Parsing Natural Language Queries for Extracting Data from Large-Scale Geospatial Transportation Asset Repositories

Tuyen Le[1]; H. David Jeong[2]; Stephen B. Gilbert[3]; and
Evgeny Chukharev-Hudilainen[4]

[1]Postdoctoral Research Associate, Dept. of Civil, Construction and Environmental Engineering, Iowa State Univ., Ames, IA 50011. E-mail: ttle@iastate.edu
[2]Associate Professor, Dept. of Civil, Construction and Environmental Engineering, Iowa State Univ., Ames, IA 50011. E-mail: djeong@iastate.edu
[3]Assistant Professor, Dept. of Industrial and Manufacturing Systems Engineering, Iowa State Univ., Ames, IA 50011. E-mail: gilbert@iastate.edu
[4]Assistant Professor, Applied Linguistics and Technology Program, Iowa State Univ., Ames, IA 50011. E-mail: evgeny@iastate.edu

Abstract

Recent advances in data and information technologies have enabled extensive digital datasets to be available to decision makers throughout the life cycle of a transportation project. However, most of these data are not yet fully reused due to the challenging and time-consuming process of extracting the desired data for a specific purpose. Digital datasets are presented only in computer-readable formats and they are mostly complicated. Extracting data from complex and large data sources is significantly time-consuming and requires considerable expertise. Thus, there is a need for a user-friendly data exploration framework that allows users to present their data interests in human language. To fulfill that demand, this study employs natural language processing (NLP) techniques to develop a natural language interface (NLI) which can understand users' intent and automatically convert their inputs in the human language into formal queries. This paper presents the results of an important task of the development of such a NLI that is to establish a method for classifying the tokens of an ad-hoc query in accordance with their semantic contribution to the corresponding formal query. The method was validated on a small test set of 30 plain English questions manually annotated by an expert. The result shows an impressive accuracy of over 95%. The token classification presented in this paper is expected to provide a fundamental means for developing an effective NLI to transportation asset databases.

INTRODUCTION

Data has become a critical component of any transportation asset management (TAM) program. Asset data are used for predicting the performance, determining maintenance activities and resource allocation plans, and analyzing design methods. In attempts to support TAM, various national and state programs have been initiated and collected numerous types of asset data, for instance, asset inventory, condition, traffic, and materials of public civil infrastructures for decades. Examples of those efforts include the Highway Performance Monitoring Systems (HPMS), the Longterm Pavement Performance (LTPP) program, and the National Bridge Inventory (NBI) program. These programs require State Departments of Transportation (DOTs) to provide annual or bi-annual submissions of public transportation asset datasets with millions of records and hundreds of attributes to the federal agencies. As a result of these efforts, a huge amount of data has been captured and become publicly accessible.

Extracting data from such large databases for a particular domain of interest is an important task of data analytics. The state-of-practices on digital data retrieval in the civil infrastructure domain, however, still rely on laborious and time-consuming methods which have imposed big burdens on professionals (Khattak et al. 2015). Users are required to have a deep understanding of data structures, meanings behind each data label and query languages. As formal query languages are difficult to learn especially for non-computer-professionals (Androutsopoulos et al. 1995), data extraction has become a big hurdle to data utilization in the civil infrastructure sector. To assist users in extracting the desired data, several graphical and form-based interfaces [e.g., LTPP InfoPave (LTPP InfoPave 2017)] to asset databases have been developed. These systems offer an easier way where the users can just click on a predefined form to filter or restrict the range of the data. However, to be able to use those visual and interactive systems, users are required to learn the interaction manner with computers, for instance, how to select constraints, and to understand the data structure (Androutsopoulos et al. 1995). In addition, for complex data restrictions, formal query codes are still needed to express data needs. Thus, the ability to extract the desired data from large-scale databases without an extensive understanding of the contents and structure of each database is apparently beneficial to the acquisition of transportation data (Seedah and Leite 2015).

Natural language interfaces to databases (NLIDB) are systems that allow the end users to use human language questions to express the data of interests. Those systems do not require any knowledge of a query language or a database schema. With an ideal NLIDB, data can be quickly extracted with minimized efforts (Androutsopoulos et al. 1995). This paper presents an initial work of an ongoing project that aims to develop a NLIDB system for different types of users including non-technical experts to quickly and easily obtain the desired transportation asset data. Particularly, the focus of this paper is to discuss the method of aligning the tokens of an ad-hoc query into different parts of a spatial query language (POQ). This study employs an integrated method of rules and ontology to identify POQ for each of the tokens. The following sections will first discuss a brief review of related studies and then explain the token classification in detail.

BACKGROUND

A natural language query interface to a database (NLIDB) is a computer system that allows users to use natural language (e.g., English) to express their data extraction needs. NLIDB is able to automatically translate human language questions to a formal machine-readable query language such as SQL and XQuery. An ideal NLIDB allows users to obtain the desired information or a subset of data without requiring knowledge of a computer query language or the data schema (Li et al. 2007). Two typically main tasks of NLIDBs include (1) understanding of the semantic of users' queries and (2) translating the query into a formal language which can be executed by computers. The semantic understanding and the ambiguity of human language is a typical challenging task of a NLIDB. The performance of NLIDBs is largely dependent on how close the user's intent is interpreted. Recent advancements in Natural Language Processing (NLP) techniques and semantic resources (e.g., ontology, lexicon) have significantly improved the performance in interpreting user's intent through their ad-hoc questions.

In the last decade, a plethora of research have been conducted to developed natural language query systems for various different query languages including SQL (Popescu et al. 2003, Saha et al. 2016), XQuery (Li et al. 2006, Li et al. 2007a, Li et al. 2007b), and Sparql (Zou et al. 2014, Dubey et al. 2016). These systems have been implemented for different types of data such as bibliographic information (Zhu et al. 2016), spatial analysis of crimes (Zhang et al. 2009), biological data (Jamil 2017), and geospatial maps (Cai et al. 2005, Du et al. 2005, Lawrence et al. 2016, Haas et al. 2016). The communication between users and computers varies among NLIDB systems. For example, PRECISE (Popescu et al. 2003) is a one-way interaction system that returns a result only for those requests successfully processed. NALIX (Li et al. 2006, Li et al. 2007a, Li et al. 2007b) and NALIR (Li et al. 2014a, Li et al. 2014b, Li et al. 2016) systems, conversely, allow users to provide feedbacks for sophisticated queries or when a semantic ambiguity occurs to reprocess the query. Another interactive system is ATHENA (Saha et al. 2016) which returns the users a ranked list of the results of all possibly-interpreted query codes. With these interactive mechanisms, interpretation errors for complex queries can be eliminated. Many commercial systems can understand a certain level of generic human commands towards computer devices such as Apple Siri, Amazon Alexa, and Google Home. Those systems, however, are not able to support data query from databases yet.

The state-of-the-art NLIDB systems are mainly generic and are not working well for the domain databases. The parsers used by NLIDBs are usually trained with open-domain corpus, while queries are typically domain-specific questions. Since parsing largely impacts the performance of NLIDBs, errors in this state may cause many issues for the following operations (Li and Rafiei 2017). Natural language query parsing is still an open problem to specific domains. In the civil infrastructure, this area of research is extremely limited. The most notable effort in the transportation domain is made by Seedah and Leite (2015) who developed a method for entity name recognition from freight-related natural queries. However, no framework that can deal with the semantics of queries and convert them into a formal language has been developed for the transportation data.

PROPOSED ARCHITECTURE

The purpose of this study is to develop a natural language interface for extracting data from large-scale transportation asset data. Given a natural language query *"What is the total length of all road sections in Story county in Iowa?",* the system will generate a corresponding query in a spatial query language. In this study, a spatial query language is selected because geo-located data has become a widely accepted data format to store transportation asset information. The target spatial query language selected in this study is PostGIS which is an extension for querying spatial data stored in the PostgreSQL database service.

The proposed architecture of the system for translating natural language queries into formal queries is depicted in Figure 1. The method includes three main modules. Firstly, the *token classification* module breaks the input request into separate tokens and classify them based on their roles (e.g., values, operators, attributes, time, spatial relation) in the spatial query. This task utilizes the information in the asset ontology along with a set of rules to detect the part of query (POQ) for each token. In stage 2, the semantic dependencies between tokens will be determined. Relation triples will be correspondingly identified. A triple includes *subject, predicate,* and *object*. For example, the relation between an attribute (subject) and a value (object) through an operator (predicate) can be expressed as the following triple '*length - (greater than) - 400'*. In the final step, the triple set generated from the previous step will be utilized to construct a formal query in PostGIS. Following this state, the query will be executed by the PostgreSQL server.

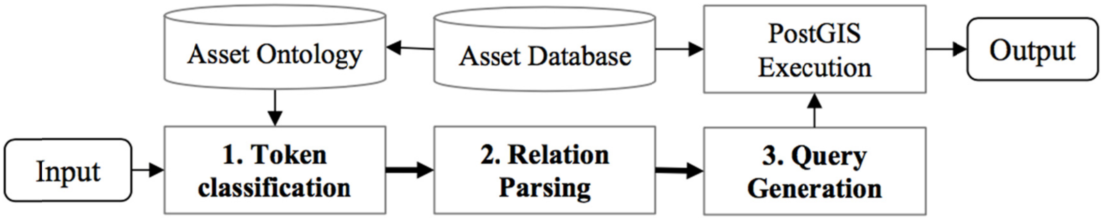

Figure 1. Natural Language Query Processing Architecture

This paper focuses on the first module of the proposed system which is to classify the tokens of a plain English query in accordance with their roles in the spatial query language. Figure 2 illustrates the tokenization and token classification process. The algorithm uses an integrated source of information from a domain ontology and a rule set to detect POQs. Sections bellows discuss different types of token classes and the procedure to tokenize and tag natural questions in detail.

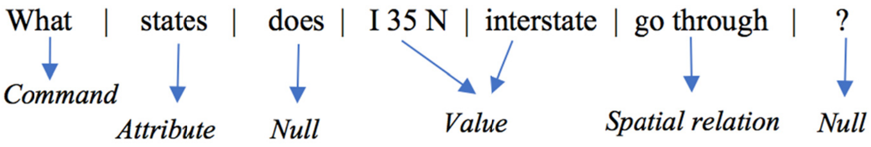

Figure 2. Example of tokenization and token classification

Part of Query

This study aims to develop an algorithm for both geo-spatial and non-geospatial data query over a transportation asset database. Table 1 below defines various POQ tags and their corresponding elements of a spatial query. The algorithm for identifying POQ tags is shown in Algorithm 1. There are three steps in the algorithm including tokenization, *uni-gram* tagging pass, and *bi-gram* tagging pass. Both of these tagging passes employ a rule-based approach to determine the tag of a particular token. The details of these steps are explained as follows.

Key phrase index and ontology

Keywords are an important indicator for a specific POQ tag. For example, the phrase ‘greater than’ implies an operator. In this study, various indexed sets of key phrases for different types of query elements are developed. In addition to general keywords of a spatial query language, domain-specific ontology was also developed to support the recognition of data attributes (e.g., rutting, AADT), places (e.g., New Jersey, New York), route names (e.g., I 35 N). The ontology, as summarized in Table 2, includes concepts and properties adopted from the HPMS data schema. Their instances, for example, roadway names and states are from HPMS datasets.

Table 1. Part of Query Tags [Modified from NaLIX (Li et al. 2007b)]

POQ Tag	Query Element	Rules	Rule Type
Command (CMT)	Select	Matching a keyword	Uni-gram
Order by (OBT)	Order By	A superlative adjective, and matching a keyword	Uni-gram
Order option (OOT)	ASC or DESC	Matching a keyword	Uni-gram
Group by (GT)	Group By	‘by’ followed by *AT*	Bi-gram
Function (FT)	Function	Matching a keyword	Uni-gram
Operator (OT)	Operator	Matching a keyword	Uni-gram
Value (VT)	Value	A number, date, instance of ontology	Uni-gram
Attribute (AT)	Variable	A property of ontology	Uni-gram
Table (TT)	Table	A concept of ontology	Uni-gram
Logic (LT)	Logic operator	Matching a keyword	Uni-gram
Quantifier (QT)	Quantifier	Matching a keyword	Uni-gram
Spatial Relation (SRT)	Spatial Relation	Matching a keyword and followed by an AT or string VT	Bi-gram
Null	-	Stop words	Uni-gram

Algorithm 1.

```
Input: Plain English question $nlq$
Output: Tagged token set $Q$
Initialize
Token Set $T \leftarrow$ Tokenization ($nlp$)
POS Tagged $S \leftarrow$ POS Taging ($T$)
POQ Tagged Tokens $Q \leftarrow$ Uni-gram Pass ($S$)
POQ Tagged Tokens $Q \leftarrow$ Bi-gram Pass ($Q$)
End
```

Table 2. Summary of roadway asset ontology

Element	Count
Class	229
Object property	39
Data property	26
Individual	209
Equivalent class	21

Tokenization and Part of Speech Tagging

This phase aims to break natural language queries into separate tokens, determine their part of speech (POS) tags, and identify dependencies among tokens. Given a natural language query in Figure 2, the question will be broken into seven different tokens. To obtain this result, the question is first tokenized using a generic NLP tokenizer. In this study, OpenNLP Tokenizer is utilized. Since a general tokenizer does not recognize entity names (e.g., *I 35 N*), multi-word terms or key phrases (e.g., *go through*) as a single token, sequences of tokens matching a key phrase in the indexed set are connected through an underscore symbol '_' to create multi-element tokens. The preprocessed question is then tagged using a generic NLP tagging technique. The output of this stage is a set of tokens along with their POS tags which will be further analyzed in the *uni-gram* and *bi-gram* POQ tagging passes.

Uni-gram Pass

The POQ tagging rules in this study can be classified into *uni-gram rules* and *bi-gram rules*. Uni-gram rules are those that rely on the token itself and its POS tag to determine its POQ tag without considering information of neighboring tokens. Conversely, in the bi-gram pass, the POQ of a token is identified based on the information of tokens that occur before and after it. Of those POQ tags shown in Table 1, all of them except for 'Group By' and 'Spatial Relation' tags are identified in this uni-gram pass. The uni-gram pass applies all the uni-gram rules on the POS tagged tokens. In this phase, other tokens that are not satisfied any uni-gram rules are temporarily assigned to the 'Null' tag. The temporary result from this phase is used as the input of the bi-gram pass.

Bi-gram Pass

The bi-gram pass aims to re-identify POQ tags for those tokens processed in the uni-gram pass. In this pass, bi-gram rules are applied. In specific, this pass is to resolve ambiguity for a certain token by taking into account the context token which occurs right after it. For 'Group By' tokens, the word 'by' is a good trigger (e.g., '*show me total length by state'*). However, the semantics of 'by' itself is ambiguous unless its context is considered. For example, in the request '*list all roads administered by state agencies'*, it is an error to assign the 'Group By' tag to the word 'by'. Similarly, for the spatial relation, the word 'in' (e.g., show all routes in Iowa) can either refer to the *contain* spatial relation or to a *date* value (e.g., show routes constructed in 2001). In this stage, given a pair of a token and its POQ tag resulting from the previous pass, its tag is re-classified by applying the bi-gram rules provided in Table 1.

IMPLEMENTATION AND EVALUATION

Experiment Setup

In order to evaluate the performance of the token classification method, an experiment was conducted. In this experiment, the method was applied on a testing data set of 30 different natural questions about pavement asset that are manually annotated. Since obtaining natural questions for testing purposes is a hard task, in this study we adopted the natural language queries from the GeoQue dataset (Zelle and Mooney 1996) in the geographical domain. To construct this data set, 30 questions of the GeoQue dataset was randomly selected, and then modified in accordance with the pavement management domain. When modifying the original questions, their structures of natural questions are adopted, only the target and constraint attributes are changed in accordance with pavement asset databases. For example, the question '*What rivers go through more than three states?*' is modified into '*What roadways go through more than three states?*' where only the word 'rivers' is replaced with 'roadways'. These modified questions were then tagged by an expert. The expert is required to manually tokenize the questions and determine POQ for each of the tokens.

A Java prototype based on the proposed method was developed to support evaluation. The uni-gram and bi-gram rules were manually encoded in the Java program. The method was evaluated by accuracy (see Equation 1) which represents the percentage of tokens are correctly tagged. The system was tested with four different models including (1) without bi-gram pass, (2) with bi-gram pass, (3) without ontology, and (4) with ontology.

$$Accuracy = \frac{Correctly\ Tagged\ Tokens}{Total\ tokens} \quad (1)$$

Performance Evaluation

Table 3 below shows the summary of the results of the experiment. As shown in the table, the model achieves an impressive accuracy of over 95%. A comparison between different models evidently shows that the inclusion of the bi-gram pass and the domain ontology significantly enhance the system performance. Specifically, the bi-gram pass enables an accuracy increase of about 10%. The integration of domain knowledge increases the system performance by around 12% for both 'uni-gram pass only' and 'with bi-gram pass' models. However, these results are based on small dataset which consists of only 30 questions. In future study, we plan to construct a larger test dataset to validate the reliability of the system. Another opportunity for future improvement is to include synonyms into those attributes and concept names in the domain ontology. Finally, this token classification can be further implemented to develop a complete natural language interface for querying transportation asset data.

Table 3. Performance of POS tagging

Model	Accuracy (%)
Uni-gram pass without ontology	72.39
Uni-gram pass with ontology	84.66
Bi-gram pass without ontology	82.21
Bi-gram with ontology	95.09

CONCLUSIONS

Data extraction is one of the biggest technical burdens imposed on professionals in the area of transportation asset management. In order to obtain precise information from large and complex datasets, the end users are required to have considerable knowledge of spatial query language and the data schema. This paper presents the preliminary results of an ongoing research that aims to develop a natural language interface to transportation asset databases. This paper focuses on developing an algorithm for determining the POQ tag for each of the tokens of a plain English request. The algorithm involves a set of rules and a domain ontology to support the token classification.

An experiment on a small dataset of 30 natural language queries indicates an impressively high accuracy of over 95%. Future research, however, is still needed to test the system on a larger corpus and to include synonyms of terms into the rule-based classifier.

This study is expected to fundamentally transform the way in which professionals interact with large and complex datasets. This study provides a foundational platform for further studies to develop an interface that enables users to express data queries in plain English. Once data acquisition becomes readily available and easy to be obtained with little effort, more focus can be distributed to improving the quality of data analytics and decision making.

ACKNOWLEDGEMENTS

This research was funded by the National Science Foundation (NSF) under the Award Number 1635309. The authors gratefully acknowledge NSF's support. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

REFERENCES

Androutsopoulos, I., Ritchie, G. D., & Thanisch, P. (1995). Natural language interfaces to databases -- an introduction. Natural Language Engineering, 1, 29-81. doi:10.1017/S135132490000005X

Cai, G., Wang, H., MacEachren, A. M., & Fuhrmann, S. (2005). Natural Conversational Interfaces to Geospatial Databases. Transactions in GIS, 9, 199-221. doi:10.1111/j.1467-9671.2005.00213.x

Du, S., Qin, Q., Chen, D., & Wang, L. (2005). Spatial data query based on natural language spatial relations. Geoscience and Remote Sensing Symposium, 2005. IGARSS'05. Proceedings. 2005 IEEE International, 2, pp. 1210-1213.

Dubey, M., Dasgupta, S., Sharma, A., Höffner, K., & Lehmann, J. (2016). AskNow: A Framework for Natural Language Query Formalization in SPARQL. In H. Sack, E. Blomqvist, M. dÁquin, C. Ghidini, S. P. Ponzetto, & C. Lange (Eds.), The Semantic Web. Latest Advances and New Domains: 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 -- June 2, 2016, Proceedings (pp. 300-316). Cham: Springer International Publishing. doi:10.1007/978-3-319-34129-3_19

Haas, C., & Riezler, S. (2016, June). A Corpus and Semantic Parser for Multilingual Natural Language Querying of OpenStreetMap. To appear in Proceedings of the North American Chapter of the Association for Computational Linguistics {(NAACL)}. San.

Jamil, H. M. (2017). Knowledge Rich Natural Language Queries over Structured Biological Databases. CoRR, abs/1703.10692. Retrieved from http://arxiv.org/abs/1703.10692

Khattak, A. J., Wang, X., Son, S., & Liu, J. (2015). Data Needs Assessment for Making Transportation Decisions in Virginia. Tech. rep.

Lawrence, C., & Riezler, S. (2016). NLmaps: A Natural Language Interface to Query OpenStreetMap. COLING (Demos), (pp. 6-10).

Li, F., & Jagadish, H. V. (2014a). Constructing an Interactive Natural Language Interface for Relational Databases. Proc. VLDB Endow., 8, 73-84.

Li, F., & Jagadish, H. V. (2014b). NaLIR: An Interactive Natural Language Interface for Querying Relational Databases. Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (pp. 709-712). New York, NY, USA: ACM. doi:10.1145/2588555.2594519

Li, F., & Jagadish, H. V. (2016). Understanding Natural Language Queries over Relational Databases. SIGMOD Rec., 45, 6-13. doi:10.1145/2949741.2949744

Li, Y., & Rafiei, D. (2017). Natural Language Data Management and Interfaces: Recent Development and Open Challenges. Proceedings of the 2017 ACM International Conference on Management of Data (pp. 1765-1770). New York, NY, USA: ACM. doi:10.1145/3035918.3054783

Li, Y., Chaudhuri, I., Yang, H., Singh, S., & Jagadish, H. V. (2007a). Enabling domain-awareness for a generic natural language interface. AAAI, (pp. 833-838).

Li, Y., Yang, H., & Jagadish, H. V. (2006). Constructing a generic natural language interface for an XML database. EDBT, 3896, pp. 737-754.

Li, Y., Yang, H., & Jagadish, H. V. (2007b). NaLIX: A Generic Natural Language Search Environment for XML Data. ACM Trans. Database Syst., 32.

LTPP InfoPave (2017). <https://infopave.fhwa.dot.gov/> (August 27, 2017).

Popescu, A.-M., Etzioni, O., & Kautz, H. (2003). Towards a Theory of Natural Language Interfaces to Databases. Proceedings of the 8th International Conference on Intelligent User Interfaces (pp. 149-157). New York, NY, USA: ACM. doi:10.1145/604045.604070

Saha, D., Floratou, A., Sankaranarayanan, K., Minhas, U. F., Mittal, A. R., & Özcan, F. (2016, #aug#). ATHENA: An Ontology-driven System for Natural Language Querying over Relational Data Stores. Proc. VLDB Endow., 9, 1209-1220. doi:10.14778/2994509.2994536

Seedah, D. P., & Leite, F. (2015). Information Extraction for Freight-Related Natural Language Queries. Computing in Civil Engineering 2015. United: ASCE. doi:10.1061/9780784479247.053

Zelle, J. M., & Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. Proceedings of the national conference on artificial intelligence, (pp. 1050-1055).

Zhang, C., Huang, Y., Mihalcea, R., & Cuellar, H. (2009). A natural language interface for crime-related spatial queries. 2009 IEEE International Conference on Intelligence and Security Informatics, (pp. 164-166).

Zhu, Y., Yan, E., & Song, I.-Y. (2016). A natural language interface to a graph-based bibliographic information retrieval system. CoRR, abs/1612.03231. Retrieved from http://arxiv.org/abs/1612.03231

Zou, L., Huang, R., Wang, H., Yu, J. X., He, W., & Zhao, D. (2014). Natural Language Question Answering over RDF: A Graph Data Driven Approach. Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (pp. 313-324). New York, NY, USA: ACM. doi:10.1145/2588555.2610525