Python and Web Data Extraction: 

Introduction
Outline

• Overview
• Text Representation
• The Natural Language Toolkit (NLTK)
• Tutorial 3: Computing TF and TF-IDF
Natural Language Processing (NLP)

• Natural language:
  – Language that is used for everyday communication by humans

• Natural Language Processing (NLP):
  – Any kind of computer manipulation of natural language.

Tools

• Text representation
  – Tokenization
  – Stop words removal
  – Stemming
  – Simple summarization
    • Frequency
    • TF-IDF
Outline

• Overview

• Text Representation
  – Overview
  – The Natural Language Toolkit (NLTK)

• Tutorial 3: Computing TF and TF-IDF
Text Representation: A Sample Text

The raw text format is not convenient for any statistical analysis

Google is a global technology leader focused on improving the ways people connect with information. We aspire to build products and provide services that improve the lives of billions of people globally.
Tokenization

• Tokenization: splitting text into words and sentences

• The “bag of words” representation
  – Each document is a “bag”
  – The “bag” contains word tokens
  – Word order is ignored
Stopwords Removal

• Stopwords:
  – Typically function words: a, an, and, as, for, in, of, the, to
  – Are usually discarded from a text representation
  – Google global technology leader focused improving ways people connect information
Stemming

• A common root may have multiple variants
  – Accounting, accountant, accountants
  – Manage, management, managing, manager

• **Stemming** is the process of reducing words to their word “stem”
  – Accounting, accountant, accountants => account
  – Manage, management, managing, manager => manag

• May not always be used
Term frequency

• Term frequency (tf)
  – How often a word occurs in the document

• Vector Space Model
  – Each document in the corpus is represented by a vector in the word space
    \[ d_i = \{ tf_{i1}, \ldots, tf_{ij}, tf_{iM} \} \]
    • \( tf_{ij} \) represents the term frequency of word \( j \) in doc \( i \)
    • \( M \) is the number of unique words in the corpus
tf-idf Model

• The tf-idf model further considers the distinctive power of words (i.e., IDF)

\[ d_i = \{tf_{i1} \cdot idf_1, \ldots, tf_{ij} \cdot idf_j, tf_{iM} \cdot idf_M\} \]

– \( tf_{ij} \) represents the term frequency of word \( j \) in doc \( i \). The log scale \( \log(1 + tf_{ij}) \) is often used in practice

– \( idf_j \) represents the inverse document frequency of word \( j \).

The log scale is \( \log \left( \frac{N}{df_j} \right) \) is often used in practice
tf-idf versus tf

tf(example, d_2) = 3  \quad \text{idf}(example, D) = \log \frac{2}{1} \approx 0.3010

tfidf(example, d_2) = tf(example, d_2) \times \text{idf}(example, D) = 3 \times 0.3010 \approx 0.9030
Outline

• Overview
• Text Representation
  – Overview
  – The Natural Language Toolkit (NLTK)
• Text Mining Tools
• Tutorial 3: Computing TF and TF-IDF
Installing NLTK package

• The Natural Language Toolkit (NLTK) provides:
  – A set of tools for the common NLP processes

• Use **pip** in your **command line interface** to install
  
  pip install nltk
## NLTK Modules

<table>
<thead>
<tr>
<th>Task</th>
<th>NLTK modules</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessing corpora</td>
<td>nltk.corpus</td>
<td>standardized interfaces to corpora and lexicons</td>
</tr>
<tr>
<td>String processing</td>
<td>nltk.tokenize,</td>
<td>tokenizers, sentence tokenizers, stemmers</td>
</tr>
<tr>
<td></td>
<td>nltk.stem</td>
<td></td>
</tr>
<tr>
<td>Collocation discovery</td>
<td>nltk.collocations</td>
<td>t-test, chi-squared, point-wise mutual information</td>
</tr>
<tr>
<td>Part-of-speech tagging</td>
<td>nltk.tag</td>
<td>n-gram, backoff, Brill, HMM, TnT</td>
</tr>
<tr>
<td>Classification</td>
<td>nltk.classify,</td>
<td>decision tree, maximum entropy, naive Bayes, EM,</td>
</tr>
<tr>
<td></td>
<td>nltk.cluster</td>
<td>k-means</td>
</tr>
<tr>
<td>Chunking</td>
<td>nltk.chunk</td>
<td>regular expression, n-gram, named-entity</td>
</tr>
<tr>
<td>Parsing</td>
<td>nltk.parse</td>
<td>chart, feature-based, unification, probabilistic,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dependency</td>
</tr>
<tr>
<td>Semantic interpretation</td>
<td>nltk.sem,</td>
<td>lambda calculus, first-order logic, model checking</td>
</tr>
<tr>
<td></td>
<td>nltk.inference</td>
<td>precision, recall, agreement coefficients</td>
</tr>
<tr>
<td>Evaluation metrics</td>
<td>nltk.metrics</td>
<td>frequency distributions, smoothed probability</td>
</tr>
<tr>
<td>Probability and estimation</td>
<td>nltk.probability</td>
<td>distributions</td>
</tr>
<tr>
<td>Applications</td>
<td>nltk.app, nltk.chat</td>
<td>graphical concordancer, parsers, WordNet browser,</td>
</tr>
<tr>
<td></td>
<td>nltk.toolbox</td>
<td>chatbots</td>
</tr>
<tr>
<td>Linguistic fieldwork</td>
<td></td>
<td>manipulate data in SIL Toolbox format</td>
</tr>
</tbody>
</table>
Outline

• Overview
• Text Representation
  – Overview
  – The Natural Language Toolkit (NLTK)
• Text Mining Tools
• Tutorial 3: Computing TF and TF-IDF
Tutorial 3: Computing TF and TF-IDF

• Download the 5tfidf.py and put it in the same folder with previous files

• Run the script.

• You will find two new files: tf.csv and tfidf.csv
Other Resources

• Natural Language Processing with Python (for Python 2)