Wordnet Semantic Relations in a Chatbot

Josephine Petralba
University of San Jose-Recoletos
https://orcid.org/0000-0002-3943-8793
jepetralba@gmail.com

Abstract

The goal of this research is to design and implement a chatbot for querying Wordnet semantic relations. The study creates a contextual chatbot named WordnetBot, a web application that utilizes the use of technologies such as Dialogflow, React, NodeJS, Javascript, and MariaDB. The Wordnet database which leverages all other dictionaries due to its semantic relations representation was used as the data source. Phrase Structure Analysis extracts the keyword and the semantic relation from a user's message or query. It complements the Machine Learning and AI capabilities of Dialogflow in the analysis. The researcher designed an architectural framework for the integration of the different components of WordnetBot.

Keywords: WordNet, Dialogflow, chatbot, semantic relation, phrase structure

1.0 Introduction

Chatbots will open as a new channel for businesses to reach a broader audience in the future. This technology can permanently change the way humans interact with the digital world.

WordNet is a rich resource of dictionary content where the components are connected by semantic relations. WordNet's structure makes it a useful tool for Natural Language Processing (NLP).

There are several online dictionaries available. A few of those which stand out are the Google Dictionary, Dictionary.com, Wiktionary, and Wordnet. Most of the dictionaries have that look-and-feel of a common web interface but are not in a chatbot interface. One can enter a word in a text box and the application gives the dictionary elements such as word meaning, part of speech, among others. One can further engage with the interaction of these websites by drilling down through the expandable buttons, clickable tabs, and available menus.

A chatbot dictionary engages interactively with the user through a more conversational manner typed in a chatbox rather than through click actions. One can narrow down his/her search through a typed (or spoken) conversation. For example, if he/she is interested only in the antonym of the word "titanic", he/she will just type the question on a chatbox: "What is the antonym of titanic." He/she can query semantic relations from a Wordnet dictionary but not through a chatbot interface.

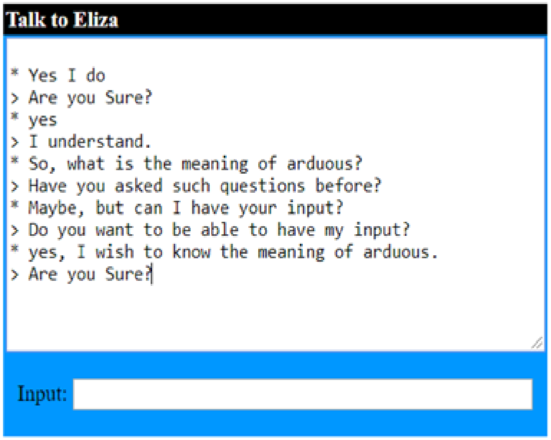
Review of Related Literature

Many studies have shown that knowledge of vocabulary is a crucial predictor of academic success. Chou (2011) states that students fail the English reading test simply because they do not know enough words in order to understand texts and answer comprehension questions. Technology opens many new possibilities for language learning. Chatbots are potentially valuable tools for providing an environment for vocabulary learning. A chatbot is a conversational software agent, that interacts with users using natural language (AbuShawar & Atwell, 2015).

Conversational agents have been used for decades to facilitate learning (Datta, et al., 2020). They can be used as tutorbots (De Pietro & Frontera, 2005) and as discussion-bots (Feng, et al., 2006). Though chatbots at present have their uses, they have not yet been designed from the ground up as a language teacher or even as an explicit language learning tool (Fryer & Carpenter, 2006).

The human language is complex (Kretzschmar, 2015). No chatbot is designed to conform to all use cases. It is difficult to find a chatbot that could answer everything. Most often, a chatbot design is for a particular domain. Eliza, one of the best-known Chatbot developed by Joseph Weizenbaum at Massachusetts Institute of Technology (MIT), cannot answer specific questions related to vocabulary learning. Figure 1 is a sample chat with Eliza. Most businesses use chatbots to answer frequently asked questions (FAQs). For example, when one wants to order pizza, he/she can just use a menu-driven chatbot that utilizes keywords to determine how to serve an appropriate response to the user.

The most sophisticated type of chatbot is one that utilizes Machine Learning (ML) and Artificial Intelligence (AI) to remember conversations with specific users. It is likely that the chatbot's measure for providing a quality user experience rises as its technical complexity increases and how much data it has for the user.



```
* Yes I do
> Are you Sure?
* yes
> I understand.
* So, what is the meaning of arduous?
> Have you asked such questions before?
* Maybe, but can I have your input?
> Do you want to be able to have my input?
* yes, I wish to know the meaning of arduous.
> Are you Sure?
```

Figure 1. *Conversation with Eliza*

Businesses and the academe are investing more in Natural Language Processing (NLP) and Natural Language Understanding (NLU) techniques in their pursuit of creating a chatbot that would be able to interact with humans in a human-like manner. NLU is an integral part of natural processing, the part that is responsible for a human-like understanding of the meaning rendered by a specific text. The most significant difference of NLU from NLP is that NLU goes beyond understanding words as it tries to interpret meaning that deals with usual human errors like mispronunciations or transposed letters or words (Sciforce, 2019).

Miller & Fellbaum (1992) suggested that semantic networks like WordNet would be valuable aids in teaching second language learners, the semantic structures of an unfamiliar language. Wordnet, an online lexical database for English, is a rich resource for mastering semantic relations. Wordnet data and its related projects can be found at Princeton University (2010) with the link https://wordnet.princeton.edu/related-projects.

This research included a study on stemming. There are many stemming algorithms developed. Some of the categories of the stemming algorithms are lookup algorithms, the production technique, suffix-stripping algorithms, lemmatization algorithms, stochastic algorithms, affix stemmers, matching algorithms, to name a few. There are also several stemming libraries available for download. For example, the desktop installation of Wordnet made its source available in its programs folder, including a stemmer implementation written in C language.

Project Objectives

A. General Objectives

The main objective of this research is to design and implement a Chatbot for querying Wordnet semantic relations.

B. Specific Objectives

There are several elements to consider in creating a chatbot. This study has the following specific goals:

1. to preprocess WordNet database content for chatbot knowledge;
2. to provide the backend application with a more straightforward interface, out of the knotty Wordnet database;
3. to do a Phrase Structure Analysis and use Machine Learning and Artificial Intelligence to develop a contextual chatbot; and
4. to design and Implement an Architectural Framework for the integration of the different components of the chatbot.

2.0 Project Methodology

To develop the app, the proponent needs to:

1. download and process the data from Wordnetport;
2. create the tables and stored procedures for fetching the data from the Wordnet database;
3. implement a stemmer in SQL stored procedures;
4. do a Phrase Structure Analysis to extract the word and semantic relation from the user query;
5. set-up the chatbot environment and train the chatbot with Wordnet content;
6. develop the chatbot user interface; and
7. integrate the components of the chatbot. These include the frontend application, backend application, NLU engine, and the database.

3.0 Research Data: Wordnet

Wordnet contains only "open-class words." Table 1 presents the count of words in which their Part Of Speech (POS) are nouns, verbs, adjectives, adjective satellites, or adverbs. The WordNet dictionary does not include determiners, prepositions, pronouns, conjunctions, and articles.

Table 1. *Wordnet Part of Speech Count*

POS	POSNAME	COUNT
n	noun	529507
v	verb	124721
a	adjective	33225
r	adverb	1256
s	adjective satellite	46294

Adjective satellite differs from the adjective in that adjective satellite is restricted to a noun. For example, "arid" is an adjective satellite to the adjective "dry". The adjective satellite "arid" must associate only with "climate". That is, "arid" only refers to "dry climate", whereas the adjective "dry" can be related to several nouns. The examples are: "dry cow", which means "cow not producing milk" and "dry cough", which means a "kind of cough without a mucous or watery discharge".

Semantic relation is the relationship between the meanings of words or phrases. Table 2 is a list of some of Wordnet's semantic relations.

The Wordnet structure groups the words or phrases into synonym sets (or synsets). The synsets connect to other synsets through semantic relations. The interlinking relationship can be viewed as a directed graph as shown in Figure 2, where a node represents a semantic component such as a word, its glossary, and sample usage. An edge represents a semantic relation.

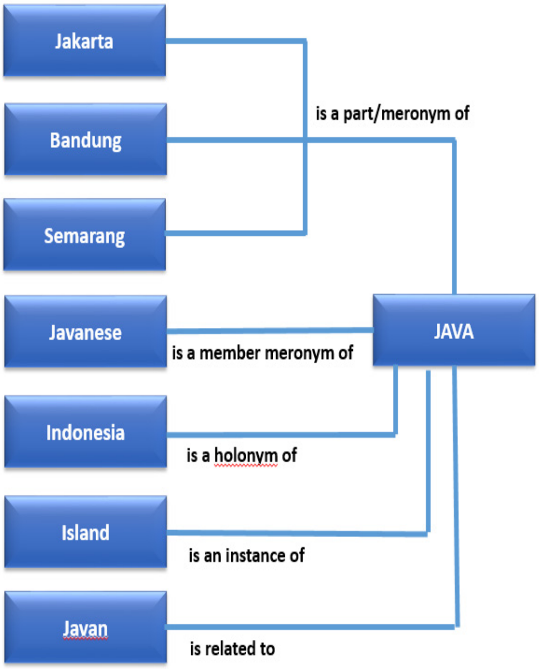


Figure 2. *WordNet Graph Structure*

Table 2. *Wordnet Semantic Relations*

Semantic Relation	Description
Antonym	A concept opposite in meaning to another
Hypernym	A concept whose meaning denotes a superordinate
Hyponym	A concept whose meaning denotes a subordinate
Substance meronym	A concept that is the substance of another idea
Part meronym	A concept that is a part of a another concept
Member meronym	A concept that is a member of another concept
Substance holonym	A concept that has another concept as a substance
Part holonym	A concept that has another concept as a part
Member holonym	A concept that has another concept as a member
Attribute	An adjective that is the value of a noun
Cause	A verb that is the cause of a result
Entail	A verb that involves unavoidably a result
Pertainym	An adjective or adverb that relates to a noun
Derivation	The process whereby new words are formed from existing words or bases by affixation
Domain category	The content of a particular field of knowledge

The MySQL SQL RDBMS consists of eighteen tables. Figure 3 (on a separate page) is the Entity-Relationship Diagram (ERD) depicting the Wordnet database structure.

The Relational Database Management System (RDBMS) suitably stores the interlinking relationship of the Wordnet database. Most of the fields in the tables are numbers that are used as primary keys and foreign keys enforcing the relationship in the RDBMS. So to query the Wordnet SQL database involves several SQL joins which is inherently complex due to the knotty property of Wordnet RDBMS. Petralba (2014a) published a paper that created distribution copies which were made available in a collection of SQL scripts out of the Wordnet RDBMS. The proponent downloaded Wordnetport (Petralba, 2014b) from the website http://wordnetport.sourceforge.net/, as data for this research, processed it in MySQL database, and created a table named RELATIONS, and stored procedures. Table 3 shows the fields of the RELATIONS table.

Table 3. *Fields of the RELATIONS Table*

FIELD	TYPE	SAMPLE RECORD
A_WORD	varchar(80)	advantageous
A_POS	char(1)	a
SEMANTIC_RELATION	varchar(50)	antonym
C_RELATED_WORDS	varchar(4000)	disadvantageous, unfavourable, unfavorable
A_SYNONYM_WORDS	varchar(4000)	advantageous, favourable, favorable
A_DEFINITION	varchar(4000)	giving an advantage
A_SAMPLESET	varchar(4000)	a contract advantageous to our country|socially advantageous to entertain often
A_SYNSETID	int(11)	300064738
A_LEXDOMAINNAME	varchar(32)	adj.all
A_SENSENUM	int(11)	1
C_DEFINITION	varchar(4000)	involving or creating circumstances detrimental to success or effectiveness
C_SAMPLESET	varchar(4000)	a disadvantageous outcome|a well known study from the 1970's showed that gender stereotyping placed women in a disadvantageous position|made an unfavorable impression
C_SYNSETID	int(11)	300065808
C_LEXDOMAINNAME	varchar(32)	adj.all
C_LEXDOMAINID	int(11)	0

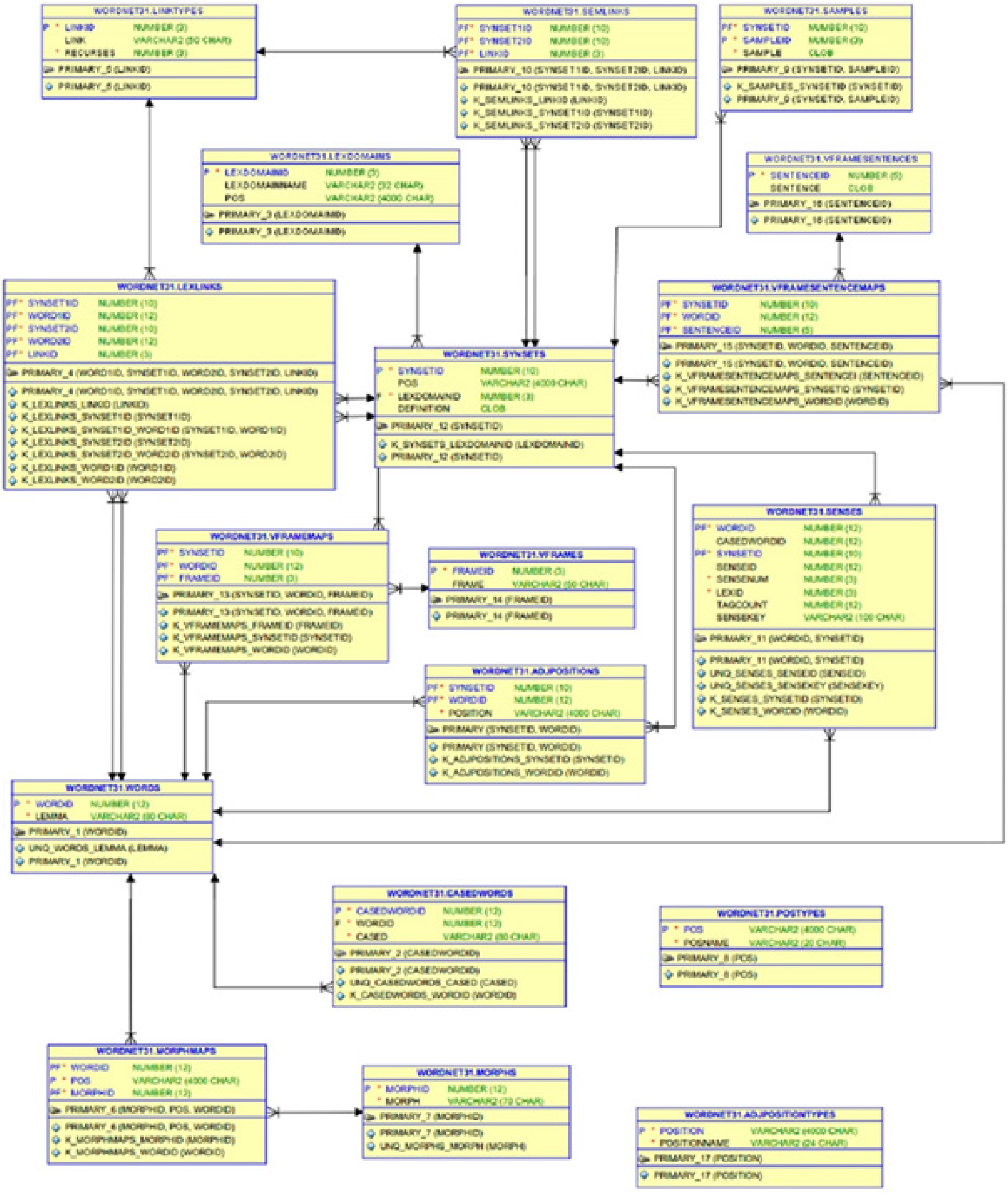

WORDNET31.LINKTYPES
P * LINKID NUMBER (3)
LINK VARCHAR2 (50 CHAR)
* RECURSES NUMBER (3)
PRIMARY_5 (LINKID)
PRIMARY_5 (LINKID)
WORDNET31.SEMLINKS
PF* SYNSET1ID NUMBER (10)
PF* SYNSET2ID NUMBER (10)
PF* LINKID NUMBER (3)
PRIMARY_10 (SYNSET1ID, SYNSET2ID, LINKID)
PRIMARY_10 (SYNSET1ID, SYNSET2ID, LINKID)
K_SEMLINKS_LINKID (LINKID)
K_SEMLINKS_SYNSET1ID (SYNSET1ID)
K_SEMLINKS_SYNSET2ID (SYNSET2ID)
WORDNET31.SAMPLES
PF* SYNSETID NUMBER (10)
P * SAMPLEID NUMBER (3)
* SAMPLE CLOB
PRIMARY_9 (SYNSETID, SAMPLEID)
K_SAMPLES_SYNSETID (SYNSETID)
PRIMARY_9 (SYNSETID, SAMPLEID)
WORDNET31.LEXDOMAINS
P * LEXDOMAINID NUMBER (3)
LEXDOMAINNAME VARCHAR2 (32 CHAR)
POS VARCHAR2 (4000 CHAR)
PRIMARY_3 (LEXDOMAINID)
PRIMARY_3 (LEXDOMAINID)
WORDNET31.VFRAMESENTENCES
P * SENTENCEID NUMBER (5)
SENTENCE CLOB
PRIMARY_16 (SENTENCEID)
PRIMARY_16 (SENTENCEID)
WORDNET31.LEXLINKS
PF* SYNSET1ID NUMBER (10)
PF* WORD1ID NUMBER (12)
PF* SYNSET2ID NUMBER (10)
PF* WORD2ID NUMBER (12)
PF* LINKID NUMBER (3)
PRIMARY_4 (WORD1ID, SYNSET1ID, WORD2ID, SYNSET2ID, LINKID)
PRIMARY_4 (WORD1ID, SYNSET1ID, WORD2ID, SYNSET2ID, LINKID)
K_LEXLINKS_LINKID (LINKID)
K_LEXLINKS_SYNSET1ID (SYNSET1ID)
K_LEXLINKS_SYNSET1ID_WORD1ID (SYNSET1ID, WORD1ID)
K_LEXLINKS_SYNSET2ID (SYNSET2ID)
K_LEXLINKS_SYNSET2ID_WORD2ID (SYNSET2ID, WORD2ID)
K_LEXLINKS_WORD1ID (WORD1ID)
K_LEXLINKS_WORD2ID (WORD2ID)
WORDNET31.SYNSETS
P * SYNSETID NUMBER (10)
POS VARCHAR2 (4000 CHAR)
F * LEXDOMAINID NUMBER (3)
DEFINITION CLOB
PRIMARY_12 (SYNSETID)
K_SYNSETS_LEXDOMAINID (LEXDOMAINID)
PRIMARY_12 (SYNSETID)
WORDNET31.VFRAMESENTENCEMAPS
PF* SYNSETID NUMBER (10)
PF* WORDID NUMBER (12)
PF* SENTENCEID NUMBER (5)
PRIMARY_15 (SYNSETID, WORDID, SENTENCEID)
PRIMARY_15 (SYNSETID, WORDID, SENTENCEID)
K_VFRAMESENTENCEMAPS_SENTENCEI (SENTENCEID)
K_VFRAMESENTENCEMAPS_SYNSETID (SYNSETID)
K_VFRAMESENTENCEMAPS_WORDID (WORDID)
WORDNET31.SENSES
PF* WORDID NUMBER (12)
CASEDWORDID NUMBER (12)
PF* SYNSETID NUMBER (10)
SENSEID NUMBER (12)
* SENSENUM NUMBER (3)
* LEXID NUMBER (3)
TAGCOUNT NUMBER (12)
SENSEKEY VARCHAR2 (100 CHAR)
PRIMARY_11 (WORDID, SYNSETID)
PRIMARY_11 (WORDID, SYNSETID)
UNQ_SENSES_SENSEID (SENSEID)
UNQ_SENSES_SENSEKEY (SENSEKEY)
K_SENSES_SYNSETID (SYNSETID)
K_SENSES_WORDID (WORDID)
WORDNET31.VFRAMEMAPS
PF* SYNSETID NUMBER (10)
PF* WORDID NUMBER (12)
PF* FRAMEID NUMBER (3)
PRIMARY_13 (SYNSETID, WORDID, FRAMEID)
PRIMARY_13 (SYNSETID, WORDID, FRAMEID)
K_VFRAMEMAPS_FRAMEID (FRAMEID)
K_VFRAMEMAPS_SYNSETID (SYNSETID)
K_VFRAMEMAPS_WORDID (WORDID)
WORDNET31.VFRAMES
P * FRAMEID NUMBER (3)
FRAME VARCHAR2 (50 CHAR)
PRIMARY_14 (FRAMEID)
PRIMARY_14 (FRAMEID)
WORDNET31.ADJPOSITIONS
PF* SYNSETID NUMBER (10)
PF* WORDID NUMBER (12)
* POSITION VARCHAR2 (4000 CHAR)
PRIMARY (SYNSETID, WORDID)
PRIMARY (SYNSETID, WORDID)
K_ADJPOSITIONS_SYNSETID (SYNSETID)
K_ADJPOSITIONS_WORDID (WORDID)
WORDNET31.WORDS
P * WORDID NUMBER (12)
* LEMMA VARCHAR2 (80 CHAR)
PRIMARY_1 (WORDID)
UNQ_WORDS_LEMMA (LEMMA)
PRIMARY_1 (WORDID)
WORDNET31.CASEDWORDS
P * CASEDWORDID NUMBER (12)
F * WORDID NUMBER (12)
* CASED VARCHAR2 (80 CHAR)
PRIMARY_2 (CASEDWORDID)
PRIMARY_2 (CASEDWORDID)
UNQ_CASEDWORDS_CASED (CASED)
K_CASEDWORDS_WORDID (WORDID)
WORDNET31.POSTYPES
P * POS VARCHAR2 (4000 CHAR)
* POSNAME VARCHAR2 (20 CHAR)
PRIMARY_8 (POS)
PRIMARY_8 (POS)
WORDNET31.MORPHMAPS
PF* WORDID NUMBER (12)
P * POS VARCHAR2 (4000 CHAR)
PF* MORPHID NUMBER (12)
PRIMARY_6 (MORPHID, POS, WORDID)
PRIMARY_6 (MORPHID, POS, WORDID)
K_MORPHMAPS_MORPHID (MORPHID)
K_MORPHMAPS_WORDID (WORDID)
WORDNET31.MORPHS
P * MORPHID NUMBER (12)
* MORPH VARCHAR2 (70 CHAR)
PRIMARY_7 (MORPHID)
PRIMARY_7 (MORPHID)
UNQ_MORPHS_MORPH (MORPH)
WORDNET31.ADJPOSITIONTYPES
P * POSITION VARCHAR2 (4000 CHAR)
* POSITIONNAME VARCHAR2 (24 CHAR)
PRIMARY_17 (POSITION)
PRIMARY_17 (POSITION)

Figure 3. *Wordnet Database Entity Relationship Diagram*

The proponent created stored procedures that serve as the interface between the database and the backend application. Table 4 is the list of the stored procedures created.

Table 4. *Stored Procedure and Sample Call*

Procedure Name and Sample Call
get_rec_for_word_pos_def_sentance input: word output: all records example: call chatbot.get_rec_for_word_pos_def_sentence(‘good’);
get_rec_for_word_relation input: word and semantic output: all records example: call chatbot.get_rec_for_word_relation(‘good’,’antonym’);
get_rec_for_word_relation_pas input: word.semantic relations and pos output: all records example: call chatbot.get_rec_for_word_relation_pos(‘good’,’antonym’,’n’);
get_1rec_for_word_relation_pos input: word, semantic relations and pos output: one record example: call chatbot.get_1rec_for_word_relation_pos(‘good’,’antonym’,’n’);
get_synonym_for_word input: word output: all synonyms example: call chatbot.get_synonym_for_word(‘good’);
get_all_semantic_relation input: none output: all semantic relations example: call chatbot.get_all_semantic_relation();

Stemming is the process of reducing derived words to their stem, base, or root form.

Examples:

Stemming the word “quizzes” will obtain its base form “quiz”.

Stemming the word “men” will obtain its base form “man”.

Stemming the word “wolves” will obtain its base form “wolf”.

In creating a dictionary search, stemming the search words is vital. One has to use the base form to locate the entry in the dictionary database. Comparative studies on the different stemming libraries available were tried out. Consequently, these libraries were found out to be inefficient because if a library code is updated he/she has to recompile the program into a new executable. The development direction was then bent on integrating the stemming process inside the database. Wordnet SQL database has a table named MORPHOLOGY, which consists mostly of irregular word inflections. Table 5 is the list of the first few records on the morphology table.

Table 5. *Few records in MORPHOLOGY Table*

Word	POS	Morph
aardwolf	n	aardwolves
abacus	n	abaci
abet	v	abetted
abet	v	abetting
abhor	v	abhorred
abhor	v	abhorring
abide	v	abode

MORPHOLOGY table has 4996 rows. Some “Stemming Algorithms” process any search key if it is not found in the RELATIONS and MORPHOLOGY table. Table 6 summarizes the rules for stemming.

The first row means that for derived words ending with “s”, a root word is obtained by replacing “s” with an empty string”. For example, the root word of “girls” is “girl”.

Table 6. *Rules for Stemming*

Rule	Sense Type	Suffix to Remove	Suffix to Replace
1	NOUN	“s”	“”
2	VERB	“s”	“”
3	VERB	“ed”	“e”
4	VERB	“ed”	“”
5	ADJ	“er”	“”
6	ADJ	“er”	“e”
7	VERB	“es”	“e”
8	VERB	“es”	“”
9	ADJ	“est”	“”
10	ADJ	“est”	“e”
11	NOUN	“ies”	“y”
12	VERB	“ies”	“y”
13	VERB	“ing”	“e”
14	VERB	“ing”	“”
15	NOUN	“men”	“man”
16	NOUN	“ses”	“s”
17	NOUN	“xes”	“x”
18	NOUN	“zes”	“z”
19	NOUN	“ches”	“ch”
20	NOUN	“shes”	“sh”

4.0 Phrase Structure

The intent is the user's intention on what he/she tries to communicate with the chatbot. For example, if a user types "Give me the hypernym of ‘horse’", his/her intent is to get the hypernym of "horse" from the Wordnet database. Entities are data that one wants to extract from an input string. For example, in an input string, "Give me the hypernym of ‘horse’", the entities that he/she wants to be extracted are "hypernym" and "horse". It is the Dialogflow that does the analysis of the phrase structure of the input string to extract the entities.

Prepositions are words that show the

relationship between a noun or pronoun and some other words. As in “of” in the phrase “of a horse”, or “to” in the phrase “to operate”. In processing the input string, the prepositions usually serve as the delimiter between entities. For example, for the string “hypernym of horse”, the preposition “of” serves as a delimiter between two entities “hypernym” and “horse”. As compared with a verb, noun, or other parts of speech, there are only about 150 prepositions in the English language. To name a few these are: “of”, “to”, “in”, among others.

The Dialogflow’s intent feature groups the phrases with similar structures as intents. For example, the following phrases have a similar structure:

"What is the meaning of conclusion."
"Give me the antonym of streak."
"Hypernym of horse."

It can be observed in these examples that the keyword extracted comes after a preposition "of". These are “conclusion”, “streak”, and “horse”. The semantic relation usually precedes the preposition “of”. The semantic relations extracted are "meaning", "antonym", and “hypernym". A rule for the group of phrases with this phrase structure is then set. Another example of phrases with a similar structure is the following:

"Can you use paradigm in a sample sentence"
"Pencil in a sentence."

In these phrases, the word comes before the pattern.

“in a sentence”
“in a sample sentence”

A set of rules groups phrases with similar structure. In doing so, one will be able to identify the entities to be extracted from a phrase. For example, in English, "pretty island" is a valid adjective phrase because the adjective which is "pretty" should precede the noun, which is "island" It is awkward to say "island pretty".

The Context-Free Grammar (CFG) describes how a user forms strings according to the defined rules. The Backus-Naur form (BNF) is a formal notation for encoding a CFG. The BNF notation for creating a valid adjective phrase in English would be as follows:

```
<adjective phrase>
      ::= <more adjectives><noun><more
      adjectives>
      ::=  <more  adjectives><adjective>|
      <adjective>
```

The phrase "pretty island" is matched to the rule as follows:

```
<adjective phrase>
      ::= <more adjectives><noun>
      ::= <adjective><noun>
      ::= "pretty"<noun>
      ::= "pretty" "island"
```

A Parse Tree is a visual tool for representing how the string matches the rule. Figure 4 is the parse tree for the phrase "pretty island".

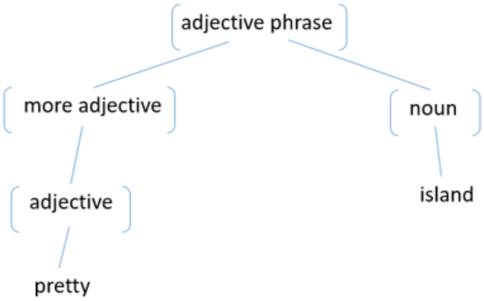


Figure 4. *Parse Tree for “pretty island”*

The leaves are the strings or the end node in each branch of the tree. In the figure, the leaf nodes are “pretty” and “island”. This is the set of strings that are formed.

The Terminal symbols consist of the set of the leaf node strings. In this example, the Terminal symbols are the letters in the English alphabet. The inner nodes in the tree are called the nonterminals or the variables of the CFG.

In the example, there are four variables namely: <adjective phrase>, <more adjectives>, <noun> and <adjective>. The phrase "pretty cute island" is matched with the rule as follows:

<adjective phrase>
::= <more adjectives><noun>
::=<more adjectives><adjective>
<noun>
::=<adjective><adjective><noun>
::= "pretty"<adjective><noun>
::= "pretty" "cute" <noun>
::= "pretty" "cute" "island"

Figure 5 is its parse tree.

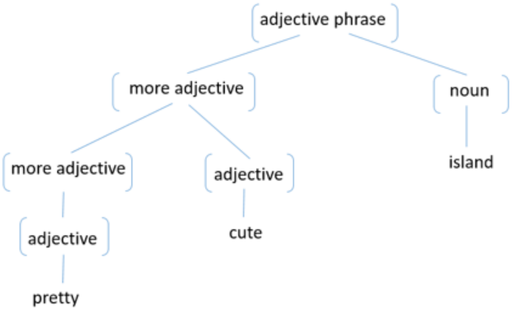


Figure 5. *Parse Tree for “pretty cute island”*

The CFG for forming a valid adjective phrase as described in this example does not apply to the Spanish language. A speaker doesn't say "Bonita Isla" in Spanish. Throughout this paper, the BNF notation represents the rules for grouping phrases of similar structure.

The rules for variables <Other Terminals>, <Relation>, and <Preposition> are the following:

<Other Terminals> ::=
{zero or more letters, digits, or special symbols}

<Relation> ::= also | antonym | attribute | cause | derivation | domain category | domain member category | domain member region | domain member usage | domain region | domain usage | entail | hypernym | hyponym | instance hypernym | instance hyponym | member holonym | member meronym | part holonym | part meronym | participle | pertainym | similar | substance holonym | substance meronym | verb group

<Preposition> ::= “to” | “of” | “for”
<Word> ::=
{word to be searched in Wordnet database}

This rule-based method expressed as CFG in BNF-notation is used to formulate groupings of phrases with similar structures. The following are the groupings of phrase structures formulated in this research.

4.1 Name: Relation Preposition Word

Rule:
<Phrase>
::= <Other Terminals><Relation>
<Preposition><Word>

Description: The word comes after the relation and preposition.

Sample phrases:
- o “What word is similar to robust”
- o “The verb group of arduous”
- o “Give me the definition of qualitative”

The phrase "what word is similar to robust" is matched to the rule:

```
<Phrase>
        ::=<Other Terminals><Relation>
        <Preposition><Word>
        ::= "what word is"<Relation>
        <Preposition> <Word>
        ::= "what word is" "similar"
        <Preposition> <Word>
        ::= "what word is" "similar" "to"
        <Word>
        ::= "what word is" "similar" "to"
        "robust"
```

Figure 6 is its parse tree:

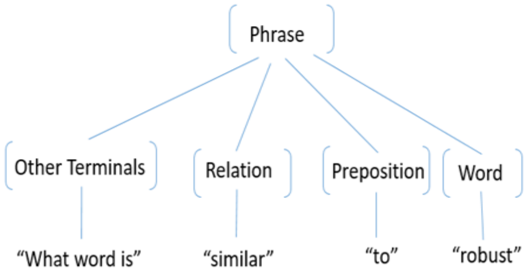


Figure 6. *Parse Tree for "what word is similar to robust"*

4.2 Name: Word in a sentence

Rule:

```
<Phrase>::= <Other Terminals><Word>
                ("in a sentence" | "in a sample
                sentence")
```

Description: The word comes before the phrase "in a sentence" or "in a sample sentence."

Sample phrase:

o Can you use " lean in a sample sentence"

The phrase "Can you use lean in a sample sentence" is matched to the rule:

```
<Phrase>
        ::=<Other Terminals><Word>
            ("in a sentence" | "in a sample sentence")
        ::= "can you use"<Word>
            ("in a sentence" | "in a sample sentence")
        ::= "can you use" "lean"
            ("in a sentence" | "in a sample sentence")
        ::= "can you use" "lean" "in a sample
        sentence"
```

Figure 7 is its parse tree:

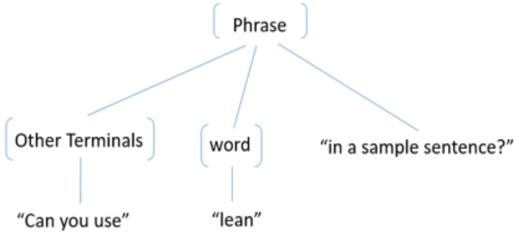


Figure 7. *Parse Tree for "Can you use lean in a sample sentence"*

4.3 Name: Word is also

Rule:

```
<Phrase>::= <Word> ("is" | "are") "also"
```

Description: The word comes before the phrase "is also" or "are also"

Sample phrases:

- o Wolves are also
- o run is also
- o explain is also

The phrase "Wolves are also" is matched the rule:

```
<Phrase>::= <Word> ("is" | "are") "also"
<Phrase>
        ::=<Word> ("is" | "are") "also"
        ::= "wolves" ("is" | "are") "also"
        ::= "wolves" "are" "also"
```

Figure 8 is its parse tree:

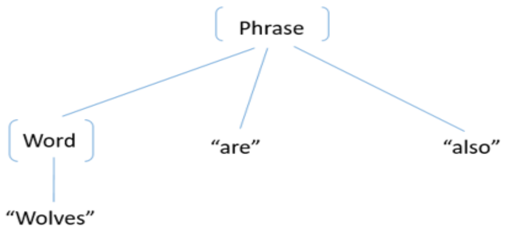


Figure 8. *Parse Tree for “Wolves are also”*

4.4 Name: Word Give Relation

Rule:

<Phrase>::= <Word>(“ ”|“ , ”)"give me"(“its”|
“a”) <Relation>

Description: The analysis locates the word at the beginning of the phrase followed by a verb phrase with the word "give" being the verb and the semantic relation being the direct object.

Sample sentences:

- “Berry, give me its meaning”.
- “Adjacent, give me a sample sentence".
- “Medical, give me its part meronym”.

The sentence “Berry, give me its meaning” is matched the rule:

<Phrase>
::=<Word>(“ ”|“ , ”)"give me"(“its”|“a”) <Relation>
::= "berry"(“ ”|“ , ”)"give me"(“its”|“a”) <Relation>
::= "berry" “ , ”"give me"(“its”|“a”) <Relation>
::= "berry" “ , ” "give me" “its” <Relation>
::= "berry" “ , ” "give me" “its” "meaning"

Figure 9 is its parse tree:

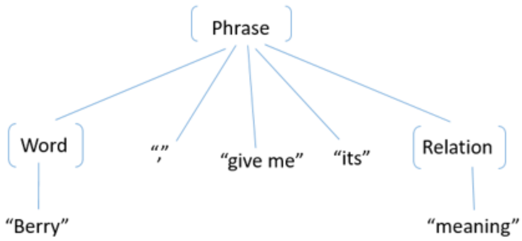


Figure 9. *Parse Tree for “Berry, give me its meaning”*

4.5 Name: Word What Relation

Rule:

<Phrase>
::= <Word>(“ ”|“ , ”)"what"(“is”|“are”) "its"
<Relation>

Description: The analysis locates the word at the beginning of the phrase followed by a direct question introduced by a “what” phrase.

Sample phrases:

- “Hot, what is its antonym.”
- “Species, what is its member holonym.”
- “Play, what are its verb groups.”

4.6 What is/are word

Rule:

<Phrase>::= "what is" <word>

Description: The analysis locates the word after the phrase "what is".

Sample phrases:

- “What is skill?”
- “What is response?”
- “What is regulate?”

4.7 What do you mean by the word

Rule:

<Phrase>::= "what do you mean" <word>

Description: The analysis locates the word after the phrase "what do you mean by"

Sample phrases:

- “What do you mean by venture?”
- “What do you mean by levy?”
- “What do you mean by inherent?”

4.8 Define word

Rule:

<Phrase>::= (“ ”|“how do you”) "define"
<Word>

Description: The analysis locates the word after "define".

Sample phrases:

o "Define insight."

o "How do you define ideology?"

o "Can you define empirical?"

4.9 Relation Subject / Subject Relation

Rule:

<Phrase>::= <Word><Relation> |
<Relation><Word>

Description: The word and its relation are adjacent.

Sample phrases:

o "physics domain category"

o "derivation try"

o "entail run"

5.0 Chatbot Design and Implementation

Using the basic approach and the default configurations of Dialogflow is simple as just tokenizing an input string and tag a tokenized word as an entity. However, relying on this approach will not work because:

- dialogflow has a limit of 30,000 entity entries (Google, 2020), and there are 142,781 distinct words from Wordnet; and
- it may extract unnecessary words. For example, consider the case when a user asks, "Can you give me the meaning of arduous?". The tokenized words "can", "give", "me", "meaning" and "arduous" would be included in the set of words fetched from the WordNet database and may simply be treated as entities. But in context, the user only needs the word "arduous" as it is the word that the user wants the meaning of. This basic system leads to ambiguity and less precision.

With those limitations of the basic approach and with the requirement for a more profound phrase structure analysis to build a more contextual chatbot, the entities are extracted from Dialogflow using its entity detection feature - a machine learning capability of Dialogflow. The research utilizes this machine learning feature of Dialogflow, where data set for a phrase structure were used in the training.

In contrast with a basic chatbot development where the development is only done in the Dialogflow console, the research came up with a system architecture which has four main components: the frontend, backend, Natural Language Understanding (NLU) engine, and database.

The frontend through a web User Interface (UI) gets the input string from the user. The frontend displays the chatbot response or the output string. The backend application serves as the interface of all the other three components.

These three components cannot directly communicate with each other. Dialogflow classifies the input string to its respective intents and extracts the semantic relation from the string. The database stores the Wordnet data.

The frontend application makes use of React, a popular Java Script library. Backend application uses Node.js. This is a server-side Java Script built upon Google's v8 engine and Express, a framework for building backend apps and helps with requests and routing. Dialogflow is an NLU platform that uses machine learning to provide the most useful response in a conversational user interface. The database used is MariaDB, which is an open-source relational database software that provides an SQL interface for accessing data. MariaDB is a fork of Oracle's MySQL. Figure 10 outlines the Architectural design of the system.

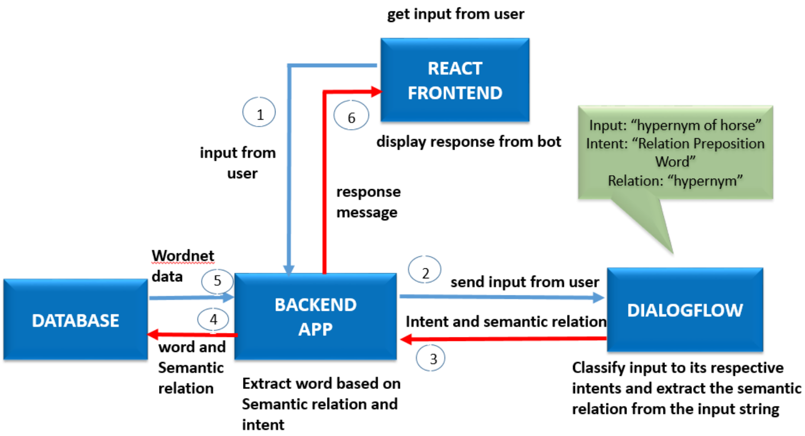

get input from user
REACT FRONTEND
1
6
input from user
display response from bot
Input: “hypernym of horse”
Intent: “Relation Preposition Word”
Relation: “hypernym”
response message
Wordnet data
5
2
send input from user
DATABASE
BACKEND APP
DIALOGFLOW
4
Intent and semantic relation
word and Semantic relation
3
Extract word based on Semantic relation and intent
Classify input to its respective intents and extract the semantic relation from the input string

Figure 10. *Architectural Design*

The user-chatbot conversation has the following steps. Please refer to the labeled arrows in the diagram.

1. The input string or query typed on a web user interface by the user is passed by the frontend to the backend.
2. The backend serves as an interface between the frontend and the Dialogflow. It forwards the input string to the Dialogflow.
3. The Phrase Structure and semantic relation are the results of the Dialogflow processing. This result is passed to the backend.
4. Based on the intent (phrase structure) from the Dialogflow, the backend extracts the target keyword from the input string. Using the target keyword and the semantic relation as parameters invokes the database stored procedure(s) to fetch from the database.
5. Wordnet data is the result set of the processing. The backend receives the result set.
6. The backend processes the result set, and then it returns the output string to the frontend for rendering.

All the communication between contents is done via the HTTP protocol. To show an illustrative example, please refer to Figure 11. The labeled arrows in the diagram show the process. The processing of the input string in each component.

1. Suppose the user enters the query "Give me the hypernym of horse." The backend receives the input string "Give me the hypernym of horse."
2. The backend then sends the input string to Dialogflow.
3. Dialogflow processes the input string and determines the Phrase structure, which is:

   Relation Preposition Word
   The Semantic Relation identified is
   Hypernym.

4. Based on the Phrase Structure, the backend extracts the word. The word obtained by the backend is the word

"horse". It then invokes the Database Stored Procedure to fetch the data from the database given the target keyword and semantic relation as parameters.

5. Message responses from Wordnet data are the result sets returned by the backend.
   Table 7 shows a portion of the result set returned from the database.
6. The frontend then renders the output for the user as shown in Figure 12.

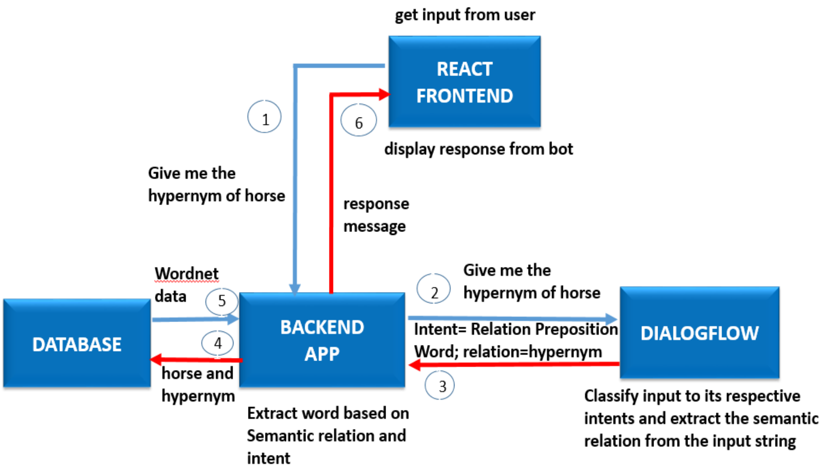
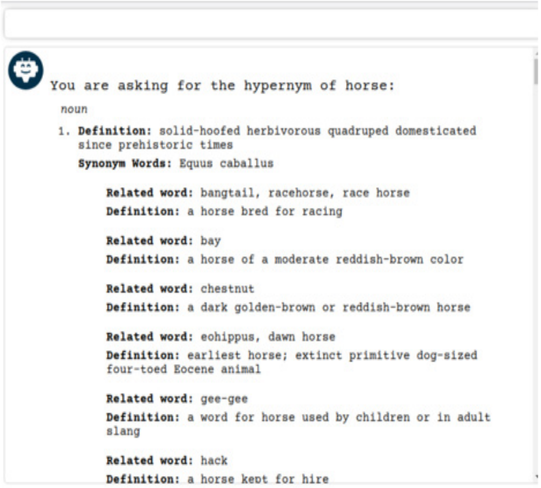

get input from user
REACT FRONTEND
1
6
display response from bot
Give me the hypernym of horse
response message
Wordnet data
5
BACKEND APP
2
Give me the hypernym of horse
Intent= Relation Preposition Word; relation=hypernym
DIALOGFLOW
DATABASE
4
horse and hypernym
3
Extract word based on Semantic relation and intent
Classify input to its respective intents and extract the semantic relation from the input string

Figure 11. *Architectural Design with Example*

Table 7. *Portion of the Database result set*

A_ WORD	A_ POS	SEMANTIC_ RELATION	C_RELATED _WORDS	A_SYNONYM _WORDS	A_DEFINITION	A_ SENSENUM	C_DEFINITION
horse	n	hypernym	bangtail, racehorse, race horse	horse,Equus caballus	solid-hoofed herbivorous quadruped domesticated since prehistoric times	1	a horse bred for racing
horse	n	hypernym	bay	horse,Equus caballus	solid-hoofed herbivorous quadruped domesticated since prehistoric times	1	a horse of a moderate reddish-brown color
horse	n	hypernym	buck, vaulting horse, long horse	gymnastic horse,horse	a padded gymnastic apparatus on legs	2	a gymnastic horse without pommels and with one end elongated; used lengthwise for vaulting
horse	n	hypernym	chestnut	horse,Equus caballus	solid-hoofed herbivorous quadruped domesticated since prehistoric times	1	a dark golden-brown or reddish-brown horse
horse	n	hypernym	eohippus, dawn horse	horse,Equus caballus	solid-hoofed herbivorous quadruped domesticated since prehistoric times	1	earliest horse; extinct primitive dog-sized four-toed Eocene animal

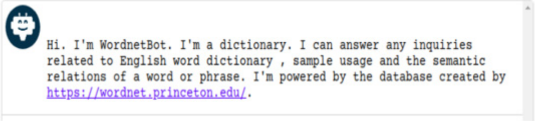


```
You are asking for the hypernym of horse:
 noun
 1. Definition: solid-hoofed herbivorous quadruped domesticated
    since prehistoric times
    Synonym Words: Equus caballus

        Related word: bangtail, racehorse, race horse
        Definition: a horse bred for racing

        Related word: bay
        Definition: a horse of a moderate reddish-brown color

        Related word: chestnut
        Definition: a dark golden-brown or reddish-brown horse

        Related word: eohippus, dawn horse
        Definition: earliest horse; extinct primitive dog-sized
        four-toed Eocene animal

        Related word: gee-gee
        Definition: a word for horse used by children or in adult
        slang

        Related word: hack
        Definition: a horse kept for hire
```

Figure 12. *Portion of returned output*

6.0 Chatbot Conversation with the User

The conversation is initiated by the display of an Introduction Message by WordnetBot, as shown in Figure 13.



```
Hi. I'm WordnetBot. I'm a dictionary. I can answer any inquiries
related to English word dictionary , sample usage and the semantic
relations of a word or phrase. I'm powered by the database created by
https://wordnet.princeton.edu/.
```

Figure 13. *Introduction Message*

The flowchart shown in Figure 14 illustrates the flow of conversation between WordnetBot and the user.

The system trains WordnetBot to reply to the usual greeting message. Among the training phrases entered into the chatbot engine are the following:

Just going to say hi
Heya
Hello hi
Howdy
Hey there
Hi there
Greetings
Hey
Long time no see
Hello
Lovely day isn't it
I greet you
Hello again
Hi
Hello there
A good day
Purpose
What is your name
What you
Who you
What are you
Tell me about you
What is your purpose
Who are you

If a message is identified as a greeting, then the Introduction Message is displayed again as shown in Figure 15.

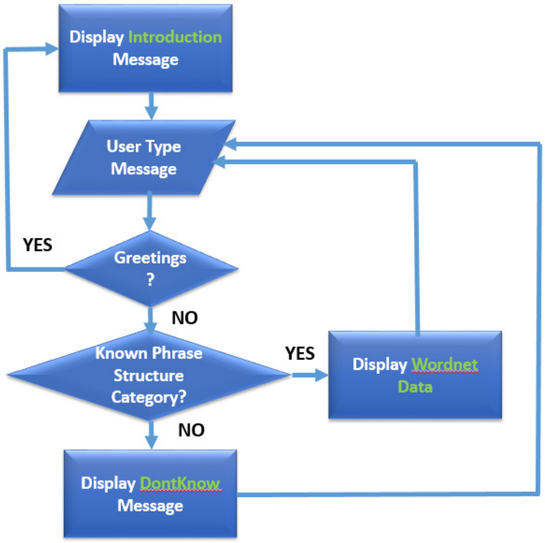
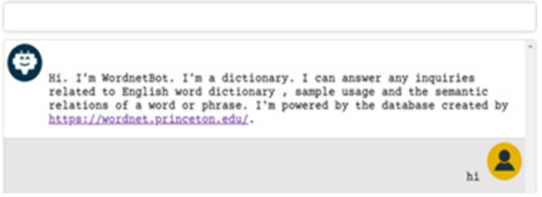


Figure 14. *Flow of Conversation*

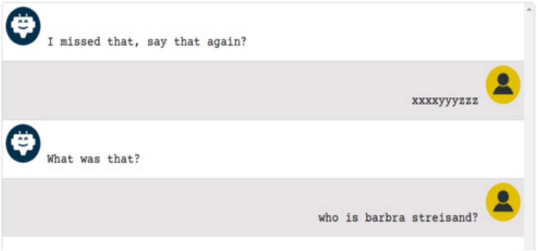


Hi. I'm WordnetBot. I'm a dictionary. I can answer any inquiries related to English word dictionary , sample usage and the semantic relations of a word or phrase. I'm powered by the database created by https://wordnet.princeton.edu/.

hi

Figure 15. *Introduction Message*

WordnetBot is used effectively for a specific purpose. Don't Know Messages are displayed if the message does not fall into any of the Phrase Structures. Shown in Figure 16 is an example of a Don't Know Message.



I missed that, say that again?

xxxxyyyzzz

What was that?

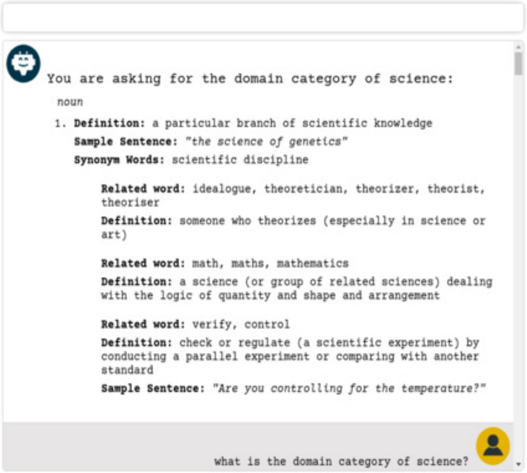
who is barbra streisand?

Figure 16. *Don't Know Message*

A query such as "Who is Barbra Streisand?" is a sensible question, but is not related to Wordnet Semantic Relations. For random messages such as "xxxxxyyyyzzz", WordnetBot will also display the Don't Know Message. The system then randomly picks Don't Know Messages for display. Examples of Don't Know Messages are the following:

I didn't get that. Can you repeat?
I missed what you said. What was that?
Sorry, could you say that again?
Sorry, can you say that again?
Can you say that again?
Sorry, I didn't get that. Can you rephrase it?
Sorry, what was that?
One more time?
What was that?
Say that one more time?
I didn't get that. Can you repeat it?
I missed that, say that again?

Figure 17 shows an example of a message, which is a valid Phrase Structure. WordnetBot will then display the results fetched from the Wordnet database.



You are asking for the domain category of science:

noun

1. **Definition:** a particular branch of scientific knowledge
   Sample Sentence: *"the science of genetics"*
   Synonym Words: scientific discipline

   Related word: idealogue, theoretician, theorizer, theorist, theoriser
   Definition: someone who theorizes (especially in science or art)

   Related word: math, maths, mathematics
   Definition: a science (or group of related sciences) dealing with the logic of quantity and shape and arrangement

   Related word: verify, control
   Definition: check or regulate (a scientific experiment) by conducting a parallel experiment or comparing with another standard
   Sample Sentence: *"Are you controlling for the temperature?"*

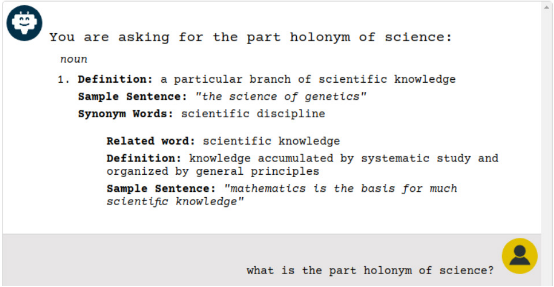
what is the domain category of science?

Figure 17. *A Valid Phrase Structure*

WordnetBot answered the intent of the query which is about the domain category of science.

Figure 18 is another example of a WordnetBot conversation where the intent is to ask the Wordnet database for the part holonym of science. WordnetBot is asked the question:

"What is the part holonym of science?"



You are asking for the part holonym of science:

noun

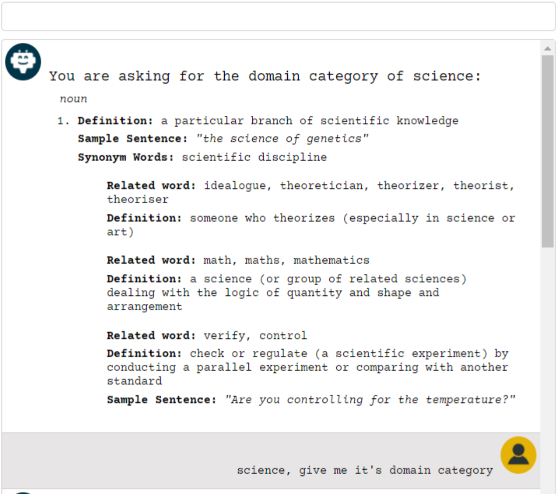
1. **Definition:** a particular branch of scientific knowledge
   Sample Sentence: *"the science of genetics"*
   Synonym Words: scientific discipline

   Related word: scientific knowledge
   Definition: knowledge accumulated by systematic study and organized by general principles
   Sample Sentence: *"mathematics is the basis for much scientific knowledge"*

what is the part holonym of science?

Figure 18. *A WordnetBot Result Query*

The analysis of the phrase structure of the input string to extract the entities is not only done using the rule-based method. For example, in Figure 19, the query "science, give me its domain category", can be parsed using any of the rules discussed in Section 6.0.



You are asking for the domain category of science:

noun

1. **Definition:** a particular branch of scientific knowledge
   Sample Sentence: *"the science of genetics"*
   Synonym Words: scientific discipline

   Related word: idealogue, theoretician, theorizer, theorist, theoriser
   Definition: someone who theorizes (especially in science or art)

   Related word: math, maths, mathematics
   Definition: a science (or group of related sciences) dealing with the logic of quantity and shape and arrangement

   Related word: verify, control
   Definition: check or regulate (a scientific experiment) by conducting a parallel experiment or comparing with another standard
   Sample Sentence: *"Are you controlling for the temperature?"*

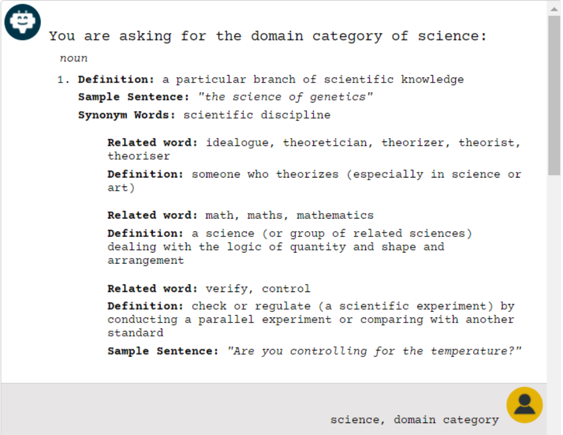
science, give me it's domain category

Figure 19. *Query used is "science, give me its domain category"*

However, in Figure 20, the same answer can still be retrieved with the query "science, domain category". Even if this phrase cannot be parsed according to the rules listed in Section 6.0, the entities can still be extracted due to Dialogflow's Machine Learning capabilities.

Therefore, entities are extracted using both the rule-based and machine learning technique feature of Dialogflow.

The main purpose of WordnetBot is to ask the meaning of a word and its semantic relations. Through the use of WordnetBot, users can gain more understanding of the other semantic relations from the Wordnet database. Once a user has come to the appreciation of the rich linguistic information from Wordnet, then the conversation with WordnetBot will be more enjoyable through a chatbot which is unique in the sense that no chatbot can answer thoroughly Wordnet semantic relations yet.



You are asking for the domain category of science:

noun

1. **Definition:** a particular branch of scientific knowledge
   Sample Sentence: *"the science of genetics"*
   Synonym Words: scientific discipline

   Related word: idealogue, theoretician, theorizer, theorist, theoriser
   Definition: someone who theorizes (especially in science or art)

   Related word: math, maths, mathematics
   Definition: a science (or group of related sciences) dealing with the logic of quantity and shape and arrangement

   Related word: verify, control
   Definition: check or regulate (a scientific experiment) by conducting a parallel experiment or comparing with another standard
   Sample Sentence: *"Are you controlling for the temperature?"*

science, domain category

Figure 20. *Query used is "science, domain category"*

7.0 Conclusion and Future Work

The paper discussed an overview of the Wordnet database. The research project brings forth the architectural design which can be used as a baseline for those who will be doing a contextual chatbot, and when the number of entries mapping to entities exceeds the allowed number of entities in Google's Dialogflow. The chatbot that was created was not a typical Menu/Button-Based or Keyboard Recognition, like most Customer Support chatbots. Rather it is a contextual chatbot which utilizes Machine Learning or Artificial Intelligence techniques. Another result of the research is the design of the database and the stored procedures that are invoked to fetch the data from the database given the word obtained from the backend and the semantic relation as parameters. Integrated within the database system, is the stemming module. As long as the intent of the query is within the domain of Wordnet Semantic Relations, WordnetBot returns expected results as compared with the results from the works similar to Wordnet.

This paper serves as a window to more related studies on Chatbots, Wordnet, and other data sources. Here are some of the recommended projects for future work.

1. Explore and utilize what data from the Wordnetport and other data sources such as dictionaries, taxonomies, and text corpus can be included for a chatbot application. One may create a chatbot for a more specific Domain Knowledge.
2. Improve the stemming algorithm implemented in this research. This involves rigorously testing the stemmer and improve from there.
3. Instead of just utilizing the Machine Learning feature of Dialogflow which was done in this paper, expand its NLU capabilities by invoking other Machine Learning libraries. A more challenging work is to implement some advanced Machine Learning and Artificial Intelligence techniques for better results.
4. Expand the flow of conversation.
5. Provide Speech Recognition capability to the Wordnet Chatbot.

References

AbuShawar, B., & Atwell, E. (2015). ALICE chatbot: trials and outputs. *Computación y Sistemas*, *19*(4), 625-632.

Chou, P. T. M. (2011). The effects of vocabulary knowledge and background knowledge on reading comprehension of Taiwanese EFL students. *Electronic Journal of Foreign Language Teaching, 8*(1).

Datta, D., Phillips, M., Chiu, J., Watson, G. S., Bywater, J. P., Barnes, L., & Brown, D. (2020). Improving classification through weak supervision in context-specific conversational agent development for teacher education. arXiv preprint arXiv:2010.12710.

De Pietro, O., & Frontera, G. (2005). Tutorbot: an application aiml-based for web-learning. *Advanced Technology for Learning, 2*(1), 29-34.

Feng, D., Shaw, E., Kim, J., & Hovy, E. (2006, January). An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of the 11th international conference on Intelligent user interfaces*, 171-177.

Fryer, L., & Carpenter, R. (2006). Bots as language learning tools. Language Learning & Technology, 10(3), 8-14.

Google. (2020). Quotas and limits Dialogflow Google Cloud. https://cloud.google.com/dialogflow/

Kretzschmar Jr, W. A. (2015). *Language and complex systems.* Cambridge University Press.

Miller, G. A., & Fellbaum, C. (1992). WordNet and the organization of lexical memory. In Intelligent tutoring systems for foreign language learning. Springer.

Petralba, J. E. (2014a). An extracted database content from WordNet for natural language processing and word games. *In 2014 International Conference on Asian Language Processing (IALP)*, 199-202.

Petralba. (2014b). WordnetPort. http://wordnetport.sourceforge.net/

Princeton University. (2010). Related Projects. https://wordnet.princeton.edu/related-projects

Sciforce. (2019). NLP vs. NLU: From understanding a language to its processing. https://medium.com/sciforce/nlp-vs-nlu-from-understanding-a-language-to-its-processing-1bf1f62453c1