

1. Introduction.

At the school level, students learn basic concepts of many different subjects to develop a better understanding of the issues and problems around them. However, not all subjects are easy to learn and understand. Many students hate chemistry because it has a lot of information to memorize. One of the first things that students have to learn in chemistry is the periodic table, which is categorized and organized to provide the similarities and properties of the elements. Nevertheless, it requires a lot of time and effort to understand and remember the periodic table. Therefore, for this project, I want to create a chatbot that can answer questions about the periodic table of elements. With the chatbot, they can conveniently recall the information of elements and immediately get an answer to their questions. As a chatbot is like a friend who is always ready to help them, it can make learning not only be more enjoyable but also be more effective.

Part 3 update: No changes.

Part 4 update: No changes.

2. Technical Overview.

My chatbot will be implemented in Python 3. I have not planned to use any APIs and external libraries.

Part 3 update: No changes.

Part 4 update: Run on terminal: `python3 Proj.py`

3. Phases.

Phase 1: I will reuse part 1 code, create grammar rules and lexicons for simple questions about the information of the periodic table and main-group elements, such as molar mass, position, and symbol. After phase 1, users can make common greeting and goodbye sentences, and ask questions like “What is the periodic table?”, “How many columns are in the periodic table?” or “What is the symbol of hydrogen?” However, in this phase, users and bot will interact in the python console (take input from the console and print output to the console). I will implement phase 1 for part 3.

Phase 2: I will expand the grammar rules and lexicons, so the bot can answer more complex questions and information about transition elements. For example, the chatbot can reply to comparison questions, yes/no questions, and properties of elements (metal or nonmetal, gas or solid at room temperature). Moreover, in this phase, I will implement a user interface which is a simple chatbox. Hence, it would be more convenient to interact with my bot. Finally, phase 2 will be implemented for part 4.

Part 3 update: Extend chatbot abilities.

The bot can answer information about position, symbol, mass of all elements on the periodic table. (not just the main-group elements)

The chatbot can respond to unexpected replies/ inputs.

It can handle some variety of input questions and have from 2 -5 different ways to respond. For example: Input: [Hi/Hello] + My name is/ I am + [name]

Output: [Hi/Hello] + name or [Hi there! / Hi ya!]

Part 4 update:

Can apply for all 118 elements/symbols

Remember the element and information from the latest question, so my chatbot can respond to “its/ it” and “what about/ how about” questions. For example: How many electrons does K have? And if in the next question, the user asks “How about/ what about Na?” -> reply the same as above question.

Respond to some customer reply, such as “I see”, “ got it”, “ thank you” and “thanks”, ‘yes’

Expand how many questions, the bot can count the number of electrons, protons of an element.

For Comparison questions, my chat bot can now compare (more or less) - reactive, metallic and nonmetallic of 2 or more elements.

For Y/N questions, it can check the symbol or number of electrons and protons, atomic number, molar mass of an element. It also can verify the comparison of reactivity, metal and nonmetal of 2 elements. For example: “Is Cl more reactive than F?”

Now, it also can answer questions about electrons, protons, atomic number and who discovered the elements.

Also have 2-5 different ways to respond for each question.

However, I do not implement the user interface as I mentioned before. As I see the main points of this project is the way my chat bot responds to users. Therefore, instead of implementing user interface, I add other features for my bot. (remember topic under discussion and response when user asks, variety of grammatical form)

4. Examples.

Example for part 3 of project:

User: Hi bot my name is Loc

Bot: Hello, How can I help you, Loc?

User: What is the position of potassium?

Bot: Potassium is the fourth element in the first column of the periodic table.

User: What is the molar mass of sodium?

Bot: the atomic mass of sodium is 22.99 g/mol.

User: Thank you. Bye.

Bot: See you and have a nice day.

Example for part 4 of project:

User: Hello

Bot: Hello

User: Which is more reactive Na or K?

Bot: Potassium(K) is more reactive than sodium (Na).

User: Is hydrogen a metal?

Bot: No, hydrogen is nonmetal.

User: What are halogen elements?

Bot: Halogen, any of the six nonmetallic elements that constitute Group 17 (Group VIIa) of the periodic table. The halogen elements are fluorine (F), chlorine (Cl), bromine (Br), iodine (I), astatine (At), and tennessine (Ts)

User: Bye, bot

Bot: See you

Part 3 update: Adding more questions and unexpected inputs response

User: Where is the Sodium found on the periodic table?

Bot: Sodium is located in group 1 and period 1 of the periodic table.

User: How many elements are on the periodic table?

Bot: There are 118 elements on the periodic table / The periodic table has 118 elements

User: What is asdsa?

Bot: hmmm.. I've never heard that before. Not sure how to respond to that.

User: dfadsfa

Bot: Sorry, I didn't get that.

Part 4 update: Adding more questions, remove halogen questions.

User: How many electrons does Sodium have? #how many questions

Bot: Sodium has 11 electrons.

User: How about K? # how about/ what about

Bot: K has 19 electrons.

User: What is its position? #what questions with “its”

Bot: K is located in group 1 and period 3 of the periodic table.

User: Is it a metal element? # Yes/No question with “it”

Bot: Yes, K is a metal

User: OK, thank you

Bot: No problem! Any more questions? #reply to thanks

User: yes

Bot: Ask me, I'm here to help. # reply to yes

User: Is K more reactive than Na? # Yes/No on comparison

Bot: yes, you are right. K is more reactive than Na

User: Does Na have 13 electrons? # Yes/No on information

Bot: No, Na has 11 electrons

User: Is 23 the molar mass of Na?

Bot Yes, the molar mass of Na is 22.99 g/mol

User: Who discovered Na? # Who question

Bot: Humphry Davy found Na.

User: Does asd ? # invalid input

Bot: hmmm.. I've never heard that before. Not sure how to respond to that.

User: OK bye. # exit

Bot: bye

5. Input Handling.

My code will use the CYKParse and tree implementation in part 1 to parse the user input. I will make the grammar rules and lexicons for some common inputs that I predict that the user will chat or ask my bot.

_ Take input from the console

_ Use CYKParse and tree class in part 1 to parse the input and make a tree.

_ Add grammar rules and lexicons.

Example : what is the symbol of iron?

'syntax' : [

['S', 'NP', 'VP', 1.0],

['NP', 'WQuestion', 0.4],

['NP', 'Noun', 0.2],

['NP', 'Article', 0.5],

['NP', 'NP', 'NP'],

['VP', 'Verb', 0.5],

['VP', 'VP', 'NP', 0.5],

['AdverbPhrase', 'Preposition', 0.5],

['AdverbPhrase', 'AdverbPhrase', 'NP', 0.2],

],

'lexicon' : [

['WQuestion', 'What', 1.0],

['Noun', 'symbol', 0.5],

['Noun', 'iron', 0.5],

['Preposition', 'of', 1.0],

['Verb', 'is', 1.0],

['Article', 'the', 1.0],

]

_ if the input is not in any grammar rules or lexicons, the chatbot will reply a message to the user.

Part 3 update:

Use `re.findall()` function to split input into a list of strings. If the sentence has the word “name” or “am”, the unknown word X will be assumed as a [Name] and append [Name, X, 0.01] to the lexicon, else the unknown word will be appended in the lexicon as [‘Unknown, X, 0.01]’. Since the probabilities associated with CNF and C1F lexicons have no effect on the output of CYKParse, I randomly choose a number < 1 and > 0 for unknown input.

I create a separate python file for grammar rules and lexicons since I add the lexicons for 118 elements.

Currently, only save the name of the user for goodbye at the end of the chat.

Part 4 update:

For a username that has more than 2 words, assume that any word coming after “name” or “am” is username if `requestInfo[‘username’]` is empty, then concatenating all of those.

Adding more lexicon and grammarRule.

Check if the word is numeric, then add [‘Number’, word, 0.01] to the lexicon. Since the probabilities associated with CNF and C1F lexicons have no effect on the output of CYKParse, I randomly choose a number < 1 and > 0 for unknown input.

6. Internal Representations and Data Sources.

The data of the periodic table rarely changes and is stable over time, so I do not need to access any network-available database in real-time. Instead, I will use the data of the periodic table that I have found on the internet. The data has already been written, formatted, and saved in a JSON file (<https://github.com/Bowserinator/Periodic-Table-JSON>). I will download the file and attach it with my python files. As JSON is the string representation of the data, I will convert it to a python dictionary to read the data by using the `loads()` function of the JSON module. The data will look like below:

```
"hydrogen": {  
  
    "name": "Hydrogen",  
  
    "appearance": "colorless gas",  
  
    "atomic_mass": 1.008,  
  
    ...
```

Since my code uses a python dictionary to store the data, the runtime for accessing the value will be $O(1)$. Hence, the delay time during each response will be insignificant.

Part 3 update: As the data only looks up by Element's names in the JSON file, I add a dictionary that key is the symbol of element and value is the atomic number to search the data quickly. To answer questions about groups and periods of the periodic table, I add a list that has information about each group/column and a list that has information about each period/row.

```
#Symbol : atomic number
symbol_lookup = {
    'h': 1, 'he': 2, 'li': 3, 'be': 4, 'b': 5, 'c': 6, 'n': 7, 'o': 8, 'f': 9, 'ne': 10,
    'na': 11, 'mg': 12, 'al': 13, 'si': 14, 'p': 15, 's': 16, 'cl': 17, 'ar': 18,
    'k': 19, 'ca': 20, 'ga': 31, 'ge': 32, 'as': 33, 'se': 34, 'br': 35, 'kr': 36,
    'rb': 37, 'sr': 38, 'in': 49, 'sn': 50, 'sb': 51, 'te': 52, 'i': 53, 'xe': 54,
    'cs': 55, 'ba': 56, 'tl': 81, 'pb': 82, 'bi': 83, 'po': 84, 'at': 85, 'rn': 86,
    'fr': 87, 'ra': 88, 'nh': 113, 'fl': 114, 'mc': 115, 'lv': 116, 'ts': 117, 'og': 118,
    'uue': 119,
    'sc': 21, 'ti': 22, 'v': 23, 'cr': 24, 'mn': 25, 'fe': 26, 'co': 27, 'ni': 28, 'cu': 29,
    'zn': 30,
    'y': 39, 'zr': 40, 'nb': 41, 'mo': 42, 'tc': 43, 'ru': 44, 'rh': 45, 'pd': 46, 'ag': 47,
    'cd': 48,
    'hf': 72, 'ta': 73, 'w': 74, 're': 75, 'os': 76, 'ir': 77, 'pt': 78, 'au': 79, 'hg': 80,
    'rf': 104, 'db': 105, 'sg': 106, 'bh': 107, 'hs': 108, 'mt': 109, 'ds': 110, 'rg': 111,
    'cn': 112,
    'la': 57, 'ce': 58, 'pr': 59, 'nd': 60, 'pm': 61, 'sm': 62, 'eu': 63, 'gd': 64, 'tb': 65,
    'dy': 66, 'ho': 67, 'er': 68, 'tm': 69, 'yb': 70, 'lu': 71,
    'ac': 89, 'th': 90, 'pa': 91, 'u': 92, 'np': 93, 'pu': 94, 'am': 95, 'cm': 96, 'bk': 97,
    'cf': 98, 'es': 99, 'fm': 100, 'md': 101, 'no': 102, 'lr': 103,
}
```

```
%(CAS name, recommended name, list of element's atomic numbers)
group_elements = [
    ('ia', 'hydrogen and alkali metals', [1, 3, 11, 19, 37, 55, 87]),
    ('iia', 'alkaline earth metals', [4, 12, 20, 38, 56, 88]),
    ('iiib', 'scandium family', [21, 39, 71, 103]),
    ('ivb', '', [22, 40, 72, 104]),
    ('vib', '', [23, 41, 73, 105]),
    ('vib', '', [24, 42, 74, 106]),
    ('viib', '', [25, 43, 75, 107]),
    ('viib', '', [26, 44, 76, 108]),
    ('viiib', '', [27, 45, 77, 109]),
    ('viiib', '', [28, 46, 78, 110]),
    ('ib', '', [29, 47, 79, 111]),
    ('iib', '', [30, 48, 80, 112]),
    ('iiia', '', [5, 13, 31, 49, 81, 113]),
    ('iva', '', [6, 14, 32, 50, 82, 114]),
    ('va', 'pnictogens', [7, 15, 33, 51, 83, 115]),
    ('via', 'chalcogens', [8, 16, 34, 52, 84, 116]),
    ('viiia', 'halogens', [9, 17, 35, 53, 85, 117]),
    ('viiia', 'noble gases', [2, 10, 18, 36, 54, 86, 118]),
]
# (first element, last element)
```

Part 4 update:

Create copy of requestInfo dictionary to remember information from previous sentence

Hard-code a list of elements atomic numbers which are nonmetal to determine element is metal or nonmetal

```
#atomic numbers of nonmetal_element
nonmetal_element = [1, 2, 5, 6, 7, 8, 9, 10, 14, 15, 16, 17, 18, 32, 33, 34, 35, 36, 51, 52, 53, 54, 84, 85, 86, 117, 118]
```

Hard-coded more lists which use to vary the way of response.

```

user_bye = ['goodbye','bye','see you', 'see ya', 'leave', 'exit','no']
user_short_reponse = ['ok','i see', 'got it', 'thank you','thanks']

greetings = ['Hi','Hello', 'Hi there', 'Hi ya']

greetings2 = ['How can I help you?','Do you have any questions for me?']

farewells = ['Goodbye!', 'Bye Bye!','See you later!', 'OK, bye.', 'See ya!']
farewells_2 = ['', ' Have a great day!','', ' Have a good one!', ' Have a nice
day','']

start = ['', 'Well! ', '', 'hmmm..', '']
end = ['', ' Any other questions?','', ' Any more questions?','', ' Any more
questions you wanna ask?','']

reply_thanks = ['You\'re welcome!','No problem!','No worries!','Don\'t mention
it. Anym more questions?', 'My pleasure.','Anytime.','Sure!']
end_2= [ 'Any other questions?', 'Any more questions?', 'Any more questions you
wanna ask?']

errors = ['Could you repeat that please?','Sorry, I didn\'t get that.','Oops, I
didn\'t get that.','hmmm.. I\'ve never heard that before. Not sure how to
respond to that.','Sorry, I don\'t know how to answer that.','Sorry, I don\'t
know what that means.']

```

7. Output handling.

My output handling will look similar to part 1:

First, I reuse the `getSentenceParse(T)` to get the parse tree corresponding to the complete sentence.

Second, I modify the `requestInfo` dictionary and `updateRequestInfo(Tr)` to pull out the information that I need for my bot to reply.

Third, I create `getInformation()` functions to update the `requestInfo`.

Finally, I implement the `reply` function depending on the `requestInfo`, so my chatbot will answer correctly. If the chatbot is not capable of answering a user's question, it will print out a message to notify the user.

Part 3 update: To avoid crashing due to the function `max()` in `getSentenceParse()`, return `None` if the tree created `CYKParse` has no key start with S/0 (the sentence does not follow the grammar rules) before passing it to the `max()` function.

In `updateRequestInfo()`, if it finds unknown words, the function will stop immediately.

Create additional functions in PeriodictTable.py to extract information from the dictionary created from JSON file.

Create lists of different ways to respond for greetings, farewells and unexpected input, then use function randint() to randomly reply to the user.

Example: farewells = ['Goodbye!', 'Bye Bye!', 'See you later!', 'OK. Bye.', 'See ya!']

```
print(farewells[randint(0, len(farewells) - 1)])
```

To end the chat, the input must contain the word 'bye' or 'goodbye'.

Part 4 update:

Add more function to support reply() functions:

repWhatQuestion() : handle what/where/who/ How about/ What about.

repReactive(): handle Which is [more/less] [reactive/metallic/nonmetallic]

YesNo() : handle question starts with Is/Does

repHowManyQuestion(): handle question starts with "How many"

resetInfo(): reset requestInfo{}

checkSymbol(), getinfo(), checkMetal(): get information of elements from datas

Create lists of different ways to respond when user thanks the bot

```
reply_thanks = ['You\'re welcome!', 'No problem!', 'No worries!', 'Don\'t mention  
it. Any more questions?', 'My pleasure.', 'Anytime.', 'Sure!']  
end_2= [ 'Any other questions?', 'Any more questions?', 'Any more questions you  
wanna ask?']
```

When the bot asks "Any more questions?", if [Yes], will respond [Ask me, I'm here to help.]. Else [No], Bot says goodbye and the chat ends.

Expand ways to end chat, when input contains word ['bye', 'goodbye', 'see you', 'see ya', 'leave', 'exit']

Part 3 example run:

```
((base) loclinh chatbot $ python3 Proj.py
User: Hi, my name is Loc
Bot: Hi, Loc. I am Evee – an A.I assistant in Chemistry. Do you have any questions for me?

User: What is the position of Potassium on the periodic table?
Bot: Well! Potassium is located in group 1 and period 4 of the periodic table. Any more questions?

User: Where is K found on the periodic table?
Bot: K is located in group 1 and period 4 of the periodic table. Any more questions you wanna ask?

User: How many elements are on the periodic table?
Bot: The periodic table has 118 elements.

User: How many elements are on row 4?
Bot: There are 18 elements in row 4.

User: How many elements are on group IIA?
Bot: There are 6 elements in group IIA.

User: What is the atomic mass of K?
Bot: The atomic/molar mass of K is 39.10 g/mol. Any other questions?

User: What is the symbol of Iron?
Bot: hmmm..The symbol of Iron is Fe.

User: What is asd?
Bot: hmmm.. I've never heard that before. Not sure how to respond to that.

User: what is the sda of K?
Bot: Sorry, I don't know what that means.

User: asd
Bot: Sorry, I didn't get that.

User: OK bye
Bot: See ya! Loc
(base) loclinh chatbot $ █
```

Part 4 example run:

```
((base) loclinh chatbot $ python3 Proj.py
User: Hi I am Loc Khong
Bot: Hi, Loc Khong. I am Eevee – an A.I assistant in Chemistry. How can I help you?

User: What is the position of Na?
Bot: hmmm..Na is located in group 1 and period 3 of the periodic table.

User: Where is Sodium found on the periodic table?
Bot: Sodium is located in group 1 and period 3 of the periodic table. Any more questions you wanna ask?

User: What about Iron?
Bot: Iron is located in group 8 and period 4 of the periodic table.

User: What about its molar mass?
Bot: The atomic/molar mass of Iron is 55.85 g/mol. Any more questions you wanna ask?

User: Is it more reactive than Copper?
Bot: Yea, you are right. Iron is more reactive than Copper. Any other questions?

User: Is it a metal?
Bot: Yea, you are right. Iron is metal.

User: How many electrons does Copper have?
Bot: Well! Copper has 29 electrons.

User: How about Iron?
Bot: Iron has 26 electrons.

User: Does it have 20 protons?
Bot: Nope, it is not correct. Iron has 26.

User: Which is more reactive: Iron or Copper?
Bot: Iron is more reactive than Copper.

User: Which is less metallic: Na or K?
Bot: I am totally sure with you that it is Na.

User: Which is more nonmetallic: Na or Cl?
Bot: It definitely is Cl.

User: Which is less reactive: Fe or S?
Bot: It is impossible for me to compare the reactivity of metal and nonmetal elements Any more questions?

User: Who discovered Na?
Bot: Na was found by Humphry Davy. Any more questions?

User: Who discovered Iron?
Bot: hmmm..It is a mystery. The first person who discovered Iron is unknown. Any other questions?

User: yes
Bot: Ask me, I'm here to help.

User: What is more asd: Na or Cl?
Bot: Sorry, I didn't get that.

User: qwe
Bot: hmmm.. I've never heard that before. Not sure how to respond to that.

User: a
Bot: Oops, I didn't get that.

User: ok Thank you
Bot: No worries! Any other questions?

User: See you
Bot: Goodbye! Have a good one! Loc Khong

(base) loclinh chatbot $ █
```

