Enabling natural language analytics over relational data using Formal Concept Analysis

C. Anantaram, Mouli Rastogi, Mrinal Rawat, and Pratik Saini

TCS Research, Tata Consultancy Services Ltd, Gwal Pahari, Gurgaon, India
(c.anantaram; mouli.r; rawat.mrinal; pratik.saini) @tcs.com

Abstract. Analysts like to pose a variety of questions over large relational databases containing data on the domain that they are analyzing. Enabling natural language question answering over such data for analysts requires mechanisms to extract exceptions in data, find steps to transform data, detect implications in the data, and apply classifications on the data. Motivated by this problem, we propose a semantically enriched deep learning pipeline that supports natural language question answering over relational databases and uses Formal Concept Analysis to find exceptions, classification and transformation steps. Our framework is based on a set of deep learning sequence tagging networks which extracts information from the NL sentence and constructs an equivalent intermediate sketch, and then maps it into the actual tables and columns of the database. The output data of the query is converted into a lattice structure which results into the (extent,intent) tuples. These tuples are then analyzed to find the exceptions, classification and transformation steps.

1 Introduction

Data analysts have to deal with a large number of complex and nested queries to dig out hidden insights from the relational datasets, spread over multiple files. Extraction of the relevant result corresponding to a given query can be easily done through a deep learnt NLQA framework, but to detect further explanations, facts, analysis and visualizations from queried output is a challenging problem. This kind of data analysis over query's result can be handled by Formal Concept Analysis, a mathematical tool that results in a concept hierarchy, makes semantical relations during the queries, and also can find the implications as well as asociations in the given dataset, can unify data and knowledge and is capable of information engineering as well as data mining. So for enabling NL analytics over such datasets for analysts, we present in this paper, a semantically enriched deep learning pipeline that a) enables natural language question answering over relational databases using a set of deep learnt sequence tagging networks, and b) carries out regularity analysis over the query results using Formal Concept Analysis to interactively explore, discover and analyze the hidden structure in the selected data [12] [11]. The deep learnt sequence tagging pipeline extracts information from the NL sentence and constructs an equivalent intermediate

sketch, and then uses that sketch to formulate the actual database query on the relevant tables and columns. Query results are used in Formal Concept Analysis to create a lattice structure of the objects and attributes. The obtained lattice structure is then used to find exceptions in the data, classification of a new object and also to find the set of steps to transform the data from one structure to another structure.

2 Formal Concept Analysis

Formal Concept Analysis provides a theoretical framework for learning hierarchies of knowledge clusters called formal concepts. A basic notion in FCA is the formal context. Given a set G of objects and a set M of attributes (also called properties), a formal context consists of a triple (G, M, I) where I specifies (Boolean) relationships between objects of G and attributes of M , i.e., I $\subseteq$ G $\times$ M .Usually, formal contexts are given under the form of a table that formalizes these relationships. A table entry indicates whether an object has the attribute, or not. Let $I(g) = \{m \in M; (g, m) \in I\}$ be the set of attributes satisfied by object g , and let $I(m) = \{g \in G; (g, m) \in I\}$ be the set of objects that satisfy the attribute m . Given a formal context (G, M, I) . Two operators $()'$ define a Galois connection between the powersets (P(G),$\subseteq$) and (P(M),$\subseteq$), with A$\subseteq$G and B$\subseteq$M:

$$A' = \{m \in M | \forall g \in A : gIm\}$$

and

$$B' = \{g \in G | \forall m \in B : gIm\}$$

.
That is to say, A$'$ is the set of all attributes which is satisfied all objects in A , whereas B$'$ is the set of all objects which satisfies all attributes in B . A formal concept of (G,M,I) is defined as a pair (A,B) with A$\in$G , B$\in$ M , A$'$=B and B$'$=A. A is called the extent of the formal concept (A,B), whereas B is called the intent.The set of all formal concepts of (G, M, I) equipped with a subconcept-superconcept partial order $\leq$ is the concept lattice denoted by $\mathcal{L}$. The and is defined as:
For A_1,$A_2\subseteq$ G and B_1,$B_2\subseteq$ M

$$(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2 (equivalent to B_2 \subseteq B_1)$$

In this case, the concept (A_1, B_1) is called sub-concept and the concept (A_2, B_2) is called super-concept.

2.1 Association and Implication Rules

Given a formal context (G,M,I) there are extracted exact rules and approximate rules (rules with statistical values, for example, support and confidence).

These rules express in an alternative way the underlying knowledge. These rules are significant as they expresses the underlying knowledge of interaction among attributes.The exact rules are classified as implication rules while the approximation rules are classified as association rules.
Definition Given a formal context whose attributes set is M. An implication is an expression S $\Longrightarrow$ T, where S,T $\subseteq$ M. An implication S $\Longrightarrow$ T, extracted from a formal context, or respective concept lattice, have to be such that S$'$ $\subseteq$ T$'$. In other words: every object which has the attributes of S, also have the attributes of T. If X is a set of attributes, then X respects an implication S $\Longrightarrow$ T iff S $\not\subseteq$ X or T $\not\subseteq$ X. An implication S $\Longrightarrow$ T holds in a set $\{X_1, ..., X_n\} \subseteq$ M iff each X_i respects S $\Longrightarrow$ T.
Definition Given a threshold minsupp $\in$ [0, 1], where the support

$$supp(X) := \frac{card(X')}{card(G)}(withX' := g \in G|\forall m \in X : (g, m) \in I),$$

association rules are determined by mining all pairs X $\Longrightarrow$ Y of subsets of M such that

$$supp(X \implies Y) := supp(X)$$

is above the threshold minsupp, and the confidence

$$conf(X \implies Y) := \frac{supp(X \cup Y)}{supp(X)}$$

is above a given threshold minconf $\in$ [0, 1].

3 Methodology

We present a novel approach where a natural language sentence is converted into the sketch (Listing 1.1) which uses deep learning models and then further using the sketch to construct the database query (SQL) and fetch the output. This output is then taken to derive some explanations or interesting facts, find outliers or exceptions and rationalize the queried data if required (fig:1).
In order to generate the query sketch, we have a pipeline of multiple sequence tagging deep neural networks: Predicate Finder Model (Select Clause), Entity Finder Model (Values in Where Clause), Meta Type Model, Operators and Aggregation Model (all using bi-directional LSTM network along with a CRF (conditional random field) output layer), where the natural language sentence is processed as a sequence tagging problem.
The architecture uses an ELMO embedding that are computed on top of two-layer bidirectional language models with character convolutions as a linear function of the internal network states [16]. Also the character-level embedding is used as it has been found helpful for specific tasks and to handle the out-of-vocabulary problem. The character-level representation is then concatenated with a word-level representation and feed into the bi-directional LSTM as input. In the next step, a CRF Layer yielding the final predictions for every word is

used [8]. We have Z = $(z_1; z_2; ...; z_n)$ as the input sentence and P to be the scores output by Bi-LSTM network. $Q_{i,j}$ is the score of a transition from *tag* i to *tag* j for the sequence of predictions $Y = (y_1; y_2; ...; y_n)$. Finally the score is defined as :

$$s(Z; Y) = \sum_{i=0}^{n} Q_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

Models details

To generate the query sketch we use four different models using the same architecture (BiLSTM-CRF) [17] explained above, where the natural language sentence is processed as a sequence tagging problem. The neural network then predicts the tag for each word using which predicates, entities, and values in the sentence are identified, and an intermediate Sketch (independent of underlying database) is created. The Sketch is then mapped into the columns of the tables with conditions to construct the actual SQL query. In the sketch generation process the order of the models matters as the input of the next model depends on the output of previous model. To train the models, we had to create the annotations. In the cases where predicate/entities present in the sentence got the direct match with columns or values present in the actual database, we extracted them using a script and in the rest of the cases we have manually annotated the data.

- **Predicate Finder Model(Select Clause):** This model identifies the target concepts (predicates) from the NL sentence. In case of database query language, predicate refers to the SELECT part of the query. Once predicates are identified, it becomes easier to extract entities from the remaining sentence.
- **Entity Finder Model(Values in Where Clause):** This model identifies the relations(values/entities) in the query. In some cases the model misses/-capture some words. To tackle this issue predicted value in the `Apache-Solr` is searched. The structured data for the domain is assumed to be present in Lucene. After the search we picked the entity from the database which has the highest similarity score.
- **Meta Type Model:** This model identifies the type of concepts (predicates and values) at the node or table level. If a concept is present in more than one table, type information helps in the process of disambiguation. This helps in making the overall framework domain agnostic.
- **Aggregations and Operators Model:** In this model, aggregations and operators are predicted for predicates and entities respectively. Our framework currently supports following set of aggregation functions: count, groupby, min, max, sum, asc sort, desc sort. Similarly, following set of operators are also supported: $=;>;<;<>;\geq;\leq;$*like*.

The models are trained independently and do not share any internal representations. However, the input of one model depends on the previous. For example, once predicates are identified we replace the predicate part in the NL sentence with some token before passing it to the next model. We capture this information from the NL sentence and create an intermediate representation (Sketch)

which is further passed to the query generator(neo4j knowledge graphs), to construct the SQL or another database query and yields results. Result table of the query is then converted into its equivalent formal context, which is a triplet of objects, attributes and incidence relation between them. This formal context is used to extract the implication and association rules [10] and create a concept lattice which derives all possible formal concepts from the context and orders them according to a subconcept-superconcept relationship [15]. This conceptual hierarchy of the queried output is further used for knowledge discovery that is implicitly present in it. Here we are focusing on three types of analysis over queried data from a relational database.

Listing 1.1: Sketch

```
{
"select":
[
{
"pred_hint": model
},
{
"pred_hint": horsepower,
"aggregation": desc_sort,
}]
"conditions":
{
"pred_hint": cylinders,
"value": 4,
"operator": =
}
}
```

3.1 Outliers Analysis

This is first type of analysis that could be perform in the queried output. Outliers are defined as rules that contradict common beliefs. These kind of rules can play an important role in the process of understanding the underlying data as well as in making critical decisions. Outliers Analysis is to uncover the exceptions hidden in the given query output. To perform this over the queried output, we firstly created a preliminary formal context from the given raw data. Then by using `Conexp tool` [13], implication and association rules are generated for complete dataset. These rules shows the correlation among different attributes. After the query is posed, concept lattice of the queried data is created and formal concepts in the form of (extent, intent) tuple are extracted from it. Intents of these formal concepts are then compared with the implication and association rules. If an intent of the queried output is violating any of the implication and association rules, then it is considered as an outlier for that query.

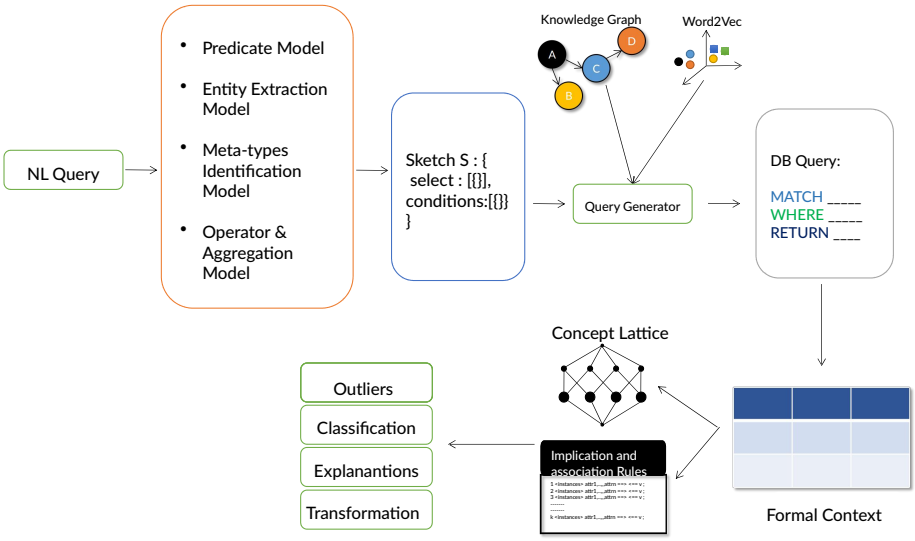


Fig. 1: High Level Architecture of the Process

3.2 Transformation Analysis

This is the second type of analysis that we introduced in our framework. Transformation analysis is used to measure two queries results, where tasks such as conversion of the underlying lattice structure of one set of query results into the lattice structure of another set of query results are required. This kind of analysis is performed by finding the difference between the intents of the formal concepts of both lattices. In our framework when two semantically enriched queries are posed, lattice structures of their respective outputs are generated. To find the possible transformation requirements, we match the intents of both concept lattices and put down the differences between them. This gives us the disparity in the kind of objects contained in both the lattices which will help in transforming one lattice to another.

3.3 Classification analysis

Classification analysis in our framework is done to predict the category of new objects. This is carried out by defining a target attribute **t** in the dataset, generating concept lattices C_i for each value v_i where **i** $\in \mathbb{N}$ of the target attribute and then comparing new object's attributes with the intents of each C_i. In this analysis, a query asking for object details is posed. Lattice structures C_i corresponding to each v_i is stored in the memory. At the run time, matching of new object's attributes set is done with intents of each C_i. If the intent of new object is contained in any one of the lattice C_j for some j $\in$ range(i), then the new

object is classified under the corresponding v_j category otherwise if more than one concept lattices contains the new object's intent then our framework cannot determine its category.

4 Experiments and Results

Census Income dataset taken from UCI machine learning repository [14] is used. This relational database contains 906 observations and 14 features of people like age, occupation, education, salary, workclass, native country etc. We construct the Neo4j knowledge graph from the csv ad also generated the implication and association rules. In this dataset we considered people names as the set of objects and applied conceptual scaling over the multivalued features mentioned above to generate the set of attributes where the objects and the attributes has a binary relation in between them.

Snapshot of the dataset is:

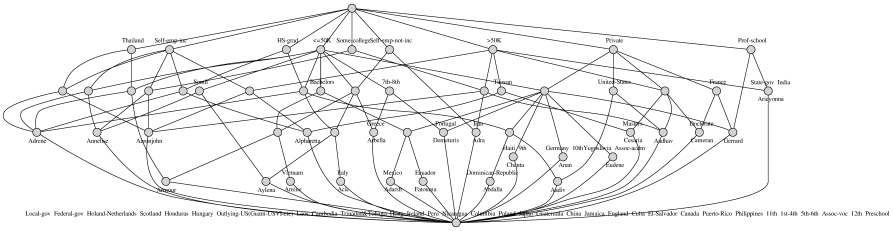
name	age	workclass	education	native country	salary
Aaban	39	State-gov	Bachelors	United-States	<=50K
Aabha	50	Self-emp-not-inc	Bachelors	United-States	<=50K
Aabid	38	Private	HS-grad	United-States	<=50K
Aabriella	53	Private	11th	United-States	<=50K
Aada	28	Private	Bachelors	Cuba	<=50K

Implication and association rules extracted from data are:

S.No.	rule	no. of instances
1	11th $\implies$ $\leq$50K	118
2	State-gov, 5th-6th $\implies$ $\leq$50K	45
3	Private, 10th $\implies$ $\leq$50K	63
4	Doctorate, State-gov $\implies$ >50K	17
5	Federal-gov, Masters $\implies$ >50K	41
6	Local-gov, 12th $\implies$ $\leq$50K	86
7	Bachelors $\implies$ >50K	178

1. Outliers Analysis

Query: List people working more than 60 hours per week and having exceptions in salary with respect to education.



Rules extracted from lattice are:

S.No.	rule
1	Gerrard↔[≤50K,Private,France,Prof-school]⇔Gerrard
2	Arbella↔[>50K,Private,Greece,10th]⇔Arbella⇔Greece
3	Amine↔[≤50K,Self-emp-not-inc,Vietnam,Bachelors]⇔Amine⇔Vietnam
4	Arieyonna↔[>50K,State-gov,India,Prof-school]⇔Arieyonna⇔State-gov,India
5	Adarsh↔[≤50K,Private,Mexico,Bachelors]⇔Adarsh⇔ Mexico
6	Aadhav↔[>50K,Private,United-States,Some-college]⇔Aadhav

Outliers

S.No.	rule
1	Arbella↔[>50K,Private,Greece,10th]⇔Arbella⇔Greece
2	Adarsh↔[≤50K,Private,Mexico,Bachelors]⇔Adarsh⇔ Mexico

Analysis

- Adarsh works >60 hours per week with salary ≤ $ 50 K and Bachelors Degree.
- Arbella works >60 hours per week with salary >$ 50 K and is only 10th grade.

2. Transformation Analysis

Query: What needs to be done to transform workclass, education and salary of men in Cuba to be like men in England?

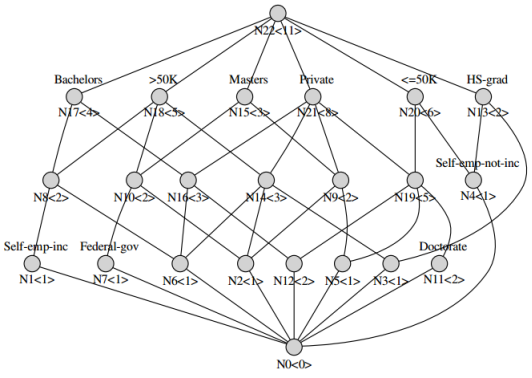


Fig. 2: England

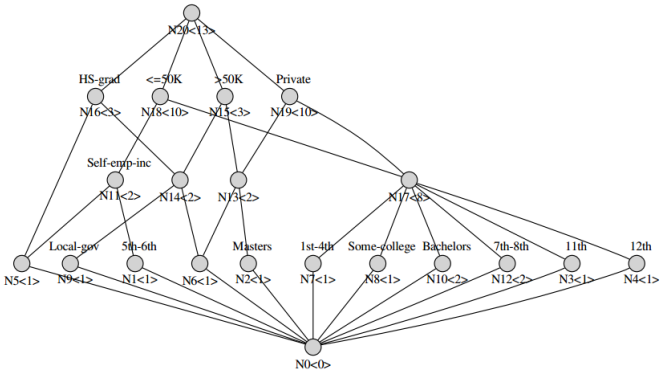


Fig. 3: Cuba

Intents need to be removed are:
a) (≤50K, Self-emp-inc, 5th-6th); b) (Private, >50K, Masters); c) (≤50K, Private, 11th); d) (≤50K, Private, 12th); e)(Private, ≤50K, 7th-8th); and f) (≤50K, Private, 1st-4th)
Intents need to be introduced are:
a) (>50K, Masters, Private); b) (Self-emp-inc, Bachelors, >50K); c) (>50K, Private, HS-grad); d) (Self-emp-not-inc, ≤50K, HS-grad); e) (Private, ≤50K, Masters); f) (Bachelors, >50K, Private); g) (>50K, Masters, Federal-gov); and h) (≤50K, Doctorate, Private)

It shows: Need of higher Education, Need of Self-Employment.

3. Classification Analysis
Query: Predict that whether Aarav has diabetes or not from his blood pressure, body mass index and age.

Person details	**Input from user**
enter name	Aarav
enter age	25
enter Blood Pressure	66
enter Body mass index	23.2

Based on the features of Aarav, it is predicted that he don't have diabetes.

5 Conclusion

We have described a framework wherein the NL sentence is semantically mapped into an intermediate logical form (Sketch) using the framework of multiple sequence tagging networks. This approach of semantic enrichment abstracts the low level semantic information from sentence and helps in generalising into various database queries (e.g. SQL, CQL). Answer of these queries are then further

interpreted using FCA to find out outliers, facts and explanations, classifications and transformations. Experimental results shows that how NLQA and FCA can help an analyst in discovering regularities in a complex data.

References

1. Amit Sangroya, Pratik Saini, Mrinal Rawat, Gautam Shroff, C. Anantaram: Natural Language Business Intelligence Question Answering through SeqtoSeq Transfer Learning, In: DLKT: The 1st Pacific Asia Workshop on Deep Learning for Knowledge Transfer, PAKDD, April(2019)
2. Victor Zhong, Caiming Xiong, Richard Socher: Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning, https://doi.org/arXiv:1709.00103, (2017)
3. Xuezhe Ma, Eduard H. Hovy:End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF.,*CoRR,abs/1603.01354*, http://arxiv.org/abs/1603.01354,https://doi.org/1603.01354, *dblp computer science bibliography, https://dblp.org*, (2016)
4. Shefali Bhat, C. Anantaram, Hemant K. Jain: Framework for Text-Based Conversational User-Interface for Business Applications. Knowledge Science, Engineering and Management, In: Second International Conference, KSEM Melbourne, Australia, *DBLP:conf/ksem/2007*, https://doi.org/10.1007/978-3-540-76719-0_31, https://doi.org/10.1007/978-3-540-76719-0_31, *https://dblp.org/rec/bib/conf/ksem/BhatAJ07*, November (2007)
5. Loper, Edward, Bird, Steven: NLTK: The Natural Language Toolkit, In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, Volume 1, *ETMTNLP '02*, pages: 63–70, https://doi.org/10.3115/1118108.1118117, https://doi.org/10.3115/1118108.1118117, Philadelphia, Pennsylvania, (2002)
6. Manning, Christopher D., Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven J., McClosky, David: The Stanford CoreNLP Natural Language Processing Toolkit, In: Association for Computational Linguistics (ACL) System Demonstrations,pages: 55–60, http://www.aclweb.org/anthology/P/P14/P14-5010, (2014)
7. Li, Fei, Jagadish, H. V.: Constructing an Interactive Natural Language Interface for Relational Databases, *Proc. VLDB Endow.*, volume: 8, pages: 73–84, http://dx.doi.org/10.14778/2735461.2735468, https://doi.org/10.14778/2735461.2735468, *VLDB Endowment*, September (2014)
8. Lample, Guillaume, Ballesteros, Miguel, Subramanian, Sandeep, Kawakami, Kazuya, Dyera, Chris: Neural Architectures for Named Entity Recognition, In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pages: 260–270, https://doi.org/10.18653/v1/N16-1030 http://aclweb.org/anthology/N16-1030, San Diego, California, (2016)
9. Dmitry I. Ignatov: Introduction to Formal Concept Analysis and Its Applications in Information Retrieval and Related Fields, Russian Summer School in Information Retrieval, December (2015)
10. K Sumangali, Ch Aswani Kumar: Determination of interesting rules in FCA using information gain, In: First International Conference on Networks and Soft Computing (ICNSC2014), IEEE, August (2014)

11. Peter D. Grnwald: The Minimum Description Length Principle, MIT Press, pages: 3-40, (2007)
12. Bernhard Ganter, Rudolf Wille: Formal Concept Analysis, Mathematical Foundations, Springer, Berlin,Heidelberg,New York, (1999)
13. Serhiy A. Yevtushenko: System of data analysis:Concept Explorer. (In Russian, In: Proceedings of the 7th national conference on Artificial Intelligence KII, pages: 127-134, Russia, (2000)
14. Dua, Dheeru, Graff, Casey: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences, (2017)
15. Ganter B., Wille R.: Formal concept analysis:mathematical foundations. Springer Science & Business Media, (2012)
16. Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer: Deep contextualized word representations. CoRR, abs/1802.05365, (2018)
17. Xuezhe Ma, Eduard H. Hovy: Endto-end sequence labeling via bi-directional lstm-cnns-crf, CoRR, abs/1603.01354, (2016)