NLP Assignment -1

Group - Aniketh Satyanarayana, Shilpi Kumari, Swetha Vijaya Raju, Yong Zhao

1. What is the function that creates Regex objects?

- re.compile() function
- Passing a string value representing your regular expression to re.compile() returns a Regex pattern object (or simply, a Regex object).

2. Why are raw strings often used when creating Regex objects?

- Raw strings are used so that backslashes do not have to be escaped.

3. What does the search() method return?

- The search() method will return None if the regex pattern is not found in the string.
- If the pattern is found, the search() method returns a Match object, which has a group() method that will return the actual matched text from the searched string.
- The search() method returns Match objects.

4. How do you get the actual strings that match the pattern from a Match object?

- A group() method that will return the actual matched text from the searched string.

5. In the regex created from r'(\d\d\d)-(\d\d\d-\d\d\d\d)', what does group 0 cover? Group 1? Group 2?

- Group 0 - matches the entire expression.
- Group 1- covers the first set of parentheses
- Group 2- group 2 covers the second set of parentheses.

6. Parentheses and periods have specific meanings in regular expression syntax. How would you specify that you want a regex to match actual parentheses and period characters?

- Periods and parentheses can be escaped with a backslash: \., \(, and \).

7. The findall() method returns a list of strings or a list of tuples of strings. What makes it return one or the other?

- The findall() method returns all non-overlapping matches of *pattern* in *string*, as a **list of strings**. The *string* is scanned left-to-right, and matches are returned in the order found.
- If one or more groups are present in the pattern,a list of groups is returned. This will be a list of tuples if the pattern has more than one group. Empty matches are included in the result.
- If the regex has no groups, a list of strings is returned.
- If the regex has groups, a list of tuples of strings is returned.

8. What does the | character signify in regular expressions?

- It is called the pipe character.
- It can be used anywhere to match one of many expressions.

- The | character signifies matching "either, or" between two groups.

9. What two things does the ? character signify in regular expressions?

- The ? character can either mean "match zero or one of the preceding group" or be used to signify non greedy matching.

10. What is the difference between the + and * characters in regular expressions?

- The + matches one or more of the preceding character or group.
- The * matches zero or more of the preceding character or group.

11. What is the difference between {3} and {3,5} in regular expressions?

- The {3} matches exactly three instances of the preceding group.
- The {3,5} matches between three and five instances.

12. What do the \d, \w, and \s shorthand character classes signify in regular expressions?

- The \d shorthand character class matches a single digit.
- The \w shorthand character class matches any single letter, number or underscore. Same as [a-zA-Z0-9_].
- \s shorthand character class matches a space character.

13. What do the \D, \W, and \S shorthand character classes signify in regular expressions?

- The \D shorthand character class matches a single character that is not a digit.
- The \W shorthand character class matches a single character that is not a letter or number or underscore.
- The \W shorthand character class matches a single character that is not a space character.

14. What is the difference between .* and .*? ?

- .* is a greedy pattern matching. - they try to match as many characters as possible that are not newline characters.
- .*? Is non greedy pattern matching - they try to match as few characters as possible that are not newline characters.

15. What is the character class syntax to match all numbers and lowercase letters?

- Either [0-9a-z] or [a-z0-9]

16. How do you make a regular expression case-insensitive?

- Passing re.I or re.IGNORECASE as the second argument to re.compile() will make the matching case insensitive.

17. What does the . character normally match? What does it match if re.DOTALL is passed as the second argument to re.compile()?

- The . character normally matches any character except the newline character.

- If re.DOTALL is passed as the second argument to re.compile(), then the dot will match any character including the newline.

18. If numRegex = re.compile(r'\d+'), what will numRegex.sub('X', '12 drummers, 11 pipers, five rings, 3 hens') return?

- 'X drummers, X pipers, five rings, X hens'

19. What does passing re.VERBOSE as the second argument to re.compile() allow you to do?

- re.VERBOSE allows you to write regular expressions that look nicer and are more readable by allowing you to visually separate logical sections of the pattern and add comments.
- Whitespace within the pattern is ignored, except when in a character class, or when preceded by an unescaped backslash.
- When a line contains a # that is not in a character class and is not preceded by an unescaped backslash, all characters from the leftmost such # through the end of the line are ignored.

20. How would you write a regex that matches a number with commas for every three digits?

It must match the following:

1. **'42'**
2. **'1,234'**
3. **'6,368,745'**

but not the following:

1. **'12,34,567' (which has only two digits between the commas)**
2. **'1234' (which lacks commas)**

- re.compile(r'^\d{1,3}(,\d{3})*$') will create this regex, but other regex strings can produce a similar regular expression.

21. How would you write a regex that matches the full name of someone whose last name is Watanabe?You can assume that the first name that comes before it will always be one word that begins with a capital letter.

The regex must match the following:

1. **'Haruto Watanabe'**
2. **'Alice Watanabe'**
3. **'RoboCop Watanabe'**

but not the following:

1. **'haruto Watanabe' (where the first name is not capitalized)**
2. **'Mr. Watanabe' (where the preceding word has a non letter character)**
3. **'Watanabe' (which has no first name)**

4. **'Haruto watanabe' (where Watanabe is not capitalized)**

- re.compile(r'[A-Z][a-z]*\sWatanabe')

22. How would you write a regex that matches a sentence where the first word is either Alice, Bob, or Carol; the second word is either eats, pets, or throws; the third word is apples, cats, or baseballs; and the sentence ends with a period. This regex should be case-insensitive.

It must match the following:

1. **'Alice eats apples.'**
2. **'Bob pets cats.'**
3. **'Carol throws baseballs.'**
4. **'Alice throws Apples.'**
5. **'BOB EATS CATS.'**

but not the following:

1. **'RoboCop eats apples.'**
2. **'ALICE THROWS FOOTBALLS.'**
3. **'Carol eats 7 cats.'**

- re.compile(r'(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)(\s\w*)*(\.)$', re.IGNORECASE)