is one with brackets [**?**]. These are multiple citations: [**?**, **?**, **?**]. Here's a reference to a subsection: 1.1.1. The body of the text and abstract must be double-spaced except for footnotes or long quotations. Fonts such as Times Roman, Bookman, New Century Schoolbook, Garamond, Palatine, and Courier are acceptable and commonly found on most computers. The same type must be used throughout the body of the text. The font size must be 10 point or larger and footnotes[1] must be two sizes smaller than the text[2] but no smaller than eight points. Chapter, section, or other headings should be of a consistent font and size throughout the ETD, as should labels for illustrations, charts, and figures.

1.1.1 Creating a Subsection

Creating a Subsubsection

This is a heading level below subsubsection And this is a quote:

[1] This is a footnote.
[2] This is another footnote.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is a table:

Table 1.1: This is a caption.

A	B
a1	b1
a2	b2
a3	b3
a4	b4

The package "upgreek" allows us to use non-italicized lower-case greek letters. See for yourself: $\upbeta$, $\boldsymbol{\upbeta}$, β, $\boldsymbol{\beta}$. Next is a numbered equation:

$$\|\boldsymbol{X}\|_{2,1} = \underbrace{\sum_{j=1}^{n} f_j(\boldsymbol{X})}_{\text{convex}} = \sum_{j=1}^{n} \|\boldsymbol{X}_{\cdot,j}\|_2 \tag{1.1}$$

The reference to equation (1.1) is clickable.

1.2 Theorems, Corollaries, Lemmas, Proofs, Remarks, Definitions, and Examples

Theorem 1. *Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.*

Proof. I'm a (very short) proof. □

Lemma 1. *I'm a lemma.*

Corollary 1. *I include a reference to Thm. 1.*

Proposition 1. *I'm a proposition.*

Remark. I'm a remark.

Definition 1. I'm a definition. I'm a definition. I'm a definition. I'm a definition. I'm a definition. I'm a definition. I'm a definition. I'm a definition. I'm a definition. I'm a definition. I'm a definition. I'm a definition.

Example. I'm an example.

1.3 Section with linebreaks in the name

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 2

Literature Review

The purpose of this chapter is to give a brief analysis of literature with respect to applied theory in our thesis.

There are two major areas of Computer Science that are related out research: Knowledge Representation (KR) and Natural Language Processing (NLP). Each of these fields provides a crucial contribution to our topic. We will examine them one by one and highlight some of the most important works.

2.1 Knowledge Representation

The field of Knowledge Representation is concerned with how the knowledge about our physical world can be stored, managed and utilized by machines or computers. KR owes its existence to the more general field of Artificial Intelligence, which prompted the study of encoding information about the physical reality into an intelligent system in such a way that it can be used by that system to solve complex problems.

The main presupposition of the whole field of KR is that, in order for an intelligent agent to resolve a difficult problem, it needs an access to some form of a knowledge specific to a particular domain and that knowledge should be stored inside the agent. This presupposition is now widely known as the *Knowledge*

Representation Hypothesis:

> Any mechanically embodied intelligent process will be comprised of structural ingredients that (a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits (b) independent of such external semantical attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge.

This original formulation of the hypothesis is due to Smith [1].

Brachman and Levesque outlined the hypothesis in their article *Expressiveness and tractability in Knowledge Representation and Reasoning* [2]. The authors argue that the trade-off between *expressiveness* of a knowledge-based system and its *tractability* (i.e. the ability to reason correctly) is intrinsic to every such system, and can only be partially solved. According to them, it's impossible to implement a knowledge-based system that will be both highly expressive and completely tractable.

Moreover, the whole enterprise of encoding knowledge directly into an agent is proven to be useful only in domain-specific applications, such as the famous *block world* [3] domain. A solution with at least partially hard-coded knowledge is tremendously difficult to scale. Since the dawn of Machine Learning, the plausibility of the hypothesis is continuously challenged. Indeed, it is questionable whether we can say about a neural network that it stores knowledge in the weights of its neurons. Thus, today the field of KR doesn't enjoy that much popularity because the main focus of the AI community has shifted to other areas, such as Deep Learning.

The most prominent advances in the KR field were the development of Description (Terminological) and Temporal logics and the formulation of the concept of Ontology. They all are of great importance to our research, so we will survey them one by one.

2.1.1 Ontologies

The word "ontology", derived from the Greek word meaning "the study of being", was unwarrantably borrowed by computer scientists from the namesake branch of philosophy concerned with the nature of reality. Despite being a useful coinage in informatics, not all philosophers are content with such state of affairs [4].

The term "ontology" in Computer Science refers to the mechanism by which reality is compartmentalized into a strictly-defined categories only to be read by machines later. According to Josephson et al. [5] an ontology comprises a body of knowledge about a particular domain of interest. However, we must distinguish between an abstract conceptualization of a particular domain and a concrete instantiation of it. The latter is usually implied when the plural word "ontologies" is used.

The typical ontology consists of:

1. A finite set of *concepts* (a.k.a. nodes, classes). Represents entities of a domain.

2. A finite set of *properties* (a.k.a. attributes, slots, roles). Represents what can be asked of a concept.

3. A finite set of *relationships* between concepts.

4. A finite set of logical *constraints*, which put the boundaries around what can and cannot be stored in an ontology.

At the first sight, the description of an ontology highly resembles that of a database. Indeed, it is true that every database can be seen as a special case of ontology, but not vise-versa. As noted in [2], the power of ontologies lies not in what can be said in it, but exactly *what can be left unsaid*. For example, suppose we want to store the birth date of our grandfather, but all we know about him is that he won a medal fighting in WWII. Then we are forced, using a database, to left the `birth_date` field empty, thus loosing the knowledge of his

heroic deed. But, we can eloquently express this knowledge in some ontological lisp-like language as:

```
(set birth_date (father (father me)) (during WWI))
```

Later we can use this fact to reason about our ancestor more efficiently.

Ontologies find their natural application in the context of our thesis. Since the original formulation of a concept, a lot of software has been developed to manage ontologies, including such systems as Protege, In4j and others. These systems have already been heavily used in the variety of different fields.

For instance, Tan Mee Ting [6] designed and implemented a genealogical ontology using Protege and evaluated its consistency with Pellet, HermiT and FACT++ reasoners. He showed that it is possible to construct a family ontology using *Semantic Web* [7] technologies with full capability of exchanging family history among all interested parties.

An ontology can be used to model any kind of family tree, but the problem arises when a user wants to query his relatives using kinship terms. No standard out-of-the-box ontology query language is able to articulate statements such as in our example above. Although an ontology can be tailored to do so, it is not in any way a trivial matter. Maarten Marx [8] addressed this issue, but in the different area. He designed an extension for XPath, the first order node-selecting language for XML.

Catherine Lai and Steven Bird [9] described the domain of linguistic trees and discussed the expressive requirements for a query language. Then they presented a language that can a wide range of queries over these trees, and showed that the language is first order complete. This language is also an extension of XPath.

Artale et al. [10] did a comprehensive survey of various temporal knowledge representation formalisms. In particular, they analysed ontology and query languages based on the linear temporal logic LTL, the multi-dimensional Halpern-Shoham interval temporal logic, as well as the metric temporal logic MTL. They note that the W3C standard ontology languages, OWL 2 QL and OWL 2 EL, are

designed to represent knowledge over a static domain, and are not well-suited for temporal data.

2.1.2 Temporal and Description Logics

The usage of logic in the field of KR is motivated by its excellence in such areas as mathematics and computer science in general. The early researches in KR saw the unharvested power of logic – especially first-order logic – as a main component in any intelligent system. Subsequent works showed that FOL can provide semantics for specific kind of KR structures: *frames* [11]. Later, Brachman and Levesque proved [2] that we don't need the *whole* FOL for that purpose, but only certain fragments of it. Moreover, different fragments of FOL have different expressive power and tractability. Thus, research in the area of Description Logic began under the label *terminological systems*, only to be later renamed to *Description Logic* when the focus was shifted to the properties of underlying logical systems.

Description Logic finds its application in the context of this thesis as a natural formalism for family trees. However, as expressive as any DL can be, formulating the concept of time requires adding another modal operator. Any logic which handles time is knows as *temporal logic*. Philosophers have tried to put time into a coherent framework since *Aristotle*, in the 20th century mathematicians and computer scientists proposed various formalisms, among which was Allens' *interval algebra* [12] and the temporal logic of Shoham [13]. Thus, what we need in this thesis is an amalgamation of a description and temporal logic.

The first successful attempt at integrating two logics is due to Schmiedel [14]. He combined the DL in the tradition of KL-ONE [15], Shohams' [13] temporal logic and Allens' [12] algebra into one unifying framework. The main features of his formalism are the complete preservation of original DL and the use of lisp-like syntax for expressing roles, concepts and time entires.

The application of temporal logic to graphs, relational databases and ontolo-

gies is also a heavily-invested subject. Barcelo and Lubkin examined [16] several temporal logics over unranked trees and characterize commonly used fragments of first-order (FO) and monadic second-order logic (MSO) for them. They also considered MSO sibling-invariant queries, that can use the sibling ordering but do not depend on the particular one used, and captured them by a variant of the -calculus with modulo quantifiers.

Alexander Tuzhilin and James Clifford defined [17] a temporal algebra that is applicable to any temporal relational data model supporting discrete linear bounded time. This algebra has the five basic relational algebra operators extended to the temporal domain and an operator of linear recursion. They showed that this algebra has the expressive power of a safe temporal calculus based on the predicate temporal logic with the "until" and "since" temporal operators.

Perry in his dissertation [18] highlighted that even in state-of-the-art ontological query languages, such as OWL, expressing the concept of time is an arduous task. He augmented the *Resource Description Framework* with temporal RDF graphs and extended the W3C-recommended SPARQL query language to support these new structures.

An adequate representation of time is the holy grail among the researchers in the field of ontology development. Baratis et al. [19] designed and implemented *TOQL*: a high-level SQL-like language which is capable of expressing temporal queries. They motivate the need for such a language by noting that conveying the concept of time using classical languages, such as OWL, is proven to be difficult, although feasible. They also developed an application that supports translation and execution of TOQL queries on temporal ontologies combined with a reasoning mechanism based on event calculus.

2.2 Natural Language Processing

The main goal of Natural Language Processing (NLP) field is to invent, study and implement algorithms and techniques that help a computer understand an ordinary language, such as English, Russian, French or Swahili.

A lot of research in NLP is dedicated to the problem of querying a relational database in some natural language. Since the early developments, a substantial progress has been achieved. For instance, Jeremy Ferrero et al. proposed and implemented [20] a solution to query any database, irrespective of its' schema, in virtually any natural language. They showed that it supports more operations that most of the other translators. They tested their program on English and French languages.

Another similar attempt was made [21] by Norouzifard et al. They implemented an expert system using Prolog to transform a sentence in a natural language to SQL. Chaudhari [22] presented a light weight technique of converting a natural language statement into equivalent SQL statement.

Nelken et al. took [23] a step further and presented a novel controlled NL interface to *temporal databases*, based on translating NL questions into *SQL/Temporal*, a temporal database query language. They noted that their translation method is considerably simpler than previous attempts in this direction.

2.3 Conclusions

In this literature review we surveyed several major field in Computer Science. We showed that each of these fields has been advanced considerably over the last half-century, especially the domain of ontologies. However, we did not find a research that would satisfy all of the following criteria:

1. Employ either a temporal ontology or a temporal database to store knowledge of temporal family relations.

2. Propose a solution for effective navigation in a genealogical tree.

3. Design and implement a text parser for querying temporal ontologies in natural language.

Although there were articles that partially fulfill some of these requirements, none of them satisfied all. This entails the novelty of our work.

Chapter 3

Methodology

...

Referencing other chapters 2, 3, 4, 5 and 6

...

Chapter 4

Implementation

...

Chapter 5

Evaluation and Discussion

...

Chapter 6

Conclusion

...

Bibliography

[1] B. C. Smith, "Reflection and semantics in a procedural language," Ph.D. dissertation, MIT, Cambrige, 1982.

[2] H. J. Levesque and R. J. Brachman, "Expressiveness and tractability in knowledge representation and reasoning," *Comput. Intell.*, vol. 1, no. 3, pp. 78–93, 1987.

[3] T. Winograd, "Procedures as a representation for data in a computer program for understanding natural language," Ph.D. dissertation, MIT, Cambrige, 1971.

[4] H. Morowitz, "The plural of 'ontology' is 'confusion'," *Wiley Periodicals*, vol. 17, no. 6, 2012.

[5] B. Chandrasekaran and J. R. Josephson, "What are ontologies, and why do we need them?" *IEEE Intelligent Systems*, 1999.

[6] T. M. Ting, "Building a family ontology to meet consistency criteria," Master's thesis, University of Tun Hussien, 2015.

[7] T. F. James Bailey, Francois Bry and S. Schaffert, "Web and semantic web query languages: A survey."

[8] M. Marx, "Xcpath, the first order complete xpath dialect."

[9] C. Lai and S. Bird, "Querying linguistic trees," 2009.

[10] A. K. V. R. F. W. Alessandro Artale, Roman Kontchakov and M. Zakharyaschev, "Ontology-mediated query answering over temporal data: A survey," *24th International Symposium on Temporal Representation and Reasoning*, vol. 1, no. 1, pp. 1–37, 2017.

[11] M. Minsky, "A framework for representing knowledge," 1974.

[12] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, 1983.

[13] Y. Shohan, "Temporal logic in ai: Semantical and ontological considerations," *Artificial Intelligence*, vol. 33, no. 1, pp. 89–104, 1987.

[14] A. Schmiedel, "A temporal terminological logic," *AAAI-90 Proceedings*, vol. 1, no. 1, pp. 640–645, 1990.

[15] R. J. Brachman and J. G. Schmolze, "An overview of kl-one knowledge representation system," *Cognitive Science*, vol. 9, no. 1, pp. 171–216, 1985.

[16] L. L. Pablo Barcelo, "Temporal logic over unranked trees," 2008.

[17] A. Tuzhilin and J. Clifford, "A temporal relational algebra as a basis for temporal relational completeness," *Proceedings of the 16th VLDB Conference*, 1990.

[18] M. S. Perry, "A framework to support spatial, temporal and thematic analysis over semantic web data," Ph.D. dissertation, University of Georgia, 2008.

[19] S. B. N. M. Evdoxios Baratis, Euripides G.M. Petrakis and N. Papadakis, "Toql: Temporal ontology query language."

[20] B. Couderc and J. Ferrero, "Fr2sql : database query in french," *22eme Traitement Automatique des Langues Naturelles*, 2015.

[21] S. M. F.Siasar djahantighi, M.Norouzifard, "Using natural language processing in order to create sql queries," *Proceedings of the International Conference on Computer and Communication Engineering*, 2008.

[22] P. P. Chaudhari, "Natural language statement to sql query translator," *International Journal of Computer Applications*, vol. 82, no. 5, 2013.

[23] R. Nelken and N. Francez, "Querying natural language databases using controlled natural language," 2001.

Appendix A

Extra Stuff

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Appendix B

Even More Extra Stuff

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.