Conversational BI: An Ontology-Driven Conversation System for Business Intelligence Applications

Abdul Quamar[1], Fatma Özcan [1], Dorian Miller[2],
Robert J Moore[1], Rebecca Niehus[2], Jeffrey Kreulen[2]
[1]IBM Research AI, [2]IBM Watson Health
[1]ahquamar|fozcan|rjmoore@us.ibm.com, [2]millerbd|rniehus|kreulen@us.ibm.com

ABSTRACT

Business intelligence (BI) applications play an important role in the enterprise to make critical business decisions. Conversational interfaces enable non-technical enterprise users to explore their data, democratizing access to data significantly. In this paper, we describe an ontology-based framework for creating a conversation system for BI applications termed as *Conversational BI*. We create an ontology from a business model underlying the BI application, and use this ontology to automatically generate various artifacts of the conversation system. These include the intents, entities, as well as the training samples for each intent. Our approach builds upon our earlier work, and exploits common BI access patterns to generate intents, their training examples and adapt the dialog structure to support typical BI operations. We have implemented our techniques in *Health Insights (HI)*, an IBM Watson Healthcare offering, providing analysis over insurance data on claims. Our user study demonstrates that our system is quite intuitive for gaining business insights from data. We also show that our approach not only captures the analysis available in the fixed application dashboards, but also enables new queries and explorations.

PVLDB Reference Format:
Abdul Quamar, Fatma Özcan, Dorian Miller, Robert J Moore, Rebecca Niehus and Jeffrey Kreulen. An Ontology-Based Conversation System for Knowledge Bases. *PVLDB*, 13(12): 3369-3381, 2020.
DOI: https://doi.org/10.14778/3415478.3415557

1. INTRODUCTION

Business Intelligence(BI) tools and applications play a key role in the enterprise to derive business decisions. BI dashboards provide a mechanism for the line of business owners and executives to explore key performance metrics (KPIs) via visual interfaces. These dashboards are usually created by technical people. In fact, there are many technical people involved in the pipeline from the data to the dashboards, including the database designers, DBAs, business analysts, etc. Figure 1 shows a typical architecture of a BI stack. The underlying data resides in a traditional RDBMS, and a business model is created in terms of an OLAP cube definition [13] that describes the underlying data in terms of *Measures* (numeric or quantifiable values), *Dimensions* (categorical or qualifying attributes), and the hierarchies and relationships between them. Then, business analysts create the BI reports and dashboards using the BI model (cube definition) [1]. The reports and the dashboards are supported by structured queries that run against the underlying database to render the visualizations to the user.

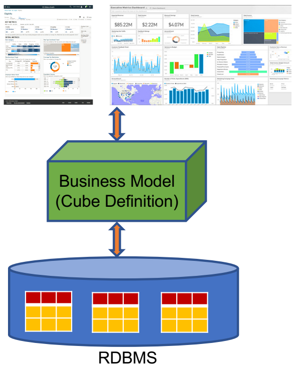


Figure 1: Traditional BI System Architecture

To obtain answers to questions that are not contained in the existing dashboard visualizations, users need to enlist the help of technical people, and the turnaround time for such cycles can be prohibitively time-consuming and expensive, delaying key business insights and decisions. Today's enterprises need faster access to their KPIs and faster decision making.

Conversational interfaces enable a wide range of personas including non-technical line of business owners and executives to explore their data, investigate various KPIs, and derive valuable business insights without relying on external technical expertise to create a dashboard for them. As such, conversational interfaces democratise access to data significantly, and also allow dynamic and more intuitive explorations of data and derivation of valuable business insights.

Today's chatbot and voice assistant platforms (e.g., Google Dialogflow, Facebook Wit.ai, Microsoft Bot Framework, IBM Watson Assistant, etc.) allow users to interact through natural language using speech or text. Using these platforms, developers can create many kinds of natural language in-

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/4.0/. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 13, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3415478.3415557

---

[1] In this paper, we use the terms cube definition and business model interchangeably

terfaces (e.g., chatbot, natural language search, etc.) for any kind of domain (e.g., weather, music, finance, travel, healthcare, etc.). These custom or domain-specific natural language assistants usually target a range of domain specific tasks, such as booking a flight, or finding a drug dosage. Such task-oriented agents limit the scope of the interaction to accomplishing the task at hand and hence are more tractable to design and build. However, these task-oriented agents fail to address the challenges involved in iterative data exploration through conversational interfaces to gain information and derive meaningful insights.

Recently, several business intelligence tools, such as Ask Data Tableau [2], Power BI [8] by Microsoft, Microstrategy [6], and the IBM's Cognos Assistant [3], also explored exploiting natural language interfaces. These early systems have many restrictions in terms of the conversational interaction they provide, as they rely on the user to specify several parameters, and only offer a fixed set of patterns.

There are several challenges in creating a conversational interface for a BI application. The first challenge is creating a data model that captures the entities, and their relationships and associated semantics that are relevant to the underlying data and the common set of BI queries and operations. We have two options: Modeling the underlying data in the RDBMS, or modeling the cube definition. We chose the latter, because a cube definition provides important BI specific information, such as measures, dimensions, dimension hierarchies, and how they are related.

The second challenge is building the necessary capability of the conversation system to capture user intent, recognize and interpret the different workload access patterns. We explore three different approaches, which we explain in detail in Section 3.3. The first two approaches use only the information available in the ontology, capturing the structural relationships between measures and dimensions. The third approach also takes into account user's access patterns.

The third and the final challenge is the integration with the underlying BI platform to issue appropriate structured queries and render the intended visualizations.

In this paper, we explore the use of conversational interfaces for BI applications. In earlier work [22], we developed an ontology-based approach to developing conversational services to explore the underlying structured data sets. In particular, we developed techniques to bootstrap the conversation work space in terms on intents, entities, and training samples, by exploiting the semantic information in an ontology. In this paper, we extend that work for BI applications. In particular, we observe that users follow certain *B*I patterns and operations when analyzing their data using BI tools. We exploit this information in the construction of the conversation work space, as well as the conversation design. We have implemented our techniques in *Health Insights (HI)*, an IBM Watson Healthcare offering, providing analysis over insurance data on claims, and our initial feedback from users has been very positive.

We demonstrate the effective exploitation of the BI access patterns to provide a more dynamic and intuitive conversational interaction to derive business insights from the underlying data, without being tied to a fixed set of pre-existing dashboards and visualizations. We evaluate our approach and show that our conversational approach to BI not only covers the use cases supported by pre-defined dashboards, but goes way beyond to assist users in better understanding the insights from existing visualizations as well as discovering new and useful insights that are not covered by the pre-defined dashboards through the dynamic generation of structured queries and integration with the underlying BI platform.

The main contributions of this paper can be summarized as:

- We propose an end-to-end ontology-based framework, and tools to create a conversation service for BI applications.
- We create an ontology from a business model, capturing all the key information for the BI application, including measures, dimensions, dimension hierarchies, and their relationships.
- We exploit common BI access patterns and use the ontology to generate several conversation space artifacts automatically, including intents, entities, and training examples.
- We adapt the dialog structure to support the BI Access patterns and operations to provide an intuitive conversational interaction for BI applications.
- We implement and demonstrate the effectiveness of our proposed techniques for *Health Insights*, an IBM Watson Healthcare offering.

The rest of the paper is organized as follows. Section 2 provides a brief overview of our ontology-driven approach for building conversational interfaces for BI applications. Section 3 describes in detail our approach for data modeling and generation of conversational artifacts including intents, entities and dialog. We discuss the implementation of our proposed techniques in a healthcare use case *Health Insights* in Section 4 and provide a detailed system evaluation in Section 5. We discuss related work in Section 6 and conclude in Section 7.

2. SYSTEM OVERVIEW

In this section we provide a brief overview of our ontology driven approach to building a conversational BI system for deriving useful insights from data in different domains.

2.1 Ontology-driven approach

In our prior work [22], we demonstrate the viability of using an ontology-based approach for building conversational systems for exploring knowledge bases. Ontologies provide a powerful abstraction of representing domain knowledge in terms of relevant entities, data properties and relationships between the entities which is much closer to and intuitive for natural language interaction. We have shown the effectiveness of capturing patterns in the expected workload and mapping them against the domain knowledge represented using an ontology to generate artifacts for building a conversational system in [22].

In this paper we build further on this effective approach to create a *Conversational BI* system for supporting natural language interfaces for BI applications, where the workload is characterized by a rich set of access patterns against an OLAP [13] business model defined over the underlying data. Figure 2 outlines our ontology-driven approach to building

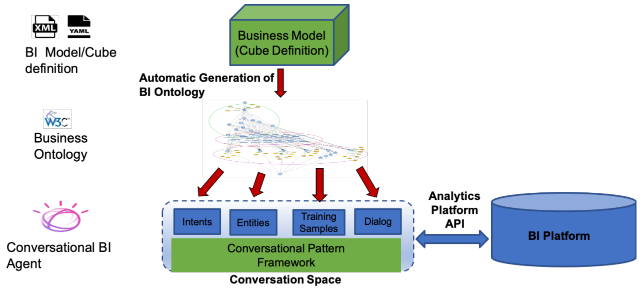


Figure 2: An Ontology Driven Approach for Conversational BI systems.

a natural conversation interface (NCI) [2] for supporting BI applications.

We create an ontology from the business model defined over the raw data in the form of an ontology which provides rich semantics, reasoning capabilities and an entity-centric view of the business model which is closer to natural language conversation. In addition to this, the ontology provides the necessary formalism to capture and represent the structure and content of the information defined in the business model using a well accepted industry standard [1]. The ontology represents a central repository for capturing the domain schema and any changes to it over time, thus making the design of our system more dynamic and enabling adaptability to different domains with additional input from subject matter experts (SMEs) (Ref Section 3.4.2).

More specifically, the ontology captures the measures and dimensions defined in the business model as entities, their taxonomy or hierarchies as described in the cube definition, in terms of parent-child relationships. Measures correspond to quantifiable elements computed over one or more elements in the physical schema such as columns in a relational schema and dimensions represent categorical or qualifying attributes. The ontology captures the individual relationships between the measures, dimensions and dimension groups, the attributes describing individual measures, dimensions as data properties. In addition to this, we also define special concepts in the ontology called *Meta Concepts*. These meta concepts represent a higher level grouping of measures/dimensions provided by SMEs or learnt from the underlying data through machine learning or deep learning techniques which we refer to as ontology enrichment. Meta concepts provide a powerful abstraction for reasoning at a semantically higher level and enable the conversation system to support the querying needs of a much wider range of personas (Section 3.4.1).

We map the common BI access patterns against the ontology and use it to drive the process of building the conversation system that allows users to interact with the underlying data using a NCI. We use IBM's cloud based Watson Assistant(WA) service to build the conversation system.

2.1.1 Automated Workflow

The automated workflow represented in Figure 2 describes the process of automatically generating the necessary artifacts for building a domain specific conversational BI system in a domain agnostic way.

The automated workflow accelerates the process of building a conversational BI system and is key to enabling rapid prototyping and system development against data in different domains. The workflow has three distinct steps. The first step involves the generation of the ontology from the business model. In the second step, the information captured/modeled in the ontology is used to drive the generation of the required artifacts/components of the *conversation space*. The conversation space consists of three main components that enable it to interact with users: intents, entities, and dialogue. Intents are goals/actions that are expressed in the user utterances, while entities represent real world objects relevant in the context of the user utterance. Typically conversational systems use a classifier or a deep neural network to identify the intent in a user utterance [15] and hence require training examples in terms of sample user utterances for each intent. The dialogue provides a response to a user conditioned on the identified intents, entities in the user's input and the current context of the conversation. The final step is the integration of the conversation space with an external data source or analytics platform that stores and processes the data. This integration is achieved through structured query generation (Section 3.7) against the analytics platform to enables the conversational system to respond to user utterances with insights in the form of charts/visualizations. As can be seen, the automated workflow utilizes the domain specific aspects including the domain ontology and the domain vocabulary (entities) and enables the creation of a domain specific conversational system while making the process itself repeatable across different domains. A detailed description of each of these steps is provided in Section 3.

3. CONVERSATIONAL BI SYSTEM

3.1 Data Modeling

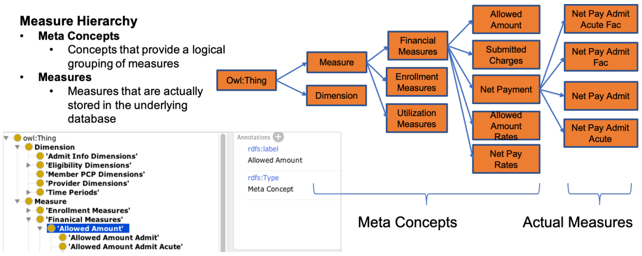
The OLAP [13] business models describe the underlying data in terms of measures, dimensions, their relationships and hierarchies. We have developed an ontology generation module that creates an OWL ontology [1] given an OLAP business module as input. For each measure and dimension specified in the business module, the ontology generator creates an OWL *concept/class* in the ontology. Further, for each measure and dimension that are connected in the business module, it creates an OWL *functional object property* with the measure as the domain and the dimension as the range of the object property. The attributes associated with the measures and dimensions in the business model are added as OWL *data properties* for the respective measure and dimension concepts in the ontology. Finally, all the dimensional hierarchies in the business model are captured as *isA* relationships between the dimensions created in the ontology.

The ontology is further enriched to define *meta-concepts* as a hierarchy of logical groupings of existing measures and dimensions extracted from the business model with the help of SMEs. These hierarchical grouping of measures and dimensions called *meta-concepts* are annotated as such in the ontology with appropriate labels marking ontology concepts as actual measures, dimensions and meta-concepts. The figures below show an example measure hierarchy (Ref Figure 3) and dimension hierarchy (Ref Figure 4) captured

---

[2] We use the terms natural conversation interface, conversational interface, conversational system interchangeably in the rest of the paper.

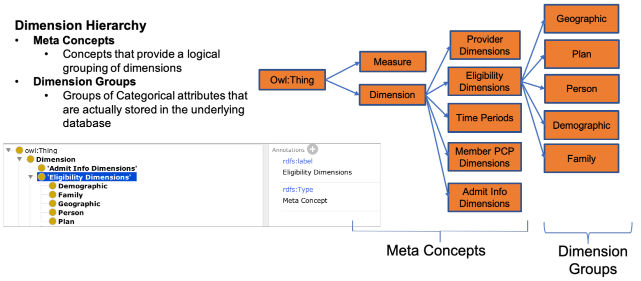
in the enriched ontology along with annotations for meta-concepts. As we describe in the next section, this entity-centric modelling of the business model is key to represent and reason about the common BI workload access patterns and operations (*BI Access Patterns* [3]) as subgraphs over the ontology and generate the necessary artifacts for the conversation system.



Measure Hierarchy

- **Meta Concepts**
  - Concepts that provide a logical grouping of measures
- **Measures**
  - Measures that are actually stored in the underlying database

Figure 3: Captured Measure Hierarchy.



Dimension Hierarchy

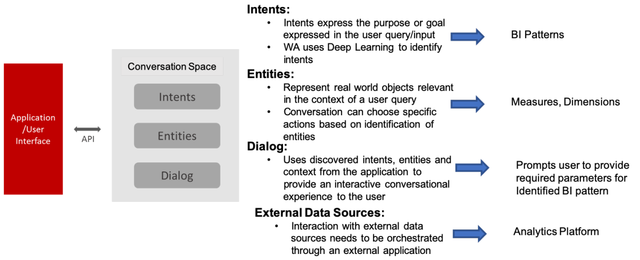
- **Meta Concepts**
  - Concepts that provide a logical grouping of dimensions
- **Dimension Groups**
  - Groups of Categorical attributes that are actually stored in the underlying database

Figure 4: Captured Dimension Hierarchy.

3.2 Ontology-driven generation of conversational artifacts

The second step in the automated workflow (Figure 2) consists of generation of conversational artifacts from the information captured in the ontology. The central tenant of the artifact generation process revolves around supporting the BI access patterns to gain business insights using a conversational interface. Figure 5 describes the artifacts required for constructing a conversation space in terms of intents, entities, dialog and how we map them to the specific elements relevant to BI. More specifically, we map Intents to BI patterns. Entities are mapped to the measures and dimensions defined in the business model and captured in the ontology. The dialog is especially designed to support interaction with the user based on the BI pattern/intent and entities detected in the user utterances and the current context of user conversation. Integration with an external data source such as an analytics platform is required to support *actions* such as responding to user requests with appropriate results including charts/visualizations.

Next we describe in detail the modeling and generation of intents, their training examples (Section 3.3) and entities (Section 3.4.1) for the conversation space. Construction of dialog is described in detail in Section 3.6. Integration with an external data source requires structured query generation which we describe in Section 3.7.



Intents:

- Intents express the purpose or goal expressed in the user query/input
- WA uses Deep Learning to identify intents

Entities:

- Represent real world objects relevant in the context of a user query
- Conversation can choose specific actions based on identification of entities

Dialog:

- Uses discovered intents, entities and context from the application to provide an interactive conversational experience to the user

External Data Sources:

- Interaction with external data sources needs to be orchestrated through an external application

Application /User Interface
API
Conversation Space
Intents
Entities
Dialog
BI Patterns
Measures, Dimensions
Prompts user to provide required parameters for Identified BI pattern
Analytics Platform

Figure 5: Conversation Space and artifacts required.

3.3 Intent Modeling for BI

As described in Figure 5, Intents capture the purpose or goal in the user query/input. While designing the conversational BI system, we considered three different approaches for modeling intents. The first two approaches approaches are based on the structural relationships between the measures and dimensions in the ontology. The third approach combines the user access patterns extracted from prior or expected workloads with the structural information in the ontology. We describe each of these approaches below and provide a brief evaluation to ascertain their effectiveness.

3.3.1 Modeling intents as combinations of Measures and Dimensions

In this approach we traverse the ontology and capture valid combinations of individual measures and dimensions as intents. For each identified measure in the ontology, the algorithm traverses each edge that connects the measure to a dimension. Each such identified pair that is connected via an edge in the ontology is identified as a *valid combination*. This is the finest granularity of generating intents for the conversation system which captures the user's goal/purpose of obtaining information about a particular measure with respect to a particular dimension.

The problem associated with this approach is both in terms of scalability and accuracy. Modeling intents at such fine granularity leads to a combinatorial explosion of the number of intents and their corresponding training examples. Further, as several intents may contain overlapping sets of measures and entities the classification accuracy in terms of F1-score drops leading to poor user experience.

3.3.2 Modeling Measures as intents

This approach models each individual measure as a separate intent. Such an approach allows us to capture the user's intent in terms of obtaining information about a particular measure, irrespective of the dimension(s) it needs to be sliced by. In order to generate training examples for each intent, we traverse the ontology to determine the valid combinations of measures and dimensions and use that to create training examples for each intent.

This approach reduces the combinatorial explosion of the number of intents as compared to the previous approach discussed above. However as the number of measures captured in the ontology from the underlying business model grow larger the number of intents and their associated training examples may still be quite large. This again may lead to significant scalability problems. Another issue with this approach is that there might be considerable overlap between the training examples of certain intents leading to low accuracy. This is due to the fact that different measures may

---

[3] We use BI Access Patterns and BI patterns interchangeably in the rest of the paper.

be related to the same dimensions. For e.g. #Admits and #Discharges can both be related to dimensions such as year or facility.

3.3.3 *Modeling BI patterns as intents*

In this approach we identify the common BI workload access patterns from prior user experience and BI application logs. Each such identified pattern is modeled as an intent. We develop ontology traversal algorithms that map these identified patterns to subgraphs over the ontology. For each such subgraph, we identify the measures, dimensions and their associated instance data crawled from the underlying data store to generate training examples for each intent.

Modeling intents as BI patterns has the critical advantage of combining user access patterns with the domain knowledge in terms of the structural relationships between the measures and dimensions in the ontology. Combining this information allows us to better model the intents conversational BI applications. In our experience this approach provides the maximum coverage (Recall) of user queries without having to deal with a combinatorial explosion in terms of the number of intents. This makes this approach most scalable. Each intent is very well defined and has sufficient distinction in terms of associated training examples thereby giving the highest accuracy in terms of F-1 scores amongst all the approaches discussed above. We describe each of these patterns in detail with examples in Section 3.3.4.

Table 1 shows a summary of comparison of the different approaches for modeling intents for the HI dataset (Ref Section 4) containing 64 Measures, 274 dimensions, 576 relationships. The comparison is done to assess the scalability of each approach in terms of number of intents and training examples that would be required to cover combinations of user utterances involving on average one measure and two dimensions. A detailed analysis for accuracy is provided in Section 5.

Table 1: Comparison of intent modeling approaches

Modeling approach	# Intents	# Training E.g.s
Measure, Dimension combination as intents	576^3	$(576^3) * 10$
Measures as intents	12	$12 * 576^2$
BI patterns as intents	7	$7 * 64 * 274$

3.3.4 *BI Conversation Patterns*

In this section we describe the commonly identified BI access patterns learnt from prior BI workloads and application logs. Each of these patterns is modeled as an intent in the conversation space and requires the generation of training examples for the same. A classifier in the conversation space is trained using these examples to classify user utterances into one of the BI patterns. Once the BI conversation pattern is identified, the conversational system extracts the relevant entities mentioned in the user utterance in terms of measures, dimensions, filter values. The dialog structure uses these extracted intents and entities and the current conversational context and provides appropriate responses. We describe below the common BI access patterns that we have used in our Health Insights use case (Section 4).

- **BI Analysis pattern.** This pattern is the most common BI pattern that allows users to see a measure(s) sliced along a particular dimension(s) and optionally applying a filter(s). Figure 6 shows the pattern along with an example.

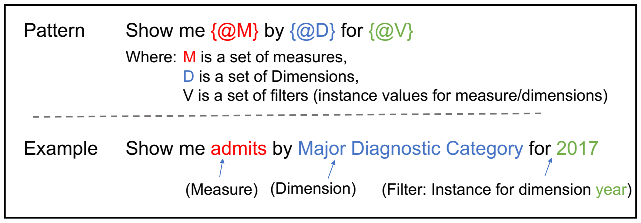

Pattern
Show me {@M} by {@D} for {@V}
Where: M is a set of measures,
D is a set of Dimensions,
V is a set of filters (instance values for measure/dimensions)
Example
Show me admits by Major Diagnostic Category for 2017
(Measure)
(Dimension)
(Filter: Instance for dimension year)

Figure 6: BI Analysis query pattern (Best viewed in color).

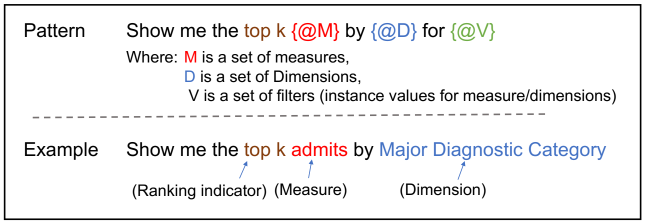
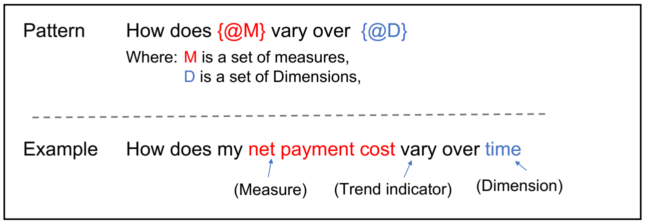

Pattern
Show me the top k {@M} by {@D} for {@V}
Where: M is a set of measures,
D is a set of Dimensions,
V is a set of filters (instance values for measure/dimensions)
Example
Show me the top k admits by Major Diagnostic Category
(Ranking indicator)
(Measure)
(Dimension)

Figure 7: BI Ranking pattern (Best viewed in color).

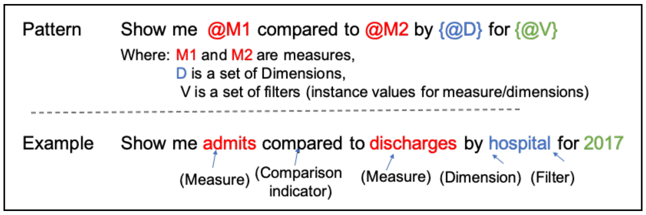
- **BI Operation patterns.** The BI operation patterns capture some common BI operations which usually follow other BI queries such as a BI Analysis query for further analysis on the results obtained. We describe these operations below and provide example user interactions associated with these operations in a use case that we have implemented, described in Section 4.
  - **Drill down operation pattern:** Access more granular information by adding dimensions to the current query.
  - **Roll up operation pattern:** Access higher level information by aggregating along the dimension hierarchy with respect to the current query.
  - **Pivot operation pattern:** Access different information by replacing dimensions in the current query.
- **BI Ranking Pattern.** The BI ranking pattern allows users to order the results by a measure value (or an aggregation applied on a measure) generally to obtain the top k values. Figure 7 shows an example of a ranking BI pattern. The results are sorted by #Admits shown along the dimension MDC (Major Diagnostic Category).
- **BI Trend pattern.** This pattern captures the variation of a measure along dimensions such as time or geography to ascertain the trend associated with the measure of interest. Figure 8 shows an example of the BI trend pattern. The results for the example query display the variation of the measure *Net Payment* by Incurred Year or Paid Year both of which are time dimensions as inferred from the ontology.

  The pattern looks similar to the BI Analysis pattern, however we chose to model it as a separate pattern as the linguistic variability of queries requesting for trends vs standard BI Analysis queries was sufficient enough to warrant a separate intent. For example the


Pattern How does {@M} vary over {@D}
Where: M is a set of measures,
D is a set of Dimensions,

Example How does my net payment cost vary over time

Figure 8: BI Trend pattern (Best viewed in color).


Pattern Show me @M1 compared to @M2 by {@D} for {@V}
Where: M1 and M2 are measures,
D is a set of Dimensions,
V is a set of filters (instance values for measure/dimensions)

Example Show me admits compared to discharges by hospital for 2017

Figure 9: BI Comparison pattern (Best viewed in color).

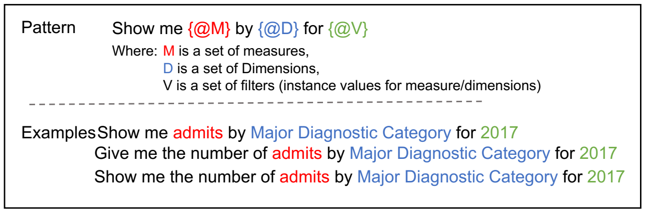
same query could also be expressed as *Show me the trends in my net payment.* This query is identified as a BI trend pattern and a default dimension of time (paid or incurred year) is chosen to show the variation of the measure *net payment.* We further describe the choice of default inferences for measures and dimensions in section 3.4.1.

- **BI Comparison pattern.** Another common BI pattern observed is the BI comparison pattern which allows users to compare two or more measures against each other along a particular dimension(s) and optionally applying filter value(s). Figure 9 shows an example BI pattern that compares the number of admits to discharges by hospital (dimension) for the year 2017 (a filter value).

3.3.5 Generation of Intent training examples

We follow a similar process as discussed in [22] for the automatic generation of training examples for the identified intents. The above-mentioned BI conversation patterns are mapped over the ontology as subgraphs and used as templates for generating training samples by plugging in the measure, dimension and filter values as discerned from the domain ontology and the instance values that map to different elements in the ontology.

More specifically, for each BI pattern modeled as an intent, the corresponding template (examples of which are shown above) is populated with the appropriate measure, dimension and filter values using an algorithm that traverses the ontology and discovers appropriate relationships between measures, dimension groups and their hierarchies and populates the templates accordingly. These generated training examples are used to train the intent classifier model in the conversation space. Figure 10 shows a sample of training examples generated for the BI Analysis Query pattern. The *initial phrases* for each intent, such as *Show me, Give me the number of, etc.* are provided as an input to the algorithm which picks these at random to generate the training examples. The automatically generated training examples are also further augmented with more examples with the help of SMEs and from queries seen in prior workloads/user experiences if available.


Pattern Show me {@M} by {@D} for {@V}
Where: M is a set of measures,
D is a set of Dimensions,
V is a set of filters (instance values for measure/dimensions)

Examples Show me admits by Major Diagnostic Category for 2017
Give me the number of admits by Major Diagnostic Category for 2017
Show me the number of admits by Major Diagnostic Category for 2017

Figure 10: Generation of Intent Training Examples (Best viewed in color).

3.4 Entity Modeling for BI

This section describes in detail how we model entities relevant to the access patterns and the underlying business model. We first discuss how measures, dimensions and their hierarchies are captured and populated as entities. Next, we describe the addition of domain specific vocabulary and synonyms to the conversation space to provide greater flexibility and improve the recall of user utterances. Finally we talk about the use of default inferences and their relevance in the system design for providing a better user experience.

3.4.1 Modeling of Measures, Dimensions, their hierarchies and relationships

Concepts in the ontology generated from the business model are annotated as the following:

- *Measures and dimensions.* These entities are part of the cube definition and are mapped to appropriate columns in the underlying relational schema. The BI Queries involving these measures and dimensions in the ontology are mapped to appropriate structured queries against an external data source to provide the required response.

- *Meta concepts.* These are part of a hierarchy which represents logical groupings of the underlying measures or dimensions and are not mapped directly to any elements in the underlying relational schema (Ref Figures 3, 4). These meta concepts might be defined and extracted from the business model if available, or are additional metadata information provided by the SMEs and included in the ontology as a post processing or enrichment step.

  Figure 11 shows example queries that demonstrate the effectiveness of modeling meta concepts in the ontology. These queries conform to the BI Analysis Query pattern and refer to *costs* as a measure that is a meta concept. On detecting a standard BI analysis query with a meta concept 'costs' as an entity, the conversation space utilizes the mappings from the ontology between costs and the actual measures such as #Admits, Net Payments, etc. and provides users with the options to choose from the actual set of measures associated with costs or provides results for all the measures associated with cost depending on user preferences in the domain.

Additionally, inference of the meta concept *Costs* is also driven by the current context of user conversation. For example, if the user has been talking about *Admissions* in his prior user utterances, measures associated with admissions would be captured in the current conversational context. Based on this *costs* may be mapped to the measure *Allowed Amount Admit.*

Clearly, we see that the mechanism we built around the creation and utilization of meta-concept groupings or mappings in our conversational system design provides a powerful mechanism to support more complex and higher level queries (Figure 11). This helps increase the applicability of our system for a wide variety of personas that are interested in gaining business insights at different levels from the underlying data.

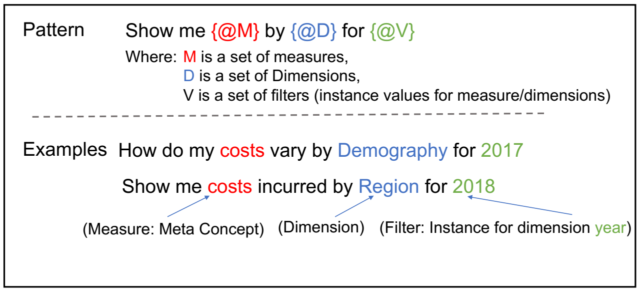

Pattern
Show me {@M} by {@D} for {@V}
Where: M is a set of measures,
D is a set of Dimensions,
V is a set of filters (instance values for measure/dimensions)
Examples
How do my costs vary by Demography for 2017
Show me costs incurred by Region for 2018
(Measure: Meta Concept)
(Dimension)
(Filter: Instance for dimension year)

Figure 11: Example Queries referring to a measure Meta-Concept.

3.4.2 *Domain specific vocabulary and synonyms*

Domain specific vocabulary and synonyms allow users to express queries using terminology that is common to the domain and does not restrict users to use query terms that are specific to either the terminology/vocabulary used in the ontology or instances of data corresponding to the ontology. Our system incorporates domain specific vocabulary and synonyms collected from SMEs/domain experts including standard taxonomies such as SNOMED in the medical domain as well as taxonomies developed by SMEs such as those related to diagnosis, therapeutic drug classes, etc. These dictionaries/taxonomies help map the synonyms and other vocabulary terms to entities in the ontology and help in increasing the recall of entities that can be inferred by the conversational system from user utterance thereby allowing users a flexible mechanism to support a variety of queries against the underlying data.

3.5 Defaults and learning from experience

3.5.1 *Default inferences*

An important aspect of conversational system design especially relevant to user experience is the use of default inferences. These are used for inferring missing parameters in a query that the users assume the system would infer automatically given the context of the conversation. These often include inferring default measures for a particular dimension and vice-versa in a conversational thread with a user. For e.g. *Show me the top-K DRGs for pregnancy* requires to show the #Admits or Allowed Amount as inferred measures (not explicitly mentioned in the user utterance) for the dimension DRG (Diagnosis Related Group) and sort the results by the measure value. These default inferences are made by integrating dictionaries containing this information obtained from SMEs into the conversational workspace.

Using default inferences helps improve user experience by avoiding asking too many follow-up questions and can be dynamically adjusted as users either accept the default inference or provide feedback that enables us to update/modify the default inferences used by the system.

3.5.2 *Learning from feedback*

As a conversational system is tested with real users, alternative phrasings of known intents and synonyms of known entities will emerge. As these are discovered through testing, they are added to intent training examples or entity synonym lists so that the system learns over time. Utterances that were not recognized by the system are obtained from the application logs [4], and alternative phrasings or synonyms are identified to be added to the system. This addition of new data can be automated through particular conversation patterns that enable the system and user to identify a new bit of data and then add it to the training corpus without the intervention of a developer or designer.

3.6 Building the Dialog

Natural language interaction platforms, such as IBM's Watson Assistant, enable many different styles of interaction. At their core, they consist of intents and entities, for understanding users' natural language inputs, and a dialog manager, with context variables, for deciding how to respond. In this section, we briefly define the particular interaction model we used and then detail how we built it and adapted it for BI.

3.6.1 *Query Model*

The simplest interaction model for a natural language-based system is perhaps the *query model*. Under this model, users submit queries and the system responds with answers, much like a search engine.

Example
01 U: Show me admits by DRG for 2017
02 A: Here are Admits by Diagnosis Related Group for 2017:
03 ((chart appears))

Example
01 U: Show me admits
02 A: I'm afraid that is an invalid query.

While this simple query model can be powerful in enabling access to domain information, it is not *conversational*. The system only produces one of two possible responses: Answer or No Answer. In addition, each 2-utterance sequence is independent. If the user utters a second query, it will be interpreted without any context from the previous query or answer. Finally, the system does not recognize conversational utterances, such as displays of appreciation or requests for repeats.

---

[4] All application logs used for learning are anonymized and are devoid of any personal information for maintaining data privacy.

3.6.2 Natural Conversation Model

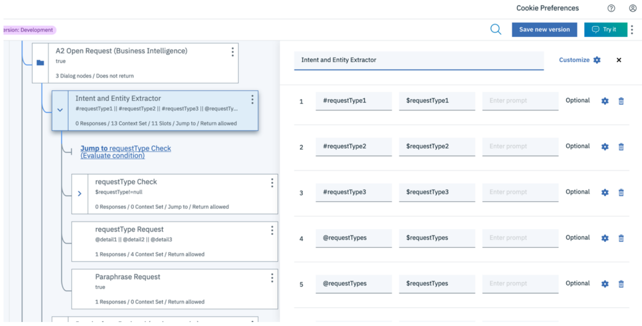
One alternative to a query-oriented interaction model is a natural conversation model. Although the term ”conversation” is used for many different kinds of interaction, we define a *natural conversation interface* or *natural conversation agent* as one that exhibits the ability for natural language interaction (understanding and responding in natural language), persisting context across turns of conversation and conversation management [21].

We created our natural conversation interface by using the Natural Conversation Framework (NCF) [21]. The NCF provides a pattern language of over 100 reusable interaction patterns, which we have implemented on the Watson Assistant platform. We reused primarily the NCF's *Open Request* for enabling series of complex requests, in addition to some of its *conversation management* modules. The Open Request module enables standard ”slot-filling,” or agent-initiated detail elicitation, but it also includes multiple features for making the interaction with the user more conversational. It allows users great flexibility in the ways they express their requests, and it remembers the context across utterances so the system does not forget what it is talking about based on prior user utterances. We provide a sample interaction for such an interaction pattern in Section 4.

3.6.3 Natural Conversation for BI

In order to adapt our natural conversation interface to the use case of business intelligence (BI), we created a dialog logic table [22] (Ref Section 4(Table 2)). The table specifies the relationships among each of the particular intents, entities and responses. For example, it specifies which parameters are required for each intent, or request type, and which are optional, as well as specifying the natural language framing for both the users' and agent's utterances. From such a table, it is easy to build a corresponding dialog tree that encodes the interaction patterns.

Intent and Entity Extractor. The core of the NCF's Open Request module [21] is the intent and entity extractor, which allows for a more natural and conversational interaction. Every user utterance is funneled through the extractor so that no use-case-specific intent or entity is missed. This enables users to produce their query incrementally, across multiple utterances, instead of requiring them to produce it in a single utterance or to repeat the same entities for a new intent (as in slot-filling). Figure 12 shows an example of our intent entity extractor, which captures each request type, such as analysis query or trend query, and detail, such as admits or incurred year (not shown), to a context variable.



A2 Open Request (Business Intelligence)
true
3 Dialog nodes / Does not return

Intent and Entity Extractor
#requestType1 || #requestType2 || #requestType3 || @requestTy...
0 Responses / 13 Context Set / 11 Slots / Jump to / Return allowed

Jump to requestType Check
(Evaluate condition)

requestType Check
$requestType!=null
0 Responses / 0 Context Set / Jump to / Return allowed

requestType Request
@detail1 || @detail2 || @detail3
1 Responses / 4 Context Set / Return allowed

Paraphrase Request
true
1 Responses / 0 Context Set / Return allowed

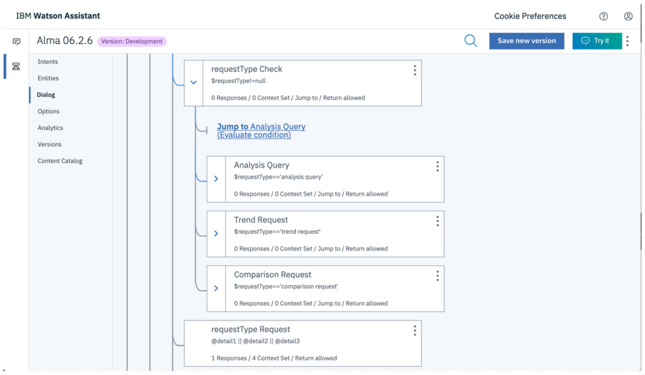
Intent and Entity Extractor

1	#requestType1	$requestType1	Enter prompt	Optional
2	#requestType2	$requestType2	Enter prompt	Optional
3	#requestType3	$requestType3	Enter prompt	Optional
4	@requestTypes	$requestTypes	Enter prompt	Optional
5	@requestTypes	$requestTypes	Enter prompt	Optional

Figure 12: Intents and Entity Extractor.

Dialog Structure for handling BI query patterns. Figure 13 shows an example dialog tree structure in which each BI Query pattern, modelled as an intent, is assigned a separate dialog node(s) to trigger an appropriate response to the user or to elicit further information if required. Modeling the dialog structure in such a manner allows the conversation system to respond to each BI query pattern uniquely, as well as to assist in appropriate structured query generation (Section 3.7).

Query Completeness and Detail Elicitors. We incorporate a query completeness check mechanism using a special node *Complete Request* in the dialog tree to verify the completeness of each BI Query pattern (intent) identified in the user utterance. The completeness check is a two step process. First the system checks whether the user utterance has all the *required entities* for the identified intent as per the dialog logic table. If not, the system checks the current conversational context to see if the required entities have already been provided by the user in a prior user utterance. If yes, then the query is marked complete. If not, we use the detail elicitors (or sometimes called *slots*), mechanism to elicit further information from the user which he or she might have failed to provide in the initial user utterance, through oversight or lack of knowledge. When the query is marked as complete the system provides an appropriate response which might involve the use of structured query generation to obtain results or visualizations from an external data source.



requestType Check
$requestType!=null
0 Responses / 0 Context Set / Jump to / Return allowed

Jump to Analysis Query
(Evaluate condition)

Analysis Query
$requestType=='analysis query'
0 Responses / 0 Context Set / Jump to / Return allowed

Trend Request
$requestType=='trend request'
0 Responses / 0 Context Set / Jump to / Return allowed

Comparison Request
$requestType=='comparison request'
0 Responses / 0 Context Set / Jump to / Return allowed

requestType Request
@detail1 || @detail2 || @detail3
1 Responses / 4 Context Set / Return allowed

Figure 13: Dialog Structure for handling BI query patterns

Query Validation. Query validation is an additional step we introduce to verify the semantic correctness of the user query that conforms to a particular BI pattern/operation and is marked as complete by the query completeness check mechanism described above. The validation of the query is done using information captured in the ontology. For e.g. A user utterance/query might conform to the BI Query Analysis pattern (Figure 6) and contain the required entities such as a measure, a dimension and a filter value as per the dialog logic table 2. The validation process traverses the ontology to verify if there is a valid relationship(s) between the measure, dimension and filter values specified in the BI Analysis Query by the user. If so, a structured query is generated against an external data source to respond to the user. If not, the user is informed of the incorrectness observed and asked to modify the query.

Support for BI Operation Patterns. As mentioned in Section 3.3.4, BI operation patterns capture typical BI operations that allow users to further investigate the results obtained from other BI patterns such as a BI Analysis pattern. BI operations are supported using an incremental (of follow-up) request mechanism. The initial set of measures, dimensions and filter values specified in say a BI Analysis Query are captured in the current conversational context and all further BI Operations are executed on this set. BI operations such as *Drill Down*, *Roll Up* along a dimension hierarchy or *Pivot* are supported incrementally by changing the appropriate values in the conversational context.

3.7 Structured Query Generation

In this section we briefly describe our mechanism for structured query generation against APIs exposed by an external data source (or analytical platform) such as Cognos [5] [4], to provide appropriate responses to user queries including charts and visualizations.

We use a simple template based mechanism for structured query generation. For the Cognos analytics platform a *widget* acts as a template which is populated using the information in the conversation context to form the actual structured query. Each BI Query pattern (or intent) is mapped to a specific widget template. Although in our current implementation we use Cognos, our techniques and design are not specific to any particular external data source or analytics platform. The template-based query generation mechanism is flexible and can be used to support any back-end analytics platform.

The widget templates allow the specification of the information required in terms of measures, dimensions, aggregations, filters, etc. as gathered from the conversational context. The choice of the actual format of the response or visualization (such as a bar chart, scatter plot, line chart, etc.) appropriate for the requested information is deferred to the analytics platform which uses other internal recommendation tools and libraries to make the appropriate choice.

Other more sophisticated deep learning based techniques such as Seq2Seq networks [24] could be employed in general for structured query generation conditioned on the availability of enough training data for the appropriate analytics platform. We however observe that since the workload for BI applications is mostly characterized by the BI Query patterns, a template based mechanism as described above is sufficient to address the requirements of structured query generation for the majority of practically observed workloads. We leave the detailed exploration of other deep learning based techniques and their effectiveness for supporting structured query generation for BI applications as future work.

4. USE CASE: HEALTH INSIGHTS

In this section we describe the building of a Conversational BI application using our ontology driven approach for Health Insights, an IBM Watson Healthcare offering [5].

4.1 Health Insights Overview

The Health Insights (HI) product, an IBM Watson Healthcare offering which includes five different curated datasets of healthcare insurance data related to claims and transactions from a population covered by an insurance's healthcare plans. The integrated data across five different datasets includes basic information about participants'drug prescriptions and admissions, service, key performance factors such as service categories, data on individual patient episodes, which is a collection of claims that are part of the same incident to treat a patient. Finally, HI also includes the IBM MarketScan dataset [7] contributed by large employers, managed care organizations, hospitals. The dataset contains anonymized patient data including medical, drug and dental history, productivity including workplace absence, laboratory results, health risk assessments (HRAs), hospital discharges and electronic medical records (EMRs).

HI Business Module and Ontology generation. The HI data across several different data stores is attached to the Cognos analytics platform using Rest APIs. A business model was defined over this data that models the information in the underlying dataset in the form of measures, dimensions, their relationships and hierarchies. The business model defined a total of 64 Measures, 274 dimensions and 576 distinct relationships between the different measures and dimensions. The business model was further enhanced using SME domain knowledge to group the underlying measures and dimensions into logical groups to create a hierarchical structure. The hierarchical tree structure for the measures grouped the 64 leaf level measures into 12 measures at the second level and these 12 measures in turn were grouped into 3 top level measures. Similarly the 274 leaf level dimensions were grouped into 8 second level dimension groups and 5 top level dimension groups. Figures 3, 4 capture a snapshot of this grouping where each higher level grouping of a measure or dimension is referred to as a meta concept. We automatically generate on ontology from this business model using the mechanism described in Section 3.1 in an OWL format thus providing an entity-centric view of the business model.

HI Conversation artifact generation. We derived the conversational artifacts for HI from the generated ontology including a total of 7 intents one corresponding to each BI Query Pattern and about 20 intents to support conversation management. Automatically generated training examples (Section 3.3.5) for each of these intents were also included in the conversation space to train the intent classifier. Each identified measure, dimension and meta-concept was added as an entity. Instance values of the leaf level measures and dimensions crawled from the underlying data were also added as entities to the conversation space. SME knowledge was utilized to add synonyms for each of the populated entities for better recall and user experience.

HI Dialog Structure. Table 2 and Table 3 show versions of the dialog logic table that have been adapted specifically for the BI Query patterns. Table 2 illustrates an example of how three kinds of BI Query patterns, can be represented: BI Analysis Query pattern, BI Trend pattern and BI Comparison pattern(column 1). One example is given of each intent (column 2), although in practice, this would contain many variations for the same intent. A list of required entities that is shared across these intents is given (column 3), along with agent elicitations (column 4) for each required entity. Shared optional entities are also provided (column 5). Agent responses to each intent are provided (column 6).

---

[5] Cognos is a registered trademark of IBM.

Table 2: Dialogue Logic Table with BI Queries for HI.

Intent Name	Intent Example	Required Entities	Agent Elicitation	Optional Entities	Agent Response
BI Analysis Query	Show me people admitted in 2017	Measure(s), Dimension(s)	By which dimension? For which time period?	Filter value, Facilities	Here are the admits treat for 2017...
BI Trend Query	How does net pay cost vary over time?	Measures, Time	For which Measure? For which time period?	Geographies, facilities	Here is the net pay cost by incurred year...
BI Comparison Query	Show me admits vs discharges for 2017	Measures, dimensions	For which Measures? By which dimension	Filter value	Here are admits vs discharges by hospital...

Table 3: Dialogue Logic Table with BI Operations for HI.

Intent Name	Intent Example	Required Entities	Agent Elicitation	Optional Entities	Agent Response
Drill Down Request	Show me by MDC	Dimensions, ranks, time	What do you want to add?	NA	Here are the admits by Medical diagnostic Code for 2017...
Roll Up Request	Remove MDC	Dimensions ranks,time	What do you want to remove?	NA	Here are admits for 2017...
Pivot Request	How about by DRG?	Dimensions, ranks, time	What do you want to replace?	NA	Here are the admits by Diagnostic Related Group for 2017...

Meta-concepts are captured as optional entities (column 5), which are then used to trigger agent elicitations (column 4) for more specific, required entities (column 3).

Table 3 shows the specifications of additional dialog structures that go beyond the standard "slot-filling" structure, but still fall within the Open Request structure. It contains operations on a prior BI request and its associated context. For example, taking the Analysis Query from Table 2, the user can perform three BI operations (Table 3) on the agent response: Drill Down, Roll Up and Pivot. They enable the user to add, remove or replace dimensions in the current BI query, which is persisted in the conversational context from prior turns in the interaction. In this way, the user can articulate new BI queries incrementally.

Once all of the parts specified in the Dialog Logic Tables above (Tables 1 and 2), a dialog structure of BI conversation patterns can be created easily.

Example: BI Natural Conversation
01 User: Show me people admitted in 2017?
02 Agent: By which dimension?
03 User: DRG
04 Agent: Here is Admits by Diagnosis Related Group for 2017:
05 (chart #1 appears)
06 User: How about MDC?
07 Agent: Here is Admits by Medical Diagnostic Code for 2017:
08 ((chart #2 appears))
09 User: Show distribution by hospital
10 Agent: Here is Admits by Medical Diagnostic Code and by Facility for 2017:
11 ((chart #3 appears))
12 User: remove facility
13 Agent: Here is Admits by Medical Diagnostic Code for 2017:
14 ((chart #4 appears))
15 User: ok thanks
16 Agent: You're welcome!
17 User: Show me top 5 Admits
18 Agent: Here are the top 5 Admits by Medical Diagnostic Code for 2017:
19 ((chart #5 appears))
20 User: ok
21 User: How does my net pay cost vary over time?
22 Agent: Here is Net Payment Cost by Incurred Year:
23 ((chart #6 appears))
24 User: ok
25 User: Show me admits vs discharges by hospital for 2018?
26 Agent: Here admits vs discharges by Facility for 2018:
27 ((chart #7 appears))
28 User: ok thank you very much
29 Agent: You're welcome!
30 ((pause))
31 Agent: Anything else?

The above example illustrates a conversational interaction using a Natural Conversation Interface (NCI) that has been adapted for BI analysis to support the HI application. In this example, there is a BI Analysis Query sequence (lines 01-05) followed by three BI operations: pivot (lines 06-08), drill down (lines 09-11) and roll up (lines 12-14). There is then an example of a BI Trend Query sequence (lines 21-23) and a BI Comparison Query sequence (lines 25-27). We also see the use of synonyms (hospital, facility) here. Around these BI-related conversational sequences are examples of generic conversation management sequences (lines 15-16, 20, 24 and 28-31). Although this example opens with the Analysis Query pattern, which follows the standard "slot-filling" pattern in dialog design, it proceeds to demonstrate additional interaction patterns, incremental requests (or BI operations) and conversation management, which go beyond simple slot filling.

5. SYSTEM EVALUATION

Our conversational BI application, implemented in *Health Insights (HI)*, was hosted in the IBM cloud and utilizes several other cloud services including IBM Watson Assistant for building the conversational space. We first describe the evaluation of our ontology-driven generative approach for creating conversational artifacts. More specifically we evaluate the effectiveness of our proposed intent modeling techniques for BI applications in terms of their coverage and accuracy. Next, we describe a detailed user study that was conducted to ascertain the overall effectiveness of our proposed conversational BI system. Our prototype for the user study used an on-premise deployment of the Cognos analytics platform that was loaded with a subset of data from Health Insights. Finally, we summarize the section with some lessons learned from our experience of building the conversational BI application.

5.1 Intent Modeling evaluation

We evaluated our ontology-driven intent modelling approach based on (1) the coverage it provides for accessing a statically defined set of dashboards for HI and (2) the accuracy with which the system can identify the correct intents from the user utterances based on a user study (Section 5.2).

5.1.1 *Intent modeling coverage evaluation*

We evaluate the coverage of our ontology-based intent modeling approach in terms of the subset of statically defined dashboard visualizations for HI that can be accessed using our conversational BI interface. Having said that, we would like to note that our system is not limited to accessing the information from these statically defined visualizations. Our proposed conversational interface can support new queries and explorations that conform to one of the common BI patterns modelled as intents.

For the purposes of the evaluation, we defined a total of 150 different visualizations statically across 37 dashboards grouped under 4 different analysis themes with the help of SMEs. For each of these statically defined visualizations we characterized them into three complexity categories based on their information content: (1) Simple visualizations that required a single query to be issued by the analysis platform against a database. (2) Complex visualizations that require multiple queries to be issued against the database to create the visualization and (3) Visualizations that require domain specific inference and expertise of SMEs to construct the query, such as *Savings from a 25% reduction in potentially avoidable ER visits*. Such a query would require domain expertise to classify which visits were potentially avoidable, and what underlying measures would be used to calculate the potential savings.

Figure 14 shows the distribution of the 150 statically defined visualizations by the different analysis themes and their complexity category.

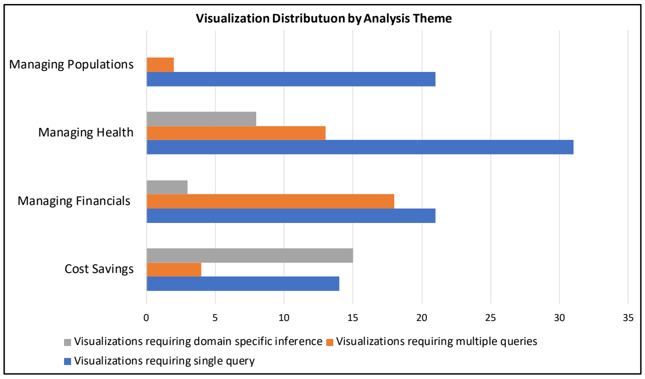


Figure 14: Visualization Distribution By Analysis Theme (Best viewed in color).

Figure 15 provides Conversational BI coverage by analysis theme. This coverage is computed in terms of the number of visualizations that can be accessed through the conversational interface with user interaction across one or more turns of conversation (or iterations). Information for visualizations that have been statically defined using multiple queries (category 2 complexity) can be accessed over multiple turns of the conversation one for each query as long as the query falls under one of the identified BI patterns used to model the intents. Mostly visualizations that require domain specific inference or expertise from SMEs (Complexity category 3) are not covered by the current implementation of our system. The focus of our current work is on supporting the typical BI patterns which cover the vast majority of the workload for BI applications. Out of a total of 150 statically defined visualizations our conversational BI system covers 125 (83.34%) and the remaining 16.66% are visualizations that require inference. We leave further exploration of visualizations that require domain inference and customized queries to generate the same using SMEs, as future work.

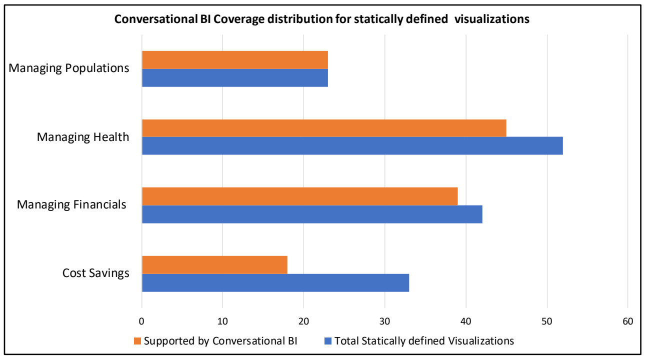


Figure 15: Conversational BI coverage by analysis theme for statically defined visualizations (Best viewed in color).

5.2 User study

We conducted a detailed user study on the pre-release version of IBM's Health Insights product with real clients, to evaluate the overall user experience and assess the usage frequency of different BI query patterns, and the accuracy of our system's intent classifier to identify these patterns as intents. The user study also provided us valuable feedback which we capture as lessons learnt(Section 5.3).

We conducted the user study over several sessions where the focus of each data exploration session was limited to a subset of the ontology relevant to different aspects of the information supported by the HI product such as Admissions, Enrollment, etc. Within each such session, we focused on identifying the relevant subset of data to visualize using appropriate filters, such as filtering the data set for specific drugs in the therapeutic class for diabetes.

Table 4 shows the results of the intent usage frequency and their F1-scores. As shown in the table, we have high F1-scores for most patterns, except the BI Operations pattern, in our initial user study. We traced the cause to the automatically generated training examples: They were not covering the different ways actual users expressed the BI operations query. Learning from this experience, we introduced a number of variations of *initial phrases* to our automatically generated training examples for this intent to help improve its classification accuracy and recall.

A large number of our users for the study specialized in the healthcare insurance domain, and were not familiar with writing structured queries against the Cognos or other business intelligence tools/platforms. Through the conversational interface, participants are able to intuitively access a series of charts/visualizations without specific knowledge of Cognos, or writing structured queries. Our conversation system was able to guide users through clarifying prompts to collect necessary information to create a chart/visualization.

Table 4: BI Query Pattern Detection Effectiveness.

BI Query Pattern	Usage Frequency	F1-Score
BI Analysis Query	32%	0.97%
BI Comparison Query	12%	0.98%
BI Trend Query	21%	0.93%
BI Ranking Query	18%	0.98%
BI Operation	17%	0.85%

5.3 Lessons learned

We learned several valuable lessons through our experience of building and evaluating our system through a user study. First, we received a very positive feedback from the users in terms of a ease of use and the ability to query the system using natural language without knowledge of schema or a programming/querying language. Second, our boot strapping mechanism is very effective in creating a rich and effective conversational workspace for BI applications. Our BI patterns cover 83.34% of statically defined visualizations and in addition enable users to access visualizations that have not been pre-defined.

Third, we realized that although the ontology-based automation of building a conversational BI system accelerates the process to building a prototype, extensive testing in the real world helps improve the system through feedback. More specifically, the feedback consisted of improving the domain vocabulary of the system for better recall by adding new variation of terms as synonyms that users actually use to refer to specific entities. Similarly, we added variations of start phrases for training examples for several intents to improve their classification accuracy. Particularly, we realized that users are not accustomed to expressing BI operations such as roll-up, drill down and pivot in natural language, and use a wide variety of variations for expressing the same. Finally, another important lesson learned for better user experience was that users preferred the system not to ask too many clarifying questions and instead preferred the system to use defaults for missing information which we have incorporated (Section 3.5.1).

6. RELATED WORK

We cover relevant related work in this section under three different categories described below.

Natural language support in existing BI tools Several existing business intelligence tools, such as Ask Data Tableau [2], Power BI [8] by Microsoft, Microstrategy [6], and the IBM's Cognos Assistant [3], support a natural language interface. However, these systems are restricted in terms of the conversational interaction they provide. A majority of these systems rely heavily on the user to drive the conversation. More specifically, they leave the onus on the user to select from a large number of options and parameters through user interfaces for getting to an appropriate visualization without much system support. Our system, on the other hand, uses information in the ontology to guide the user through meaningful conversational interactions and elicits further information to access appropriate visualizations. Further, unlike these systems our ontology-driven approach provides a formal mechanism for defining a semantically rich entity-centric view of the business model capturing both actual measures, dimensions and higher level groupings to support more complex queries catering to the querying needs of a wider range of personas. Further, our novel automated workflow for constructing a conversational BI system, enables rapid prototyping and building conversational BI systems for different domains.

Current conversational systems Existing conversational systems can be classified into three different categories [15] based on the kind of natural language interaction they support. First, are one shot question answer systems, second are general purpose chatbots such as Microsoft Cortana [19], Apple Siri [10], Amazon Alexa [9], etc. that can converse on a range of different topics such as weather, music, news or can be used to accomplish general tasks such as controlling devices, timers etc. and are agnostic to any particular domain. The third category are task-oriented agents that target tasks in specific domains such as travel, finance, healthcare and are limited in scope to specific tasks such as booking a flight, finding account balance, etc. These task oriented chatbots however fail to address the challenges involved in data exploration and derivation of meaningful insights especially for business applications. We propose an ontology-based approach for building conversational systems for supporting BI applications through natural language interfaces.

Approaches for dialogue management Recent advances in machine learning, particularly in neural networks, have allowed for complex dialogue management methods and conversation flexibility for conversational interfaces. The approaches that are commonly used in building the dialogue structure for a conversational interface are: (1) Rule-based approaches [18, 17] used in finite-state dialogue management systems are simple to construct for tasks that are straightforward and well-structured, but have the disadvantage of restricting user input to predetermined words and phrases. (2) Frame-based systems [14, 11, 16] address some of the limitations of finite state dialogue management by enabling a more flexible dialogue. Frame-based systems enable the user to provide more information as required by the system while keeping track of what information is required and ask questions accordingly. (3) Agent-based systems [12, 25, 23, 20]. Agent-based methods for dialogue management are typically statistical models and require to be trained on a corpora of prior user interactions for better adaptation. We found the frame based systems most suitable for adaptation for building a conversational BI systems to support the commonly observed BI query patterns.

7. CONCLUSIONS

In this paper, we describe an end-to-end ontology-driven approach for building a conversational interface to explore and derive business insights for a wide range of personas ranging from business analysts, to data scientists to top level executives and owners of data. We capture the domain semantics in an ontology created from the business model, and exploit the patterns in typical BI workloads to provide a more dynamic and intuitive conversational interaction to derive BI insights from the underlying data in different domains. Using the ontology, we provide an automated workflow to bootstrap the conversation space artifacts, including intents, entities, and training examples, while allowing the incorporation of user feedback and SME inputs. We implemented our techniques in *Health Insights (HI)*, and provided lessons learned, as well as a detailed evaluation.

8. REFERENCES

[1] OWL 2 web ontology language document overview. https://www.w3.org/TR/owl2-overview/.

[2] Ask Data — Tableau Software. https://www.tableau.com/products/new-features/ask-data, March 2020.

[3] Cognos Assistant. https://tinyurl.com/u3sdaxa, March 2020.

[4] IBM Cognos Analytics. https://www.ibm.com/products/cognos-analytics, March 2020.

[5] IBM Health Insights. https://www.ibm.com/us-en/marketplace/health-insights, March 2020.

[6] Kb442148: Natural language query in a nutshell in microstrategy web. https://community.microstrategy.com/s/article/Natural-Language-Query-in-A-Nutshell-MicroStrategy-11-0?language=en$_US, March$2020.

[7] Marketscan. https://www.ibm.com/products/marketscan-research-databases, March 2020.

[8] Power BI — Microsoft Power Platform. https://powerbi.microsoft.com/en-us/, March 2020.

[9] Amazon[US]. Amazon alexa. https://developer.amazon.com/alexa, 2018.

[10] Apple[US]. Siri. https://www.apple.com/ios/siri/, 2018.

[11] M. Beveridge and J. Fox. Automatic generation of spoken dialogue from medical plans and ontologies. *J. of Biomedical Informatics*, 39(5):482–499, 2006.

[12] Bing-Hwang Juang and S. Furui. Automatic recognition and understanding of spoken language - a first step toward natural human-machine communication. *Proceedings of the IEEE*, 88(8):1142–1165, 2000.

[13] S. Chaudhuri and U. Dayal. An overview of data warehousing and olap technology. *SIGMOD Rec.*, 26:65–74, 1997.

[14] K. K. Fitzpatrick, A. Darcy, and M. Vierhile. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): A randomized controlled trial. *JMIR Ment Health*, 4(2):e19, 2017.

[15] J. Gao, M. Galley, and L. Li. Neural approaches to conversational AI. *CoRR*, abs/1809.08267, 2018.

[16] T. Giorgino, I. Azzini, C. Rognoni, S. Quaglini, M. Stefanelli, R. Gretter, and D. Falavigna. Automated spoken dialogue system for hypertensive patient home management. *International Journal of Medical Informatics*, 74(2):159 – 167, 2005.

[17] S. Mallios and N. G. Bourbakis. A survey on human machine dialogue systems. In *IISA*, pages 1–7, 2016.

[18] M. F. McTear. Spoken dialogue technology: Enabling the conversational user interface. *ACM Comput. Surv.*, 34(1):90–169, 2002.

[19] Microsoft[US]. Microsoft cortana. https://www.microsoft.com/en-us/windows/cortana, 2018.

[20] A. S. Miner, A. Milstein, S. Schueller, et al. Smartphone-Based Conversational Agents and Responses to Questions About Mental Health, Interpersonal Violence, and Physical Health. *JAMA Internal Medicine*, 176(5):619–625, 2016.

[21] R. J. Moore and R. Arar. *Conversational UX Design: A Practitioner's Guide to the Natural Conversation Framework*. ACM, New York, NY, USA, 2019.

[22] A. Quamar, C. Lei, D. Miller, F. Ozcan, J. Kreulen, R. J. Moore, and V. Efthymiou. An ontology-based conversation system forknowledge bases. In *SIGMOD*, 2020.

[23] N. M. Radziwill and M. C. Benton. Evaluating quality of chatbots and intelligent conversational agents. *CoRR*, abs/1704.04579, 2017.

[24] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[25] S. J. Young, M. Gasic, B. Thomson, and J. D. Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.