Interfaz de Lenguaje Natural a Base de Datos Usando la API de Reconocimiento de Voz y Conversión a Texto DialogFlow

Natural Language Interface to Database Using the DialogFlow Voice Recognition and Text Conversion API

Julio Alejandro Villeda Maldonado, José Arturo Gaona Cuadra
Universidad Autónoma de Querétaro
Facultad de Informática
Querétaro, México
julio.villeda@live.com.mx, j.arturo.gaona@aptiv.com

Resumen — **En el presente trabajo se presenta una Interfaz de Lenguaje Natural a Base de Datos utilizando la API de reconocimiento de voz y conversión a texto DialogFlow. La capacidad de los usuarios para recuperar información almacenada en bases de datos está limitada por la falta de entendimiento de la sentencia SELECT y todas las cláusulas que la pueden componer. Si un usuario no tiene el conocimiento suficiente sobre esta sentencia no puede recuperar información de la base de datos. Es por esto que surge la necesidad de brindar medios a los usuarios que les facilite la interacción y obtención de datos con las bases de datos relacionales utilizando el lenguaje natural. Los usuarios sin experiencia en lenguajes de bases de datos se habilitarán para recuperar información con esta interfaz; en tanto que los usuarios con experiencia reducirán el tiempo de creación de consultas.**

Palabras Clave - interfaz; base de datos; lenguaje natural; aprendizaje automatizado.

Abstract — **A Natural Language Interface to Database using the DialogFlow voice recognition and text conversion API is presented in this work. The user's ability to retrieve information stored in databases is limited by the lack of understanding of the SELECT statement and the clauses that can make it up. If a user does not have sufficient knowledge about this statement, he is unable to obtain information from the database. This is why the need arises to provide a mechanism to users that facilitates the interaction and data acquisition with the databases using a natural language. Users without experience in database languages will be enabled to retrieve information with this interface; while experienced users will reduce query creation time.**

Keywords - interface; database; natural language; machine learning.

I. INTRODUCCIÓN

En los últimos años el campo de las Tecnologías de Información (TI) ha crecido a un ritmo acelerado y sus áreas de aplicaciones se han vuelto cada vez más amplias, estableciendo la base de todo tipo de aplicaciones, tanto de carácter básico como de naturaleza crucial o de alto impacto. Cada uno de estos sistemas tienen sus propios principios de interacción entre sus componentes, pero casi todos ellos comparten como punto crucial el almacenamiento de datos. Sin embargo, a través de los años se ha identificado una carencia en la recuperación de información derivado de la falta de entendimiento del lenguaje de consultas SQL, por lo que en este trabajo se presenta una interfaz que permite tanto a usuarios con experiencia como a usuarios sin experiencia poder interactuar con los datos almacenados en bases de datos relacionales, particularmente para recuperar información empleando el lenguaje natural sin necesidad de que los usuarios tengan dominio del lenguaje de consultas SQL.

En este trabajo se explora una alternativa a las Interfaces de Lenguaje Natural a Base de Datos (ILNBD) que existen al día de hoy. La interfaz que aquí se presenta está realizada con la API de reconocimiento de voz y conversión a texto DialogFlow. Con esta API la interfaz ofrece un poderoso Entendimiento del Lenguaje Natural (ELN) o NLU por sus siglas en inglés (Natural Language Understanding), siendo incluso multilenguaje. El motor procesa y entiende entradas en lenguaje natural, y al ser un analizador muy robusto entiende y procesa los matices y variantes del lenguaje español. A través de entrenamiento y el uso de "Machine Learning", la interfaz cubre las carencias que han presentado otras interfaces en el pasado, particularmente las interfaces orientadas al dominio y la dificultad de mantener el diccionario. Con la interfaz que aquí se presenta se genera un modelo de lenguaje más variado y más robusto que empareja mejor la sentencia de entrada del usuario.

En este trabajo primero se hizo una breve revisión de los trabajos existentes en el área de las Interfaces de Lenguaje Natural a Base de Datos, después se hizo una descripción del problema a desarrollar para presentar la fundamentación teórica que respalda al trabajo realizado. En seguida se muestran los pasos realizados para el desarrollo de este proyecto. Por último, se presentan los resultados, conclusiones y trabajo a futuro.

II. TRABAJOS RELACIONADOS

En esta sección se realizó un análisis de las ILNBD desarrolladas en los últimos 5 años, observando las interfaces propuestas, las técnicas empleadas, los resultados obtenidos y el trabajo por realizar en cada una de ellas.

En primer lugar, en el trabajo titulado "Translating Controlled Natural Language Query into SQL Query using Pattern Matching Technique" se propuso un sistema para recuperar datos desde una base de datos usando el lenguaje hindú, integrando un analizador morfológico y un analizador de

978-1-7281-5555-5/19/$31.00 ©2019 IEEE

grupo de palabras para obtener las palabras principales de una consulta de entrada en leguaje hindú. En el trabajo se empleó la técnica "Pattern Matching" para encontrar las palabras clave y empleó una interfaz del lenguaje hindú controlado con algunas características de consultas sugeridas para reducir la ambigüedad y reducir el tiempo de entrada de la consulta hindú. En este sistema se empleó un diccionario orientado al dominio para determinar el tipo de palabras claves usadas en una consulta de entrada y se presentó una importante limitación, la cual radicó en la dificultad de mantener la base de datos y el diccionario orientado al dominio cuando la interfaz se aplicaba a base de datos muy grandes. Dentro de los trabajos a futuro de esta investigación se buscó el desarrollo de una interfaz independiente de su dominio y además que dicha interfaz fuera multilenguaje [1].

Posteriormente, en el trabajo titulado "Translation of natural language queries to structured data sources" se desarrolló un prototipo de interface de usuario de lenguaje natural a consultas SQL para base de datos. Este prototipo requiere la estructura de datos describiendo la base de datos, en particular, requiere información acerca de las tablas y los nombres de los campos. La interfaz tiene algunas restricciones, tal como no usar pronombres personales ni demostrativos y se recomienda empezar las consultas con palabras especiales tales como: "Obtener", "Mostrar", "Enumerar", entre otras [2].

En el trabajo titulado "Natural Language Interface to Database using Modified Co-ocurrence Matrix Technique" se discutió el diseño y la implementación de un sistema que utiliza el método de la matriz de co-ocurrencia de palabras modificadas para proveer acceso a una base de datos utilizando el idioma inglés. En este trabajo se emplearon varias técnicas tales como: análisis sintáctico, extracción de palabras clave, detener la eliminación de palabras, generación de la matriz de co-ocurrencia, el uso de WordNet, algoritmo de derivación y mapeo semántico. Este sistema que se desarrolló da respuestas correctas a consultas sencillas, consultas con condiciones lógicas y funciones de agregado, sin embargo, el sistema no soporta todas las formas de las consultas de SQL [3].

En el trabajo "Implementation of Prototype of Natural Language Interface to Database" se realizó un traductor de lenguaje natural a SQL, integrando tanto el esquema de la base de datos como información del dominio, la cual se ingresa por medio de un archivo CSV. El sistema propuesto utiliza las preposiciones, los artículos y principales adverbios del lenguaje español como parte de su análisis y se utiliza un algoritmo de aproximación, que de acuerdo a las palabras que se entienden de la entrada del usuario, se busca generar una consulta SQL. En dicha investigación se enfocaron en resolver la dificultad que tienen los usuarios no técnicos para acceder a la base de datos, implementando para ello una interfaz del lenguaje natural. Como resultado de las pruebas que se realizaron en dicha interfaz se comprobó que el prototipo es de utilidad, pero es necesario mejorarlo enfocándose en incrementar la capacidad de generación de código SQL utilizando sentencias del tipo "GROUP BY"-"HAVING", con funciones de SQL y consultas anidadas, incluso siendo deseable brindar una retroalimentación al usuario indicándole qué partes de su consulta no son precisas [4].

En el trabajo llamado "An independent-domain Natural Language Interface for Relational Database: Case Arabic Language" se presentó la arquitectura de una interfaz genérica para consultar bases de datos usando lenguaje arábigo. La interfaz que propusieron funciona independientemente del dominio de la base de datos y tiene la capacidad para mejorar a través de la experiencia su base de conocimiento. Dicha interfaz estaba basada en "Machine Learning" y procesamiento de lenguaje natural. Este sistema ofreció algunas ventajas, primero que el sistema era independiente del lenguaje, dominio y modelo de la base de datos, además de que es capaz de mejorar automáticamente su base de conocimiento a través de la experiencia [5].

Por último, en el trabajo titulado "Seq2SQL: Generating Structured Queries From Natural Language using Reinforcement Language" se propuso un sistema Seq2SQL, una red neuronal profunda para traducir preguntas del lenguaje natural a su correspondiente consulta en SQL. El modelo que se presentó aprovecha la estructura de las consultas de SQL para reducir significativamente el espacio de salida de las consultas generadas. Además, el modelo integra un sistema de recompensas sobre la base de datos para aprender una política para generar partes no ordenadas de la consulta. Este modelo es menos adecuado para la optimización a través de la pérdida de entropía cruzada. Es importante destacar que, como parte de este trabajo, se publicó WikiSQL, el cual es un conjunto de datos de 87,726 ejemplos anotados a mano de preguntas y consultas de SQL distribuidas a través de 26,375 tablas de Wikipedia. Este conjunto de datos fue utilizado para entrenar el modelo y mejoró la confiabilidad de ejecución del 35.9% al 60.3% y confiabilidad de forma lógica del 23.4% al 49.2% respecto a interfaces similares diseñadas previamente [6].

Después de revisar los trabajos previos se identificó que las Interfaces de Lenguaje Natural a Base de Datos aún no son 100% confiables. Algunas de ellas encuentran su principal limitante en el dominio de la base de datos, aunque, por otro lado, las investigaciones en el área de Inteligencia Artificial, tales como: redes neuronales, "Pattern Matching" y "Machine Learning" son bastantes promisorias, por lo que en este trabajo se busca explorar una alternativa en estas áreas con DialogFlow.

III. DEFINICIÓN DEL PROBLEMA

La falta de dominio del lenguaje SQL, específicamente de la sentencia SELECT y de las cláusulas que la componen en conjunto con todas sus variantes, genera importantes limitantes para recuperar información de bases de datos relacionales tanto para usuarios con experiencia como para usuarios sin experiencia en el lenguaje SQL. Se considera una base de datos donde se han colocado ciertas tablas con sus respectivos campos, y que dichas tablas están propiamente normalizadas. Si un usuario quiere acceder a los datos de estas tablas, se necesita que dicho usuario tenga experiencia en el lenguaje de consultas SQL. Es por esto que se buscan métodos que permitan a los usuarios sin experiencia en SQL poder interactuar con la base de datos sin que tenga que conocer el lenguaje SQL, y únicamente tener conocimiento del dominio de la base de datos.

El Procesamiento de Lenguaje Natural (PLN) es uno de estos métodos, sin embargo, las computadoras no entienden el lenguaje natural, por lo que el desarrollo de una ILNBD

978-1-7281-5555-5/19/$31.00 ©2019 IEEE

permitirá habilitar la comunicación entre los usuarios y las bases de datos usando el lenguaje natural. El objetivo de esta ILNBD es eliminar la necesidad de que los usuarios comprendan y dominen el lenguaje SQL y que puedan recuperar información desde bases de datos relacionales; al mismo tiempo permitir que los usuarios con experiencia mejoren la cantidad de consultas que puedan hacer reduciendo el tiempo en el que las generan.

IV. Fundamentación Teórica

El marco teórico, que se desarrolla a continuación, permite conocer los conceptos básicos necesarios para el entendimiento del desarrollo de este trabajo. Primero se parte con la descripción de la cláusula SELECT; después, se exploran los conceptos de Procesamiento de Lenguaje Natural (PLN), las Interfaces de Lenguaje Natural (ILN) y, por último, las Interfaces de Lenguaje Natural a Base de datos (ILNBD) y algunos ejemplos, para conocer las arquitecturas y sus componentes.

A. Claúsula SELECT

Recuperar información de las bases de datos relacionales es una tarea que requiere por lo menos del dominio de la sentencia SELECT del Lenguaje de Consultas Estructurado (SQL, por sus siglas en inglés Structured Query Language). Este lenguaje tiene una estructura y formato específico ya predefinido, que consta de 2 partes: (1) El Lenguaje de Definición de Datos (DDL, por sus siglas en inglés Data Definition Language y, (2) Lenguaje de Manipulación de Datos (DML, por sus siglas en inglés Data Manipulation Language) que incluye aquellas sentencias que sirven para manipular o procesar los datos, como por ejemplo la inserción (INSERT), borrado (DELETE), actualización (UPDATE) y recuperación (SELECT) de datos en las tablas.

De estas 4 sentencias, la que se estará abordando a detalle en este trabajo es la sentencia SELECT. La sintaxis completa de esta sentencia es compleja, pero se puede resumir de la siguiente manera:

SELECT select_list

[FROM table_source]

[WHERE search_condition]

[GROUP BY group_by_expression]

[HAVING search_condition]

[ORDER BY order_expression [ASC|DESC]]

Debido a la complejidad de la sentencia SELECT, se describe cada una de las cláusulas que la componen, incluyendo algunas de sus variantes:

1) SELECT: Determina el número y nombre de las columnas a incluir en la consulta. A continuación, se pueden observar las variantes de la cláusula SELECT:

- ALL: Específica que el conjunto de resultados puede incluir filas duplicadas, este es el valor predeterminado.

- DISTINCT: Específica que el conjunto de resultados solo puede incluir filas únicas. Los valores nulos se consideran iguales desde el punto de vista de esta palabra reservada.

- TOP(Expresión): Indica que el conjunto de resultados de la consulta solamente deberá un primer conjunto o porcentaje de filas especificado. Por lo tanto, "Expresión" puede ser un número o un porcentaje del total de las filas.

- "*" (Asterisco): Específica que se deben de devolver todas las columnas de todas las tablas del origen de datos. Las columnas serán mostradas en el orden en el que se encuentren en el origen de datos.

- <SELECT LIST>: Se específica una o varias columnas a mostrarse como parte del conjunto de datos contenidos en las tablas o vistas seleccionadas como origen de datos. Esta lista de columnas está separada por comas y máximo se podrán seleccionar 4,096 columnas.

2*) FROM:* Identifica las tablas de origen de los datos, puede ser una sola tabla o múltiples tablas. Normalmente, está cláusula es requerida en la instrucción SELECT. Sin embargo, la excepción surge cuando no hay columnas de tablas enumeradas, y los únicos elementos que se muestran son las literales, variables o expresiones aritméticas. Al usar la cláusula FROM se puede usar la cláusula JOIN para indicar que se va a ejecutar la operación de combinación especificada entre los orígenes de la tabla o vistas indicados y tener como resultado una tabla combinada. Se pueden tener diferentes tipos de combinaciones tales como INNER, FULL, LEFT, RIGHT y CROSS:

3) WHERE: Restringe el número de filas por una o varias condiciones. Algunas de las variantes de esta cláusula se mencionan a continuación*:*

- Restringir filas con una igualdad simple

- Restringir filas que contienen un valor como parte de una cadena.

- Buscar filas utilizando operadores de comparación, tales como mayor que, menor que, mayor o igual que, y menor o igual que.

- Buscar filas utilizando operadores lógicos tales como OR donde al menos una condición debe cumplirse o AND donde todas las condiciones deben de cumplirse.

- Buscar filas que están dentro de una lista de valores usando la cláusula IN.

- Buscar filas que estén comprendidas entre un rango de valores con la sentencia BETWEEN.

4) GROUP BY: Agrupas filas por valores de columnas comunes. Especifica una columna o un cálculo no agregado en una columna. Esta columna puede pertenecer a una tabla, una tabla derivada o una vista. La columna debe aparecer en la cláusula FROM de la instrucción SELECT, pero no es necesario que aparezca en la lista SELECT. Aun así, deben incluirse en la lista GROUP BY todas las columnas de la tabla o la vista de cualquier expresión no agregada de la lista de <select>.

5) HAVING: Restringe el número de filas cuando se usa agrupación de filas. Especifica una condición de búsqueda para un grupo o agregado. HAVING solo se puede utilizar con la instrucción SELECT. Normalmente, HAVING se usa con una cláusula GROUP BY. Cuando no se usa GROUP BY, hay un solo grupo implícito agregado.

978-1-7281-5555-5/19/$31.00 ©2019 IEEE

6) ORDER BY: Ordena el resultado por una o más columnas. No hay ningún límite en cuanto al número de columnas de la cláusula ORDER BY.

Se puede ver que la cláusula SELECT debido a su naturaleza implica mayor aprendizaje de cada uno de los diferentes componentes que la pueden integrar. Sin embargo, si un usuario no tiene este conocimiento no puede obtener información de la base de datos [3].

B. *Procesamiento de Lenguaje Natural (PLN)*

Derivado de la falta de capacidad para recuperar información de base de datos relacionales tanto para usuarios con experiencia como usuarios sin experiencia, es que se busca aplicar el PLN (NLP por sus siglas en inglés Natural Language Processing) para la obtención de información de base de datos. PLN es un área que estudia la interacción entre una computadora y un usuario. Tanto el entendimiento del lenguaje natural como el procesamiento del lenguaje mismo son partes fundamentales del PLN. Cuando se índica "Lenguaje Natural" se refiere al lenguaje que es usado para comunicarse diariamente por los humanos, lenguajes tales como el español o el inglés. Una de las principales dificultades de los lenguajes naturales es que han evolucionado conforme han pasado de generación a generación y son difíciles de precisar con reglas explícitas [7].

En este trabajo se está empleando el término PLN para cubrir cualquier tipo de procesamiento o manipulación computacional del lenguaje natural, por ejemplo, la Traducción Automatizada (Machine Translation), que se enfoca a convertir texto de un lenguaje humano a otro automáticamente. Analizar preguntas del lenguaje natural a SQL puede ser visto como una forma especial de traducción automatizada [8].

En este sentido, hacer preguntas a bases de datos en lenguaje natural es un método fácil y muy conveniente de acceder a los datos, especialmente para usuarios que no entienden lenguajes de consultas como SQL. PLN y Traducción Automatizada es el proceso por el cual la consulta en lenguaje natural de un usuario es convertida a una consulta SQL basado en la sentencia ingresada por el usuario [9].

C. *Interfaces de Lenguaje Natural a Bases de Datos (ILNBD)*

Dado que las computadoras no pueden entender el lenguaje natural, es necesario desarrollar interfaces. Una interfaz es una conexión que puede ser física o lógica, entre una computadora, un dispositivo periférico o un enlace de comunicaciones.

Las Interfaces de Lenguaje Natural (ILN) o NLI por sus siglas en inglés (Natural Language Interfaces) son interfaces que proveen los medios al usuario para interactuar con la computadora a través del lenguaje natural creando una conexión lógica entre las dos partes [3].

En general, la tarea de escribir una ILN involucra una gramática del lenguaje natural y una correspondiente estructura léxica, así como un conjunto de traducciones semánticas para el sistema objetivo. Las ILN pueden ser clasificadas en 2 categorías [10]:

1) Aquellas basadas en un seudo-lenguaje, donde un usuario deberá de aprender un lenguaje de comandos que sea similar al lenguaje natural.

2) Aquellas basadas en la premisa que un usuario debe poder expresarse en cualquier forma que sea natural para él y el sistema le dará sentido a su expresión de entrada.

Fig. 1 Conversión de lenguaje natural a consulta SQL

Las Interfaces de Lenguaje Natural a Base de Datos (ILNBD) o NLIDB por sus siglas en inglés (Natural Language Interface to Database) es un campo del PLN donde los datos de bases de datos relacionales son recuperados usando preguntas en el lenguaje natural. En la Fig. 1 se muestra un esquema de la traducción del lenguaje natural a una consulta SQL [3].

Si se supone que se quiere obtener toda la información acerca de los estudiantes en la tabla ESTUDIANTE, para ello se utilizaría una consulta SQL tal como:

"SELECT * FROM ESTUDIANTE;"

Una persona que no conoce SQL no tiene forma de obtener la información. Al usar una ILNDB, la consulta que se ejemplificó arriba puede ser rescrita como:

"Dame toda la información de la tabla ESTUDIANTE"

Ambas sentencias regresarán el mismo resultado, la gran diferencia es que con el uso de una ILNDB una persona sin conocimiento de SQL podrá recuperar los datos almacenados en la base de datos [9].

D. *Componentes de ILNBD*

Las ILNBD tienen 2 componentes principalmente [11]:

1) Componente Lingüístico: Produce una consulta lógica intermedia de la consulta de lenguaje natural ingresada. El componente lingüístico realiza análisis morfológico, sintáctico y semántico. Como resultado produce una consulta lógica.

2) *Componente Base de Datos*: Este componente realiza las funciones tradicionales de administración de base de datos. En esta etapa se produce una consulta de base de datos al hacer una correspondencia de las palabras de entrada en el lenguaje natural a las cláusulas correspondientes en la consulta de base de datos. La generación de la consulta está dividida en cuatro fases, donde cada etapa manipulará únicamente una parte específica de la consulta de base de datos.

a. Esta fase trata con la parte de la consulta lógica, que corresponde a los nombres de los campos para construir la cláusula SELECT.

b. En esta etapa se construye la cláusula FROM al seleccionar la parte de la consulta lógica que corresponde al nombre de la tabla o grupo de tablas.

c. En esta etapa se construye la cláusula WHERE al identificar cualquier condición que esté presente en la consulta del lenguaje natural.

d. En la última etapa, se selecciona la porción de la consulta lógica que corresponde al ordenamiento de la presentación de datos recuperados.

978-1-7281-5555-5/19/$31.00 ©2019 IEEE

E. Arquitecturas de ILNBD

Existen principalmente 4 arquitecturas de ILNBD que se describen a continuación [11]:

1) Emparejamiento de patrones: Los patrones son definidos usando algunas reglas establecidas manualmente, posteriormente la consulta en lenguaje natural es mapeada a esas reglas. Este tipo de interfaces solo pueden construir una consulta cuando un patrón se iguala con una entrada de usuario dada.

2) Interfaces basadas en sintaxis: La consulta del lenguaje natural es analizada sintácticamente y entonces se mapea el árbol de sintaxis en la estructura de cualquier lenguaje de consultas formal. El árbol de análisis contiene toda la información acerca de la estructura de la consulta y es directamente mapeada a cualquier lenguaje de consulta de base de datos y no hay análisis semántico. Los enfoques basados en sintaxis proveen información detallada acerca de la estructura de una sentencia lo cual es una de las mayores ventajas de usar este tipo de interfaces. Un árbol de análisis da la información detallada acerca de la estructura de una sentencia, pero el problema es que no se pueden mapear todos los nodos, ya que algunos nodos son dejados fuera dado que no agregan ningún significado semántico, y no siempre es claro cuales nodos deben ser mapeados y cuáles no.

3) Interfaces de Gramática Semántica: En este tipo de interfaces la consulta de la base de datos es generada al analizar la consulta de entrada y generar un árbol de análisis de la consulta del lenguaje natural y también agregando algo de significado lógico al árbol de análisis. Entonces se mapea este árbol de análisis a una consulta de base de datos. La entrada del usuario es analizada sin conceptos sintácticos y la gramática semántica es mejor situada para sistemas de dominio específico, por lo tanto, es difícil exportar a otro dominio.

4) Interfaces de Representación Intermedia: Estas interfaces transforman la consulta con significado interno descrito por una pregunta formulada en lenguaje natural. Actualmente la mayoría de ILNBD son de este tipo porque esta arquitectura convierte primero la consulta de lenguaje natural en una consulta lógica intermedia al pasar por diferentes análisis: semánticos, sintácticos y morfológicos. La consulta lógica es entonces trasladada a una expresión en el lenguaje de consultas de la base de datos, y evaluada contra la base de datos. En el enfoque del lenguaje de representación intermedia, la interfaz puede ser dividida en 2 partes:

a. Una parte empieza desde una sentencia hasta la generación de una consulta lógica.

b. La segunda parte empieza desde una consulta lógica hasta la generación de una consulta de base de datos.

Además, la consulta lógica intermedia es independiente del dominio de la base de datos, de este modo, la consulta puede ser exportada a diferentes lenguajes de base de datos, así como de dominios tales como sistemas expertos y sistemas operativos.

F. Ejemplos de ILNBD

En esta sección se describen algunos ejemplos de ILNDB que se han desarrollado y destacado de manera importante en el ramo de la investigación de sistemas de PLN, a continuación, se muestran 4 ejemplos:

1) LIFER/LADDER [12]: Fue uno de los primeros sistemas PLN, fue diseñado como una ILNBD de información acerca de los barcos de la marina de Estados Unidos. Esta interfaz usaba una gramática semántica para analizar preguntas y consultar una base de datos distribuida. Este sistema podía únicamente soportar consultas a una tabla, o consultas a múltiples tablas con condiciones de unión sencillas.

2) QUESTION-ANSWERING SYSTEM [13]: Propuso un método para construir una interfaz de respuestas-preguntas el cual es integrado con un sistema de búsqueda para libros electrónicos en una librería. Los usuarios pueden usar simples preguntas en inglés para buscar en la biblioteca información acerca de los libros electrónicos, tales como título, autor, lenguaje, categoría, publicador. En este proyecto de investigación, el foco principal está sobre problemas fundamentales en el procesamiento de consultas del lenguaje natural, tales como enfoques de análisis de sintaxis, representación de sintaxis, representación semántica y, reglas de transformación para estructura de sintaxis de estructura semántica.

3) NaLIX - Natural Language Interface for an XML-Database [14]: Es una interfaz de lenguaje natural interactiva a lenguaje XML y consiste de un clasificador de árbol de análisis, validador y un traductor, el cual es requerido para la traducción de consultas de lenguaje natural a consultas XML. Este consiste en 3 pasos:

a. Clasificador de Árbol de Análisis: El cual identifica la palabra clave o frase en la consulta de lenguaje natural que puede ser traducida en su consulta XML correspondiente.

b. Validador del Árbol de Análisis: Checa el árbol de análisis y checa si el árbol obtenido puede ser mapeado en su correspondiente consulta XML. En esta etapa también se asegura que las palabras claves, atributos y valores numéricos estén en la base de datos.

c. Traductor: Produce una salida efectiva.

4) WASP - Word Alignment-based Semantic Parsing [15]: Es una interfaz de lenguaje natural desarrollada recientemente para representación de consultas. Esta interfaz tiene la intención de lograr el objetivo más amplio, construir una representación completa, formal, simbólica y significativa de una sentencia en lenguaje natural.

G. Reconocimiento de Voz

El primer componente del Reconocimiento de Voz, es naturalmente, la voz, la cual debe ser convertida de un sonido físico a una señal eléctrica empleando un micrófono, y después convertirla a un dato digital con un convertidor análogo a digital. Una vez digitalizada la señal puede ser usada para transcribir el audio a texto.

La mayoría de sistemas de reconocimiento de voz modernos están basados en lo que se conoce como "Hidden Markov Model" (HMM). En un sistema HMM típico, la señal de voz es dividida en fragmentos de 10 milisegundos. El espectro de poder de cada fragmento, que es esencialmente una trama del poder de

978-1-7281-5555-5/19/$31.00 ©2019 IEEE

la señal como una función de frecuencia, es comparada contra un vector de números reales conocido como coeficientes Cepstral. La dimensión de dicho vector generalmente es pequeña, algunas veces no mayor de 10, aunque los sistemas más confiables tienen dimensiones de 32 o más. La salida final del sistema HMM es una secuencia de esos vectores [16].

Para decodificar la voz en texto, los grupos de vectores son empatados a uno o más fonemas (unidad fundamental del habla). Este cálculo requiere entrenamiento, dado que el fonema de una persona varia de persona a persona, e incluso varia de una declaración dicha por la misma persona de un momento a otro. Un algoritmo especial es entonces aplicado para determinar las palabras más probables que produce la secuencia dada de fonemas.

En muchos sistemas modernos, las redes neuronales son utilizadas para simplificar la señal de voz usando técnicas para transformación de características y reducción de dimensionalidad antes del reconocimiento HMM. Detectores de Actividad de Voz (VAD por sus siglas en inglés Voice Activity Detectors) también son usados para reducir una señal de audio únicamente a las porciones que son probable que contengan voz. Esto previene que el reconocedor desperdicie tiempo analizando partes innecesarias de la señal.

V. DESARROLLO

En este trabajo se desarrolló una ILNBD, cuya premisa es que un usuario se exprese en cualquier forma que sea natural para él y la interfaz le dé sentido a su sentencia de entrada. La arquitectura que esta interfaz tiene es el Emparejamiento de patrones [11]. Para declarar los patrones, se definen algunas reglas preestablecidas usando DialogFlow, la cual es una plataforma donde se puede construir interfaces conversacionales, y posteriormente se empareja la consulta en lenguaje natural con esas reglas.

A continuación, se muestran los pasos que se siguieron para la realización de la ILNBD utilizando DialogFlow:

A. Crear una cuenta de DialogFlow

Para registrar una cuenta de DialogFlow es necesario contar con una cuenta de Google e ir a https://dialogflow.com/. Automáticamente se habilitan dos herramientas de Google: "Machine Learning de Google" y "Google Cloud Speech-To-Text", mismas que son empleadas en este trabajo.

B. Crear un Agente de DialogFlow

DialogFlow permite de manera fácil lograr una experiencia conversacional al controlar el ELN. Con DialogFlow es necesario crear agentes, los cuales entienden los vastos y variados matices del lenguaje humano y traducen eso a un significado estructurado y estándar que las aplicaciones y servicios pueden entender.

Un agente procesa entradas de usuario y las convierte en datos estructurados que se usan para regresar una respuesta apropiada. Esto se define dentro de uno o varios "Intentos", los cuales definen como la entrada del usuario es mapeada a su respuesta correspondiente.

C. Crear Intentos

Un Intento básico está compuesto de los siguientes componentes:

1) Frases de entrenamiento: Son frases de ejemplo de lo que los usuarios puedan decir. DialogFlow usa estas frases de entrenamiento y las expande a muchas más frases similares para construir un modelo de lenguaje que de forma precisa empareje la entrada del usuario.

2) Acciones y parámetros: Para mejorar un modelo de lenguaje de un Intento, se puede estructurar las frases de entrenamiento con "Entidades", las cuales son categorías de datos con las que se quiere que DialogFlow empareje. Esto le dice a DialogFlow que se requiere un particular tipo de dato y no solo emparejar de manera literal la entrada del usuario. DialogFlow extrae las Entidades que encuentra como parámetros de las frases de entrenamiento. Se puede procesar estos parámetros en una lógica llamada "cumplimiento" para personalizar una respuesta del usuario.

3) Respuestas: Define un texto, dialogo, o respuesta visual para el usuario. Para enviar respuestas, se puede usar respuestas propias de DialogFlow o llamar el modo "cumplimiento" para procesar los datos extraídos y regresar una respuesta a DialogFlow.

a. Cuando los usuarios dicen algo, conocido como una expresión, el agente hace coincidir la expresión con un Intento apropiado, también conocido como clasificación de intención. Se concuerda un Intento si el modelo de lenguaje para ese intento puede coincidir de manera cercana o exacta con la expresión del usuario. El modelo de lenguaje se define especificando frases de entrenamiento o ejemplos de cosas que los usuarios quieran decir. DialogFlow toma estas frases de entrenamiento y las expande para crear el modelo de lenguaje del intento.

b. Una vez que el agente coincide con un Intento, extrae los parámetros, puede ser un color, nombre, fecha o una serie de otras categorías de datos llamados "Entidades". DialogFlow define un conjunto grande de entidades que categorizan los parámetros extraídos, o el usuario puede crear los suyos propios. También define qué extraer en sus frases de entrenamiento, anotando partes específicas de las frases de entrenamiento para especificar qué parámetros desea extraer.

c. Luego, envía una respuesta para finalizar la conversación. DialogFlow incluye un controlador de respuestas fácil de usar para devolver respuestas simples, generalmente estáticas. Si desea devolver respuestas más completas y dinámicas, puede usar la lógica llamada "Cumplimiento" para procesar los parámetros extraídos y devolver una respuesta que sea más útil.

En la Fig. 2 se observa como los componentes interactúan entre sí:

978-1-7281-5555-5/19/$31.00 ©2019 IEEE

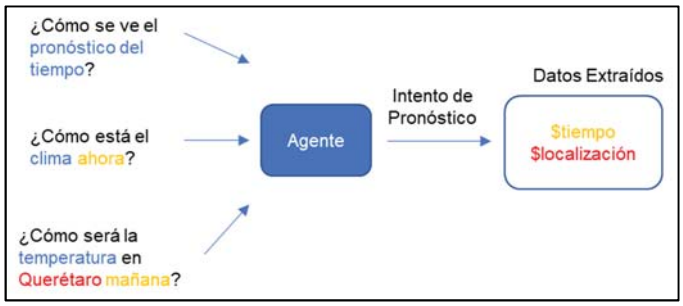

Fig. 2 Intercomunicación de los componentes de un Intento; Fuente: [17]

Cada intento tiene un controlador de respuesta incorporado que puede devolver respuestas después de que la intención coincida. Sin embargo, esta característica solo le permite construir respuestas que son estáticas o que tienen una lógica mínima. La mayoría de las veces, se usa el Cumplimiento para procesar la intención primero y luego devolver una respuesta más inteligente o útil. El Cumplimiento es una lógica personalizada que se implementa como un webhook, que los servicios solicitan, procesan y devuelven respuestas.

El agente normalmente utiliza una combinación de los dos tipos de respuesta, utilizando el controlador de respuesta incorporado de Dialogflow para devolver respuestas simples o estáticas, y luego delegar en el cumplimiento para generar respuestas más adecuadas, variadas o adecuadas.

A continuación, se describe como trabaja un simple agente.

a) El agente hace coincidir la expresión de un usuario con un intento.

b) El agente extrae parámetros de la declaración del usuario y llama a su Cumplimiento con una carga JSON que contiene los parámetros junto con una gran cantidad de otra información útil sobre la intención.

c) El cumplimiento procesa cualquier información necesaria que necesite desde el JSON, como llamar a otra API REST con los parámetros extraídos.

d) El Cumplimiento construye una respuesta y la devuelve a Dialogflow para representarla al usuario. Esta respuesta puede ser un texto simple o una respuesta enriquecida, como una tarjeta con una imagen. Dialogflow devuelve automáticamente una respuesta utilizando el controlador de respuesta incorporado de la intención como una alternativa. A veces es útil definir una respuesta en el agente y en cumplimiento para aprovechar esta función.

D. Crear Entidades

Las Entidades son mecanismos de DialogFlow para identificar y extraer datos útiles de entradas del lenguaje natural. Define como los datos serán extraídos de cada sentencia. Las entidades categorizan partes importantes de las sentencias de los usuarios. Con las entidades se empareja una categoría en lugar de una sentencia específica, lo cual brinda mayor flexibilidad. Representa el diccionario de datos, tablas y campos presentes en la base de datos.

Mientras que las intenciones le permiten al agente entender la motivación detrás de una entrada de usuario en particular, las entidades se usan para seleccionar información específica que mencionan los usuarios, nombres de tablas o nombres de campos. Cualquier dato importante que se requiera obtener de la solicitud del usuario tendrá una entidad correspondiente.

1) Tipo de entidad: Define el tipo de información que desea extraer de la entrada del usuario. Por ejemplo, “Tabla” podría ser el nombre de un tipo de entidad.

2) Entrada de entidad: Para cada tipo de entidad, hay muchas entradas de entidad, las cuales proporcionan un conjunto de palabras o frases que se consideran equivalentes. Por ejemplo, si “Tabla” es un tipo de entidad, podría definir estas tres entradas de entidad: (1) Empleados, (2) Alumnos y, (3) Profesores.

Hay 4 tipos de Entidades que se describen a continuación:

1) Entidades del sistema: Dialogflow incluye numerosas entidades del sistema, que permiten a los agentes extraer información sobre una amplia gama de conceptos sin ninguna configuración adicional. Por ejemplo, para extraer fechas, horas y ubicaciones de las entradas de lenguaje natural.

2) Entidades desarrolladas: Si se necesita extraer información sobre conceptos más allá de los cubiertos por las entidades del sistema de Dialogflow, puede definir sus propios tipos de entidades de desarrollador.

3) Entidades de sesión: También es posible definir tipos de entidades que se aplican solo a una conversación específica. Por ejemplo, puede crear un tipo de entidad para representar las opciones sensibles al tiempo disponibles para un usuario en particular al realizar una reserva. Estos son llamados tipos de entidad de sesión.

4) Gestionar entidades: Dialogflow también proporciona herrBHamientas para administrar entidades, incluidos mecanismos para exportar y cargar datos de entidades y modificar entidades mediante API.

En la Fig. 3 se puede ver de manera gráfica la interacción entre los componentes descritos:

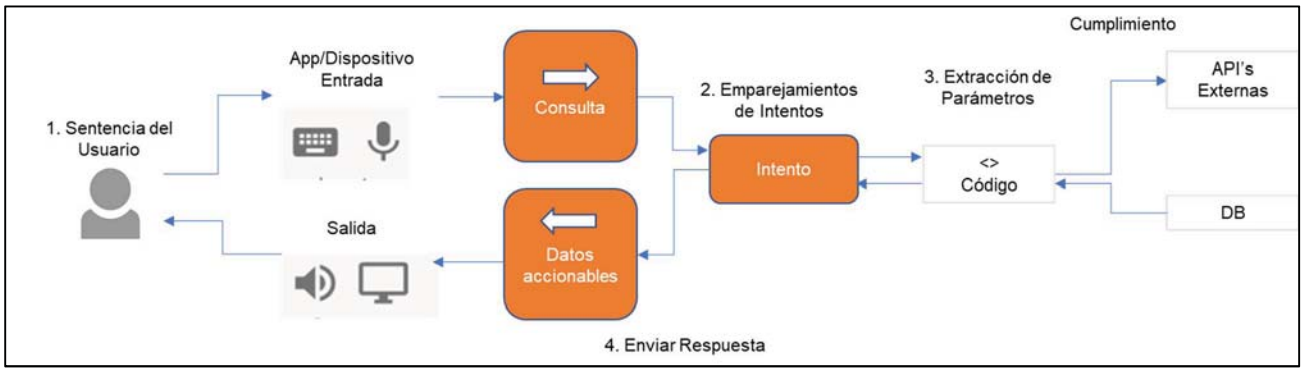

Fig. 3 Diagrama de un agente simple; Fuente: [18]

E. Implementación

A continuación, se describe el proceso general de cómo se integró DialogFlow en la ILNDB, mismo que se resume en la Fig. 4.

1) Creación de la sentencia de usuario usando voz en lenguaje natural y conversión a texto con DialogFlow.

2) Emparejamiento de Intentos: Empareja la sentencia del usuario para entregar una respuesta.

a. Frases de entrenamiento: Valida con que intento tiene mayor porcentaje de emparejamiento y determina el intento que sea más confiable de acuerdo a las frases de entrenamiento.

b. Acción y parámetros: Extrae las tablas y campos para estructurar la consulta SELECT.

c. Respuesta: Es la consulta SQL que será ejecutada en el motor de base de datos SQL Server.

3) Ejecución de Consulta Generada: DialogFlow genera la consulta lista para ser ejecutada en SQL Server.

4) Presentación de Resultados: Después de generar la consulta SQL generada en DialogFlow se presentan los resultados generados.

5) Evaluación de Resultados: Se valida que los resultados obtenidos vayan de acuerdo a la sentencia emitida por el usuario, en caso de que no sea así deberá refinar la consulta, así como las frases de entrenamiento e intentos predefinidos.

F. Integración con Google Assistant

Las acciones de Dialogflow en la integración de Google le permiten llegar a los usuarios a través de altavoces activados por voz como Google Home, teléfonos, iPhones y Android elegibles, y en el futuro, todas las experiencias donde el Asistente de Google esté disponible.

Para probar el agente en dispositivos compatibles, solo es necesario iniciar sesión en el dispositivo con la misma cuenta de Google que utilizó para iniciar sesión en DialogFlow y las Acciones en la Consola de desarrollador de Google.

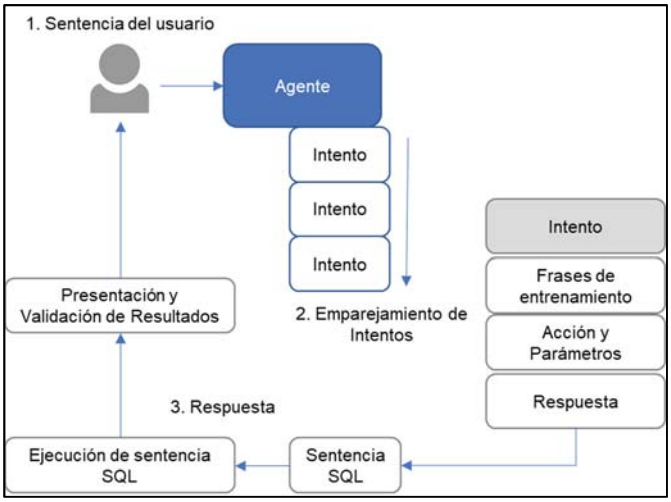

Fig. 4 Implementación de DialogFlow en la ILNBD; Fuente: [19]

VI. RESULTADOS

Para la evaluación de la ILNBD se diseñó una base de datos conteniendo cuatro tablas. Es importante destacar que la definición de las tablas y sus campos está en idioma inglés.

A continuación, se muestran las cuatro tablas utilizadas en la base de datos:

1) T_Person: Tabla que incluye los datos personales de cualquier persona, puede ser empleado o estudiante.

2) T_Employee: Define los datos específicos para un empleado, y está relacionada a la tabla T_Person.

3) T_Student: Define los datos específicos para un estudiante, y también está relacionada a la tabla T_Person

4) T_Sex: Incluye los registros masculino o femenino.

En la Fig. 5 se muestra el diagrama de base de datos.

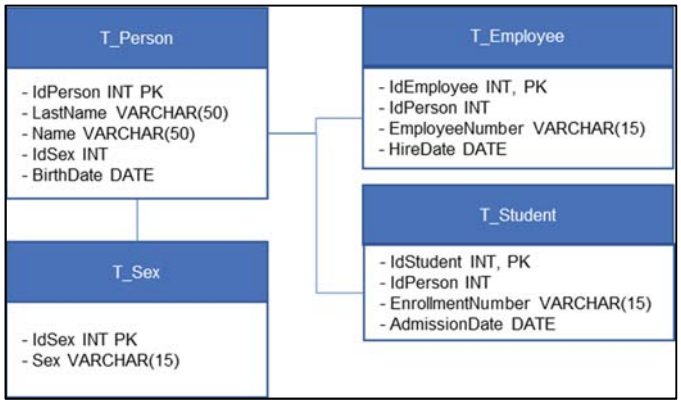

Fig. 5 Diagrama de Base de Datos, Fuente: Propia

Para evaluar la confiabilidad de la interfaz se diseñaron treinta y una preguntas en español, divididas en tres grupos:

Se realizaron dos tipos de pruebas:

1) Incluyendo reconocimiento de voz: Al realizar las 31 preguntas utilizando el micrófono y el lenguaje español solamente 74.19%, 23 de 31 de las pruebas fueron correctas, en los casos erróneos, la API de reconocimiento de voz tuvo algunas deficiencias reconociendo palabras diferentes.

2) Sin reconocimiento de voz, *consultas en texto*: DialogFlow permite introducir la consulta directamente en texto, como se formularía en lenguaje natural, en este caso los resultados fueron del 100%, es decir, las 31 consultas fueron procesadas correctamente.

Las 31 consultas fueron divididas en 3 grupos:

1) Cinco consultas SELECT ALL simples mostradas en la Tabla 1.

2) Quince consultas SELECT simple empleando variaciones de sinónimos, plural y singular mostradas en la Tabla 2.

3) Once consultas SELECT COUNT(ALL) mostrados en la Tabla 3.

TABLA 1 SELECT ALL SIMPLE

Consulta en Lenguaje Natural	**Consulta generada en SQL**
¿Cuáles son los nombres de las tablas en la base de datos?	SELECT * FROM sys.tables
Muestra todas las tablas	
Dame todas las personas	SELECT * FROM T_Person
Regresa todos los empleados	SELECT * FROM T_Employee
Muestra todos los estudiantes	SELECT * FROM T_Student

TABLA 2 SELECT ALL SIMPLE UTILIZANDO SINÓNIMOS Y LAS VARIACIONES EN SINGULAR Y PLURAL

Consulta en Lenguaje Natural	**Consulta generada en SQL**
Persona / Personas	SELECT * FROM T_Person
Gente	
Individuo / Individuos	
Empleado / Empleados	SELECT * FROM T_Employee
Colaborador / Colaboradores	
Trabajador / Trabajadores	
Estudiante / Estudiantes	SELECT * FROM T_Student
Alumno / Alumnos	

TABLA 3 SELECT COUNT(ALL)

Consulta en Lenguaje Natural	**Consulta generada en SQL**
¿Cuántas tablas tiene la base de datos?	SELECT COUNT(*) FROM sys.tables
¿Cuántos registros tiene la tabla de Personas / Gente / Individuos?	SELECT COUNT(*) FROM T_Person
¿Cuántos registros tiene la tabla de Empleados / Colaboradores / Trabajadores?	SELECT COUNT(*) FROM T_Employee
¿Cuántos registros tiene la tabla de Estudiantes / Alumnos?	SELECT COUNT(*) FROM T_Student
¿Cuántas Personas hay? / ¿Cuánta gente hay?	SELECT COUNT(*) FROM T_Person

VII. CONCLUSIONES

La API de reconocimiento de voz y conversión a texto DialogFlow ha mostrado resultados muy interesantes y promisorios para adaptarla a una Interfaz de Lenguaje Natural a Base de Datos, aunque al momento solo se han explorado tres escenarios con consultas simples. Los resultados obtenidos al momento permiten indicar que con esta API se podrá generar una ILNBD integral que habilitará a los usuarios sin experiencia en SQL a poder recuperar información de bases de datos relacionales. A los usuarios con experiencia les permitirá generar consultas en tiempos más cortos.

El aprendizaje automatizado de DialogFlow a través de entrenamiento permite que la interfaz sea bastante flexible respecto al dominio de la base de datos, ya que el mantenimiento del diccionario es muy ágil respecto a los nombres de las tablas y los campos, aprendiendo todas las variaciones que puedan tener los nombres de los objetos, por ejemplo, sinónimos,

978-1-7281-5555-5/19/$31.00 ©2019 IEEE

plurales, singulares, e incluso en el modo texto, aprender y corregir palabras escritas parcialmente correctas.

Es necesario seguir trabajando con la interfaz para integrar consultas que requieran uniones entre dos o más tablas y todas las combinaciones que se puedan tener entre ellas, tales como INNER, FULL, RIGHT, LEFT y CROSS. Asimismo, es necesario probar sentencias que contengan requerimientos de agrupación con las sentencias GROUP BY, HAVING y operaciones de grupo.

Por último, como parte del trabajo a futuro, es necesario explorar el asistente de voz de Google Assistant, para validar que el reconocimiento de voz mejore, sobre todo en pruebas realizadas directamente sobre dispositivos móviles, ya que al momento las consultas con reconocimiento de voz y DialogFlow no ofrecieron los resultados esperados. En algunos casos la conversión de voz a texto interpretó palabras distintas a las que el usuario había indicado verbalmente.

Referencias Bibliográficas

[1] R. Kumar and M. Dua, "Translating controlled natural language query into SQL query using pattern matching technique," in 2014 International Conference for Convergence of Technology, I2CT 2014, 2014.

[2] R. Posevkin and I. Bessmertny, "Translation of natural language queries to structured data sources," in 9th International Conference on Application of Information and Communication Technologies, AICT 2015 - Proceedings, 2015.

[3] A. Mohite and V. Bhojane, "Natural language interface to database using modified co-occurrence matrix technique," in 2015 International Conference on Pervasive Computing: Advance Communication Technology and Application for Society, ICPC 2015, 2015.

[4] M. M. del Rio and S. J. L. Castillo, "Implementation of prototype of natural language interface to database," 2015.

[5] H. Bais, M. Machkour, and L. Koutti, "An independent-domain natural language interface for relational database: Case Arabic language," in Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, 2017.

[6] V. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning," 2017.

[7] S. Bird, E. Klein, and E. Loper, "Language Processing and Python," Computing, 2009.

[8] P. Koehn, "Enabling Monolingual Translators : Post-Editing vs . Options," Hum. Lang. Technol. 2010 Annu. Conf. North Am. Chapter ACL, 2010.

[9] P. K. Ghosh, "Automatic SQL Query Formation from Natural Language Query," iInternational J. Comput. Appl., 2014.

[10] H. R. Tennant, K. M. Ross, and C. W. Thompson, "Usable natural language interfaces through menu-based natural language understanding," 2003.

[11] E. U. Reshma and P. C. Remya, "A review of different approaches in natural language interfaces to databases," in Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2017, 2018.

[12] G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum, "Developing a natural language interface to complex data," ACM Trans. Database Syst., 1978.

[13] N. T. Dang, D. Thi, and T. Tuyen, "Document retrieval based on question answering system," in 2009 2nd International Conference on Information and Computing Science, ICIC 2009, 2009.

[14] Y. Li, H. Yang, and H. V. Jagadish, "NaLIX: an interactive natural language interface for querying XML," SIGMOD, 2005.

[15] Y. W. Wong and R. J. Mooney, "Learning for semantic parsing with statistical machine translation," in Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics -, 2006.

[16] A. Andics, J. M. McQueen, K. M. Petersson, V. Gál, G. Rudas, and Z. Vidnyánszky, "Neural mechanisms for voice recognition," Neuroimage, 2010.

[17] Google, "Agents overview," 2019. [Online]. Available: https://dialogflow.com/docs/intro/agents?authuser=1.

[18] Google, "How does fulfillment work?," 2019. [Online]. Available: https://dialogflow.com/docs/intro/fulfillment?authuser=1.

[19] Google, "Overview," 2019. [Online]. Available: https://dialogflow.com/docs/intro?authuser=1.

978-1-7281-5555-5/19/$31.00 ©2019 IEEE