Cognitive Approach to Natural Language Processing (SD213)

Júlia Togashi de Miranda

June 2021

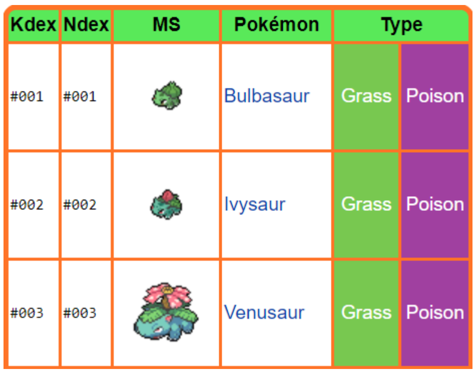
1 Introduction

This is a project in the scope of the class Cognitive Approach to Natural Language Processing (SD213). My objective will be to extend the lab work on Procedural semantics by creating a mini-knowledge base on a specific domain.

The domain I choose will be Pokémon lore. Which is a series of electronic games developed by Game Freak and published by Nintendo, first released in 1996. Each game takes place on an island full of creatures called Pokémon, which have their own characteristics and abilities. Your goal in the game is to capture and battle these creatures, each time obtaining more information about them.

2 The Database

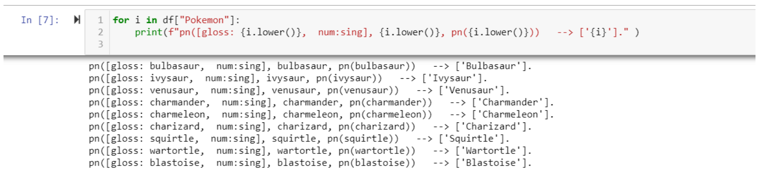
For the database of this project, I choose the specific domain of the Pokémon games. One of the reasons that there is a lot of information on the web about this topic. Also, the data is really modular: I could add more knowledge to the database as the project evolved, so that it was progressively more complex. I decided to add the information of the type of each Pokémon, that is the aspect of the game. For that, I got an Excel sheet that can be found here. I used the first generation of Pokémon, that is 151 different Pokémon with a proper name and each one with one ore two types.



Kdex	Ndex	MS	Pokémon	Type	
#001	#001		Bulbasaur	Grass	Poison
#002	#002		Ivysaur	Grass	Poison
#003	#003		Venusaur	Grass	Poison

Figure 1: Example of the information in the database

After that I used a simple program in python to automatically generate the knowledge and lexicon for my grammar.In the end, most of the changes comprehended extending the previous grammar of the lab so it contains words that are in the context of the database.



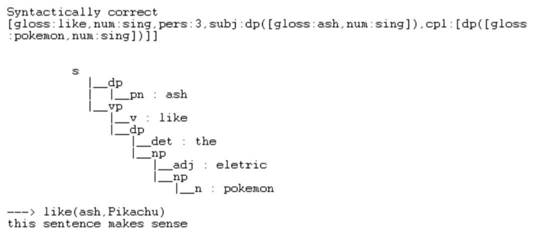
```
In [7]:  for i in df["Pokemon"]:
             print(f"pn([gloss: {i.lower()},  num:sing], {i.lower()}, pn({i.lower()}))   --> ['{i}']." )

pn([gloss: bulbasaur,  num:sing], bulbasaur, pn(bulbasaur))   --> ['Bulbasaur'].
pn([gloss: ivysaur,  num:sing], ivysaur, pn(ivysaur))   --> ['Ivysaur'].
pn([gloss: venusaur,  num:sing], venusaur, pn(venusaur))   --> ['Venusaur'].
pn([gloss: charmander,  num:sing], charmander, pn(charmander))   --> ['Charmander'].
pn([gloss: charmeleon,  num:sing], charmeleon, pn(charmeleon))   --> ['Charmeleon'].
pn([gloss: charizard,  num:sing], charizard, pn(charizard))   --> ['Charizard'].
pn([gloss: squirtle,  num:sing], squirtle, pn(squirtle))   --> ['Squirtle'].
pn([gloss: wartortle,  num:sing], wartortle, pn(wartortle))   --> ['Wartortle'].
pn([gloss: blastoise,  num:sing], blastoise, pn(blastoise))   --> ['Blastoise'].
```

Figure 2: Python script

3 Results

First of all, I checked if after adding the new knowledge, examples from past works would still work. One of these were using the verb “like”, that was already defined.



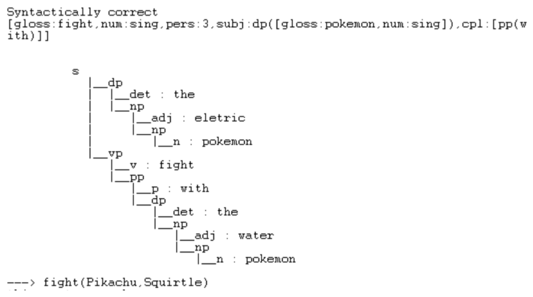
```
Syntactically correct
[gloss:like,num:sing,pers:3,subj:dp([gloss:ash,num:sing]),cpl:[dp([gloss
:pokemon,num:sing])]]

         s
           |__dp
           |    |__pn : ash
           |__vp
                |__v : like
                |__dp
                     |__det : the
                     |__np
                          |__adj : eletric
                          |__np
                               |__n : pokemon

---> like(ash,Pikachu)
this sentence makes sense
```

Figure 3: First tests to check if grammar is working

After that, I moved on to trying the new words added to the grammar. We can see bellow that the program succeeds in linking information related to both the subject and the predicate in the phrase bellow. Another interesting thing is that, as multiple Pokémon have the same type, we can get all the possible combinations.



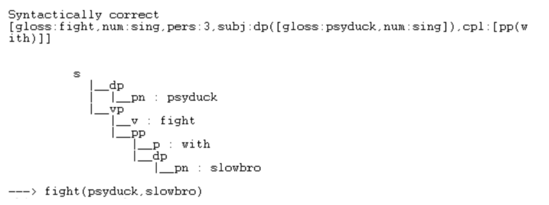
```
Syntactically correct
[gloss:fight,num:sing,pers:3,subj:dp([gloss:pokemon,num:sing]),cpl:[pp(w
ith)]]

         s
           |__dp
           |    |__det : the
           |    |__np
           |         |__adj : eletric
           |         |__np
           |              |__n : pokemon
           |__vp
                |__v : fight
                |__pp
                     |__p : with
                     |__dp
                          |__det : the
                          |__np
                               |__adj : water
                               |__np
                                    |__n : pokemon

---> fight(Pikachu,Squirtle)
```

Figure 4: Test if new lexicon also works with new knowledge

The program is also able to recognize what new phrases are talking about, as seen bellow:



```
Syntactically correct
[gloss:fight,num:sing,pers:3,subj:dp([gloss:psyduck,num:sing]),cpl:[pp(w
ith)]]


          s
            |__dp
            |   |__pn : psyduck
            |__vp
                |__v : fight
                |__pp
                    |__p : with
                    |__dp
                        |__pn : slowbro

---> fight(psyduck,slowbro)
```

Figure 5: Test for new Phrases

4 Future work

One interesting future developments of the work is to expand the database, so it contains extra information such as: the attacks of each Pokémon; the strengths and weakness of each type; the stats of combat, such health points.

That way, the work in the field of Natural Language Processing can be extended to take into account these factors to produce more complex dialog. One interesting approach would be to use CAN (Conflict-Abduction-Negation) to discuss possible scenarios of a Battle. For example:

[Context: You have 3 Pokémon, and a certain Pokémon you should battle. You're trying to decide which one you should choose to maximize your chances of winning]

A: I want to use Pikachu as an electric type is strong against water, but he has such a low health.

B: Then use Golem, he has the strongest health.

A: But he is of type ground and rock, that is doubly week against water.

B: Hum.. Then use Psyduck, he is of type water

A: But attacking a Pokémon of the same type won't give me an attack bonus.

B: But you won't get damage bonus as well…

5 Conclusion

This work was a nice opportunity to look into more details to the previous lab on Procedural semantics. It also proved to be a challenge to make answers in a natural way. For example, in my initial planning, an expected answer for "Ash likes an electric Pokémon" would be: "Yes, that Pokémon is Pikachu". This has proven to be really difficult, specially taking into account the number of phrases and possible responses.

Even if the initial objective wasn't accomplished, I was able to successfully integrate an existing database to the world knowledge of the previous work and successfully inference the meaning of the subjects and predicates in a sentence.