Tokenization. Collocations. Regular expressions.[1]

Victor Kitov

v.v.kitov@yandex.ru

[1] With materials used from "Speech and Language Processing", D. Jurafsky and J. H. Martin.

Introduction

- Course:
  - Victor Kitov
  - Anna Potapenko
  - Murat Apishev
- Lectures+seminars
  - python+scikit-learn+numpy+matplotlib+...
  - linguistic packages: NLTK, pymorphy2, gensim, ...

Recommended materials

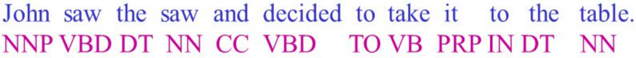
- Books:
  - Speech and Language Processing (3rd ed. draft), D. Jurafsky and J. H. Martin.
  - Speech and Language Processing (2nd ed.), D. Jurafsky and J. H. Martin. 2007.
- Video-lectures:
  - D. Jurafsky & C. Manning: Natural Language Processing.
- Resourses:
  - Resource catalog for NLP

Table of Contents

1. Overview of text mining tasks
2. Segmentation
3. Word tokenization
4. Collocations
5. Regular expressions

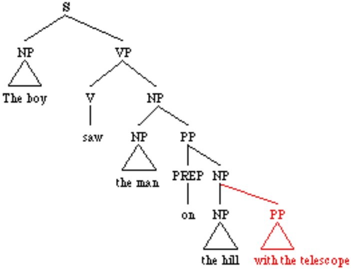
Overview of text mining tasks

- Sentence segmentation, tokenization
- Part-of-speech tagging



John	saw	the	saw	and	decided	to	take	it	to	the	table.
NNP	VBD	DT	NN	CC	VBD	TO	VB	PRP	IN	DT	NN

- Syntactic parsing



Overview of text mining tasks

- Named entity recognition
  - locate and classify named entities in text into pre-defined categories
    - people names, organizations, locations, times, quantities, monetary values, percentages
  - Jim bought 300 shares of Acme Corp. in 2006. ->
  - **[Jim]** *(Person)* bought 300 shares of **[Acme Corp.]** *(Organization)* in **[2006]** *(Time)*.
- Coreference resolution - identify expressions in a text referring to the same person or thing
  - The **music** was so loud that **it** couldn't be enjoyed.
  - Despite **her** difficulty, **Wilma** came to understand the point.
  - **Carol** told **Bob** to attend the party. **They** arrived together.
  - **Some of our colleagues** are going to be supportive. **These kinds of people** will earn our gratitude.
- Filling ontologies, information extraction.

Overview of text mining tasks

- Sentiment analysis (also known as opinion mining) - extract subjective attitudes from the text.
    - classify content into subjective (opinions) and objective (facts).
    - identify overall polarity
        - positive/negative or grade.
        - e.g.: negative movie review, rating 7 out of 10.
    - identify aspects-based attitude
        - extract individual aspects of entity
        - evaluate opinion about each aspect
        - e.g.: some cell phone review => design-excellent, battery-poor, ...

Overview of text mining tasks

- Clustering
  - identify news about the same event
  - identify books on similar subject
- Topic modelling: probabilistic co-clustering of documents and terms.
- Classification
  - classify news into different categories: politics, sports, arts, etc.
  - assign documents to authors
    - are two documents written by the same person?
  - assign documents to genres:
    - survey, scientific article, remark, textbook, etc.

Overview of text mining tasks

- Spellcheking and mistakes correction
- Automatic translation
    - Je ne l'ai pas mangé depuis six jours ->
    - I have not eaten it for six days.
- Dialog systems
    - automatic tickets reservation, ordering taxi, redirect to experts, etc.

Table of Contents

1. Overview of text mining tasks
2. Segmentation
3. Word tokenization
4. Collocations
5. Regular expressions

Sentence segmentation

- Segmentation-division of text into independent units for simplification.
- Sentence segmentation
  - natural unit of analysis for
    - POS tagging, syntactic analysis.
  - [!], [?] unambiguosly identify sentance end
  - [.] - not necessarily:
    - Mr.Johnson travelled to central office of Microsoft Inc. in the U.S.A.
    - we need to build a classifier on segmented corpus.
    - using dictionary of abbreviations may help.
  - we need a classifier to distinguish meanings of [.]

Segmentation

- Segmentation into words
- Segementation not into words, but into larger strings
  - collect phraze statistics
  - detect authorship, plagiate
- Segment into sequences of symbols
  - Man on a roof -> ma, an,n_,_o,on...
  - detect statistics of syllables, identifying language
    - e.g. classify recipes to countries of origin.
    - e.g. riso**tt**o, spaghe**tt**i - typical for Italian language

Table of Contents

Word tokenization

- Tokenization: division of text into tokens:
  - Foxes are small-to-medium-sized, omnivorous mammals. Foxes are slightly smaller than a medium-size domestic dog.
  - [Foxes] [are] [small]-[to]-[medium]-[sized], [omnivorous] [mammals]. [Foxes] [are] [slightly] [smaller] [than] [a] [medium]-[size] [domestic] [dog].
- Break dashed words? [medium]-[size] or [medium-size]?
- Count or not punctuation?
  - may reveal emotions - e.g. for sentiment analysis
  - useful for splitting into sentences , phrases=>text understanding.
  - useful for writer identification

Utterance

- Count or not utterance?
  «I do uh main- mainly business data processing»
  - types of utterances:
  - fillers: uh, um, e-mmm
  - fragments: like *[main-]*
  - may reveal emotions - e.g. for sentiment analysis
    - have different meaning, like uh, um.
  - useful in text processing-utterance begin new clause, idea.
  - useful for speaker identification

Stop words, capitalization

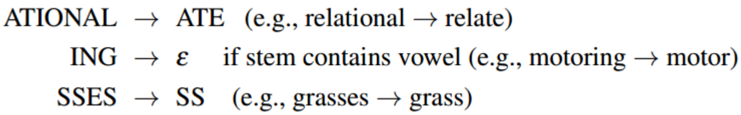
- Remove stop words?
    - and or not but,....
    - stop-words are corpus and task dependent
        - e.g. corpus of requests to city mayor - his name will be in all documents.
- Leave capitalization?
    - e.g. convert They->they?
    - capitalization is informative for, e.g., POS-tagging and named entity recognition.
    - for document classification, topic modelling - mostly not important.
    - may loose original meaning, like after US->us.

Standardization

- Standardize words or not?
    - stemming
        - remove variable endings with fixed rules
    - lemmatization
        - replace wordform with lemma
        - using dictionary
        - we look for wrods with

Stemming

- Most popular stemmer - Porter stemmer
- Stemmer as a cascade of determenistic rules, such as:



ATIONAL → ATE (e.g., relational → relate)

ING → ε if stem contains vowel (e.g., motoring → motor)

SSES → SS (e.g., grasses → grass)

Still makes errors of:

- overgeneralization:
  - organization->organ
  - policy->police
- undergeneralization:
  - analysis->analyzes
  - European->Europe

Table of Contents

1. Overview of text mining tasks
2. Segmentation
3. Word tokenization
4. Collocations
5. Regular expressions

Collocations

- Collocations are words that too frequently co-appear in text.
- Examples: New York, fast food, vice president, stock exchange, real estate, deja vu...
- Algorithm:
    - for each encountered pair of words $w_i w_j$:
        - evaluate collocation score (equal to some test statistic)
        - order word pairs by decreasing score
        - take top ranking pairs as collocations

Collocations extraction: PMI

- Pointwise mutual information:

$$PMI(w_i w_j) = \frac{p(w_i w_j)}{p(w_i)p(w_j)}$$

- $p(w_i)$ - probability to encounter word w_i in text.
- $p(w_i w_j)$ - probability to encounter word w_i and w_j immediately after.

Collocations extraction: t-test

- t-test for checking co-occurence of $w_i w_j$:
  - define $x = \mathbb{I}[w_i w_j]$
  - $\overline{x} = \frac{\#[w_i w_j]}{N}$, where N is text length
  - test statistic:

$$\frac{\overline{x} - \mu}{\sqrt{s^2/N}} \to Student(N-1) \to Normal(0,1) \text{ for } N \to \infty$$

  - where $\mu = p(w_i)p(w_j) = \frac{\#[w_i]}{N}\frac{\#[w_j]}{N}$ - expected co-occurence, given independence assumption.
  - $s^2 = \overline{x}(1-\overline{x})$ - sample variance.
  - to be a collocation test statistic should be large.

Collocations extraction: χ^2 Person test

χ^2 Pearson test for independence:

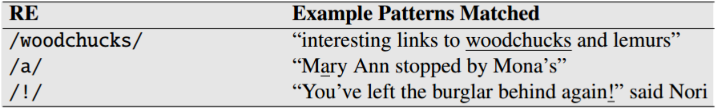
$$\begin{aligned} TS &= N\frac{[p(w_i w_j)-p(w_i)p(w_j)]^2}{p(w_i)p(w_j)} + N\frac{[p(w_i \overline{w}_j)-p(w_i)p(\overline{w}_j)]^2}{p(w_i)p(\overline{w}_j)} \\ &+ N\frac{[p(\overline{w}_i w_j)-p(\overline{w}_i)p(w_j)]^2}{p(\overline{w}_i)p(w_j)} + N\frac{[p(\overline{w}_i \overline{w}_j)-p(\overline{w}_i)p(\overline{w}_j)]^2}{p(\overline{w}_i)p(\overline{w}_j)} \end{aligned}$$

Table of Contents

Regular expressions

- re - python package for working with regular expressions.

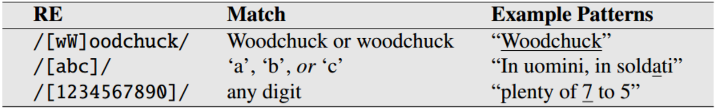
Simple match



RE	Example Patterns Matched
/woodchucks/	"interesting links to woodchucks and lemurs"
/a/	"Mary Ann stopped by Mona's"
/!/	"You've left the burglar behind again!" said Nori

- Case sensitive: /Woodchucks/ will not match woodchucks

Match any symbol from set

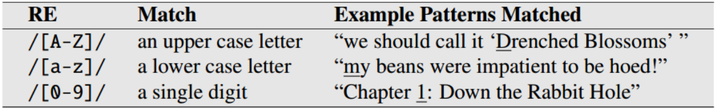


RE	Match	Example Patterns
/[wW]oodchuck/	Woodchuck or woodchuck	"Woodchuck"
/[abc]/	'a', 'b', *or* 'c'	"In uomini, in soldati"
/[1234567890]/	any digit	"plenty of 7 to 5"

Regular expressions

- match any digit: /[1234567890]/
- match any uppercase letter: /[ABCDEFGHIJKLMNOPQRSTUVWXYZ]/

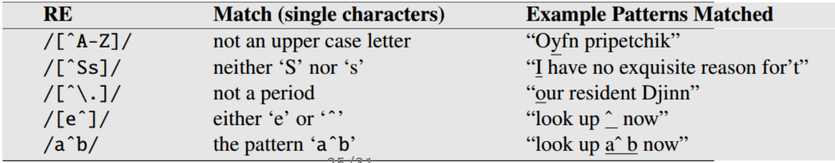
Shorter ways



RE	Match	Example Patterns Matched
/[A-Z]/	an upper case letter	“we should call it ‘Drenched Blossoms’ ”
/[a-z]/	a lower case letter	“my beans were impatient to be hoed!”
/[0-9]/	a single digit	“Chapter 1: Down the Rabbit Hole”

- range: /b-g/
  - matches b, c, d, e, f, g.

Matching except set of characters

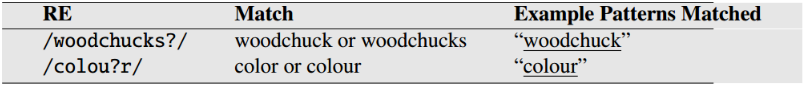


RE	Match (single characters)	Example Patterns Matched
/[ˆA-Z]/	not an upper case letter	“Oyfn pripetchik”
/[ˆSs]/	neither ‘S’ nor ‘s’	“I have no exquisite reason for’t”
/[ˆ\.]/	not a period	“our resident Djinn”
/[eˆ]/	either ‘e’ or ‘ˆ’	“look up ˆ now”
/aˆb/	the pattern ‘aˆb’	“look up aˆ b now”

Different number of occurences

p? matches pattern p or empty string.

None or single occurence



RE	Match	Example Patterns Matched
/woodchucks?/	woodchuck or woodchucks	“woodchuck”
/colou?r/	color or colour	“colour”

- p* matches 0 or more occurences of p:
  - [], [p],[pp],[ppp],...
- p+ matches 1 or more occurences:
  - [p],[pp],[ppp],...
- Recognizing sheep language: baa!, baaa!, baaaa!, ...
  - /baaa*!/
- /cat|dog/ will match [...cat...] or [...dog...].

Anchors

^ - start of string

- /**^The**/ - will match «the» only at the start of the string
- [The red brown fox]

$ - end of string

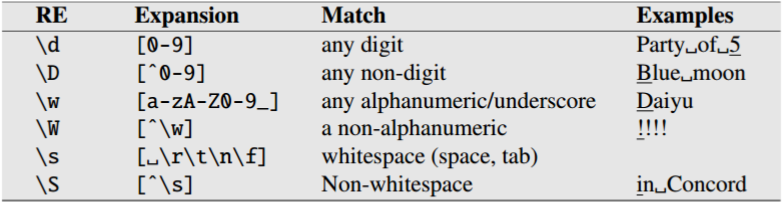
- /**.*bushes$**/ - will match «bushes» only at the start of the string
- [Fox jumped into the **bushes**.]

b matches word boundary

- a word is sequence of letters,digits and underscore
- /**bthe****b**/ matches [in **the** trees]
- /**bthe****b**/ doesn't match [other].

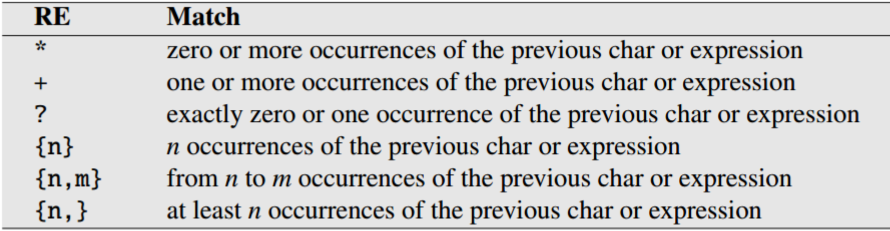
Other

- Special operators:



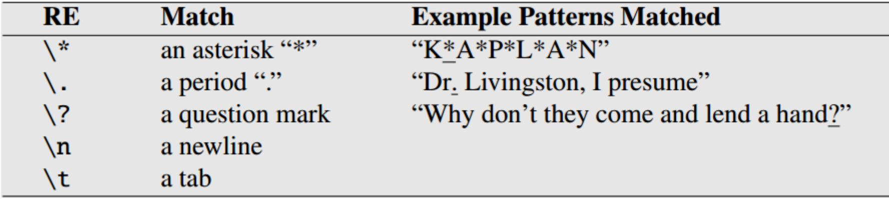
RE	Expansion	Match	Examples
`\d`	`[0-9]`	any digit	Party␣of␣5
`\D`	`[^0-9]`	any non-digit	Blue␣moon
`\w`	`[a-zA-Z0-9_]`	any alphanumeric/underscore	Daiyu
`\W`	`[^\w]`	a non-alphanumeric	!!!!
`\s`	`[␣\r\t\n\f]`	whitespace (space, tab)	
`\S`	`[^\s]`	Non-whitespace	in␣Concord

Counts to match



RE	Match
*	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

Matching reserved symbols



RE	Match	Example Patterns Matched
*	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	

Substitutions

- /the (.*)er they were, the \1er they will be/
  - will match «The bigger they were, the bigger they will be»
  - will NOT match «The bigger they were, the faster they will be»