Natural Language Processing

Tutorial 1: Regular Expressions and Text Normalization

Dr. Sun Aixin

Question Q1

- Write regular expressions for the following languages. By “word”, we mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth. (**HINT**: please consult the book Chapter 2.1 or some websites on regular expressions.)

1. The set of all alphabetic strings;
2. The set of all lower case alphabetic strings ending with a letter *b*;
3. The set of all strings with two consecutive repeated words (e.g., “Humbert Humbert” and “the the” but not “the bug” or “the big bug”);
4. All strings that start at the beginning of the line with an integer and that end at the end of the line with a word;
5. All strings that have both the word *grotto* and the word *raven* in them (but not, e.g., words like *grottos* that merely *contain* the word *grotto*);

Question 1.1, 1.2

- The set of all alphabetic strings;
  - **[a-zA-Z]+**

- The set of all lower case alphabetic strings ending with a letter b;
  - **[a-z]*b**

Question 1.3

- The set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug");
  - **(\b[a-zA-Z]+\b)\s+\1**

- Explanation
  - [a-zA-Z]+ → all alphabetic strings
  - \s → whitespace (space, tab..)
  - \1 → used to refer to back to the first pattern in the expression which is put **inside a parentheses** ( )
    - We may have \2 or \3 to refer to the second and third patterns put inside parentheses.

Question 1.4

- All strings that start at the beginning of the line with an integer and that end at the end of the line with a word
  - **^\d+\b.*\b[a-zA-Z]+$**

- Explanation
  - \d → a digit
  - \b → a word boundary
  - ^, $ → the **beginning** and **end** of a line
  - . → a wildcard expression that matches any single character (except a carriage return)
  - * → Kleene star, zero or more occurrences of the immediate previous character or regular expression
  - .* → any string of characters

Question 1.5

- All strings that have both the word `grotto` and the word `raven` in them (but not, e.g., words like `grottos` that merely contain the word `grotto`)
  - **(.*\bgrotto\b.*\braven\b.*)|(.*\braven\b.*\bgrotto\b.*)**

- Explanation
  - The two words grotto and raven may appear in any order.
  - There could be other strings around the two words

- http://regexr.com/

Question 2

- **Try all your answers** on http://regexr.com/
- You may need to change the textbox to test two cases: the textbox contains one or more matched strings, and the textbox does not contain any matched string.
- What are the errors (e.g., false positive and false negative) have you observed?

Question 2

- The set of all alphabetic strings;
  - **[a-zA-Z]+**

- The set of all lower case alphabetic strings ending with a letter b;
  - **[a-z]*b**

- The set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug");
  - **(\b[a-zA-Z]+\b)\s+\1**

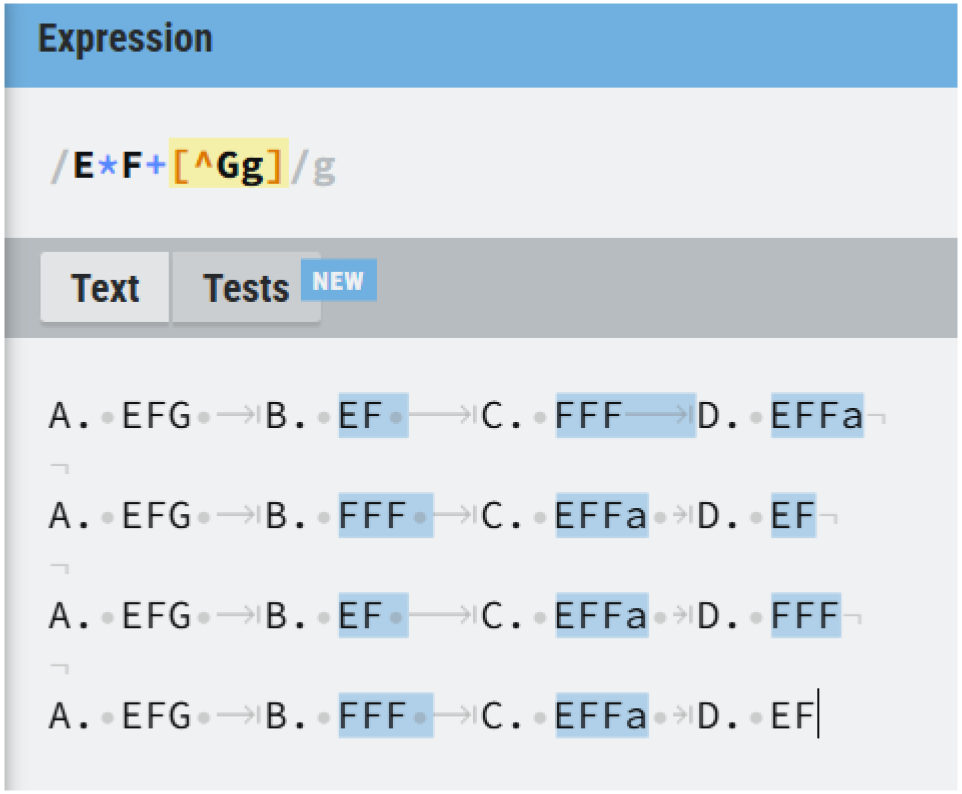
Question 2

- All strings that start at the beginning of the line with an integer and that end at the end of the line with a word
  - **^\d+\b.*\b[a-zA-Z]+$**

- All strings that have both the word `grotto` and the word `raven` in them (but not, e.g., words like `grottos` that merely contain the word `grotto`)
  - **(.*\bgrotto\b.*\braven\b.*)|(.*\braven\b.*\bgrotto\b.*)**

Question 3

- Select all strings that can be matched by regular expression /E*F+[^Gg]/
  - EFG
  - EF
  - FFF
  - EFFa



```
/E*F+[^Gg]/g
```

```
A. EFG	B. EF	C. FFF	D. EFFa
A. EFG	B. FFF	C. EFFa	D. EF
A. EFG	B. EF	C. EFFa	D. FFF
A. EFG	B. FFF	C. EFFa	D. EF
```

Question 4

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1]+\begin{cases} 1 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

- Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 1) of "idea" to "deal". Show your work.
  - **Note:<u>The costs of Ins, Del, and Sub are predefined based on the application need.</u>**

Target

Source

	#	d	e	a	l
#	**0**	1	2	3	4
i	1				
d	2				
e	3				
a	4				

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1			
d	2	1			
e	3	2			
a	4	3			

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1	2		
d	2	1	2		
e	3	2	1		
a	4	3	2		

Question 4

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1] + \begin{cases} 1 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

➢ Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 1) of "idea" to "deal". Show your work.

Target

Source

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1	2		
d	2	1	2		
e	3	2	1		
a	4	3	2		

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1	2	3	
d	2	1	2	3	
e	3	2	1	2	
a	4	3	2	1	

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1	2	3	4
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2

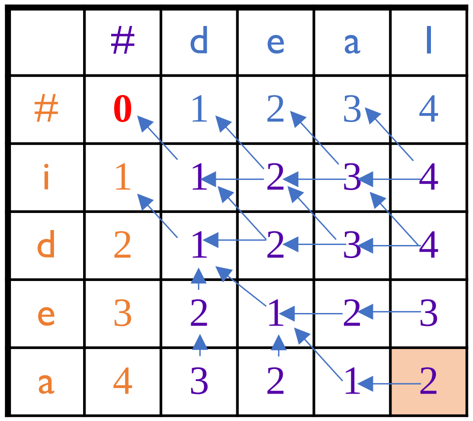
Question 4

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1]+\begin{cases} 1 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

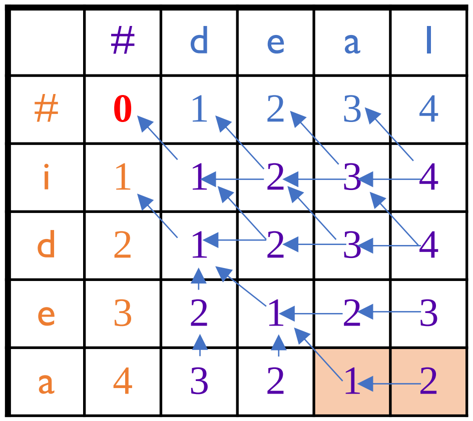
- Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 1) of "idea" to "deal". Show your work.

Target

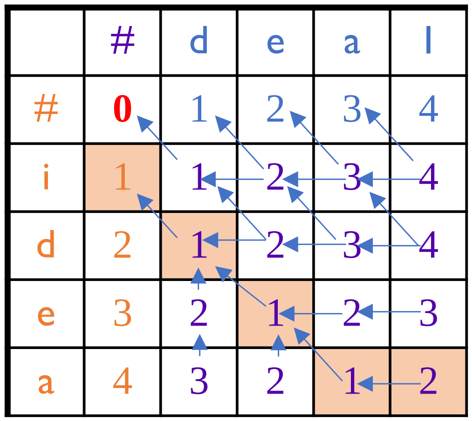
Source



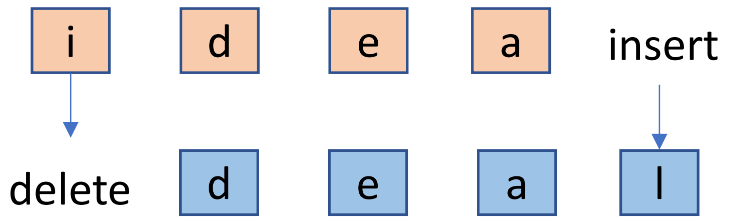
	#	d	e	a	l
#	0	1	2	3	4
i	1	1	2	3	4
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2



	#	d	e	a	l
#	0	1	2	3	4
i	1	1	2	3	4
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2



	#	d	e	a	l
#	0	1	2	3	4
i	1	1	2	3	4
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2



Question 5

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1]+\begin{cases} 2 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

- Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 2) of two sentences "computed the edit distance" to "the edit distance is computed". Show your work and show the alignment between the two strings.
  - If similar questions appear in exam, the requirement will be made clear, whether the edit distance is to be computed at character level or at word level.
  - We use the first character to represent each word, for simplicity and clarity.

Target

Source

	#	T	E	D	I	C
#	0	1	2	3	4	5
C	1					
T	2					
E	3					
D	4					

	#	T	E	D	I	C
#	0	1	2	3	4	5
C	1	2	3	4	5	4
T	2	1	2	3	4	5
E	3	2	1	2	3	4
D	4	3	2	1	2	3

Question 5

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1] + \begin{cases} 2 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

- Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 2) of two sentences "computed the edit distance" to "the edit distance is computed". Show your work and show the alignment between the two strings.

computed **the edit distance**

the edit distance is computed

Target

Source

	#	T	E	D	I	C
#	0	1	2	3	4	5
C	1	2	3	4	5	4
T	2	1	2	3	4	5
E	3	2	1	2	3	4
D	4	3	2	1	2	3

	#	T	E	D	I	C
#	0	1	2	3	4	5
C	1	2	3	4	5	4
T	2	1	2	3	4	5
E	3	2	1	2	3	4
D	4	3	2	1	2	3