Regular Expressions

Notebook:	Stanford NLP Book		
Created:	20-07-2020 23:53	**Updated:**	21-07-2020 18:04
Author:	kopalsharma2000@gmail.com		

What is Regular Expressions?
Regular Expression are sequence of characters, used to specify strings extracted from a document.

One application of Regular Expression is Text normalization
Text Normalization - Converting text into a standard form (I'm to I am, handling ☺ and #hashtags etc)
Tokenization - separating out words
Lemmitization - Determining words having the same roots (sang,sung,sing come from one root)
Stemming - simpler version of lemmitization, where just the suffix is striped of
Sentance Segmentation - Breaking text into individual sentences

There are some standard notations to refer to in RegEx

RE	Match Example	Patterns Matched
/[A-Z]/	an upper case letter	"we should call it '**D**renched Blossoms' "
#First letter in range of A-Z in highlighted		
/[a-z]/	a lower case letter	"**m**y beans were impatient to be hoed!"
#First small letter is highlighted		
/[0-9]/	a single digit	"Chapter **1**: Down the Rabbit Hole"
#First Number is highlighted		
/[wW]oodchuck/	Woodchuck or woodchuck	"Woodchuck"
#Specifies that if Capital or small any if found it must be returned.		
/[abc]/	'a', 'b', or 'c'	"In uomini, in soldati"
/[1234567890]/	any digit	"plenty of 7 to 5"
/[^A-Z]/	not an upper case letter	"O**y**fn pripetchik"
#^ symbol signifies "not". So ^A-Z means that the first NON-CAPITAL letter, which is highlighted		
/[^Ss]/	neither 'S' nor 's'	"**I** have no exquisite reason for't"
#The first non S and non s character		
/[^.]/	not a period	"our resident Djinn"
/[e^]/	either 'e' or '^'	"look up ˆ now"
/a^b/	the pattern 'a^b'	"look up aˆ b now"
/woodchucks?/	woodchuck or woodchucks	"woodchuck"
#Question mark is used to specify optional elements		
/colou?r/	color or colour	"color"
/[ab]*/	zero or more a's or b's	"aaaa" or "ababab" or "bbbb"
# Kleene(*) signifies that		
/baa+!/ ~ /baaa*!/	+ shows one of more	"baaa or baaaa or baaaa....so on"
#Kleene(+) signifies that one or more of the preceding characters can be selected. a+ is same as aa*		
/beg.n/	any character between beg and n	"**begin**, **beg'n**, **begun**"

Anchor --->
caret (^) ------- the caret ^ has three uses: to match the start of a line, to indicate

a negation inside of square brackets, and just to mean a caret.The dollar sign $ matches the end of a line. So the pattern $ is a useful
pattern for matching a space at the end of a line, and /^The dog\.$/ matches a line that contains only the phrase The dog. (We have to use the backslash here since we want the . to mean "period" and not the wildcard.)
There are also two other anchors: \b matches a word boundary, and \B matches a non-boundary.

Disjunction - to find either string1 or string2 in a text we use | operator *#/[string1 | string2]/*
Precedence - To make disjunction specific to suffixes we use ( ) operator. So to find strings guppy or guppies we do *#/gupp (y | ies)/*
Operator Precedence order is below

- Parenthesis ()
- Counters * + ? {}
- Sequences and anchors the ^my end$
- Disjunction |

greedy - patterns match the maximum string they can
non- greedy - to stop this we use *? Kleene star that matches as little text as possible. The operator +? is a Kleene plus that matches as little text as possible.

Reducing the overall error rate for an application thus involves two antagonistic efforts:

- Increasing precision (minimizing false positives)
- Increasing recall (minimizing false negatives)

RE	Match
First Patterns Matched	
*	zero or more occurrences of the previous char or expression
+	one or more occurrences of the previous char or expression
?	exactly zero or one occurrence of the previous char or expression
{n}	n occurrences of the previous char or expression
{n,m}	from n to m occurrences of the previous char or expression
{n,}	at least n occurrences of the previous char or expression
{,m}	up to m occurrences of the previous char or expression
*	an asterisk "*"
"K*****A*P*L*A*N"	
\.	a period "."
"Dr**.** Livingston, I presume"	
\?	a question mark
"Why don't they come and lend a hand**?**"	
\n	a newline
\t	a tab

This use of parentheses to store a pattern in memory is called a capture group.
Every time a capture group is used (i.e., parentheses surround a pattern), the resulting match is stored in a numbered register

Occasionally we might want to use parentheses for grouping, but don't want to capture the resulting pattern in a register. In that case we use a non-capturing group, which is specified by putting the commands ?: after the open paren, in the form (?: pattern).

Lemma - A lemma is a set of lexical forms having the same stem, the same major part-of-speech, and the same word sense.
Word form - is the full inflected or derived form of the word.
Word types - Types are the number of distinct words in a corpus
Tokens - Tokens are total number N of running words

Herdan's Law OR Heaps's Law
$|V| = k(N^b)$ where k and b are +ve constants and $0<b<1$