Natural Language Processing (NLP)
(自然語言處理)

Time: 2018/11/23 (Fri) (13:10-15:00)
Place: 淡江大學商管學院 B206
Host: 鄭啟斌 教授 (淡江大學資管系碩士班 人工智慧 課程)

Min-Yuh Day
戴敏育
Assistant Professor
專任助理教授
Dept. of Information Management, Tamkang University

http://mail.tku.edu.tw/myday/
2018-11-23
Outline

• Natural Language Processing (NLP)
• Text Analytics and Text Mining
• Natural Language Processing with Python
Natural Language Processing (NLP)
Text Analytics (TA)
Text Mining (TM)
Christopher D. Manning and Hinrich Schütze (1999), 
Foundations of 
Statistical Natural Language Processing, 
The MIT Press

Dipanjan Sarkar (2016),

**Text Analytics with Python:**
A Practical Real-World Approach to Gaining Actionable Insights from your Data, Apress

Rajesh Arumugam (2018),

**Hands-On Natural Language Processing with Python:**
A practical guide to applying deep learning architectures to your NLP applications, Packt

Nitin Hardeniya (2015), NLTK Essentials, Packt Publishing

http://www.amazon.com/NLTK-Essentials-Nitin-Hardeniya/dp/1784396907
Steven Bird, Ewan Klein and Edward Loper (2009), Natural Language Processing with Python, O'Reilly Media

http://www.amazon.com/Natural-Language-Processing-Python-Steven/dp/0596516495
Text Analytics and Text Mining

TEXT ANALYTICS

Text Mining “Knowledge Discovery in Textual Data”

- Web Mining
  - Web Content Mining
  - Web Structure Mining
  - Web Usage Mining

- Data Mining
  - Classification
  - Clustering
  - Association

Natural Language Processing
- POS Tagging
- Lemmatization
- Word Disambiguation

Information Retrieval
- Link Analysis
- Search Engines
- Document Matching

Statistics
- Artificial Intelligence
- Management Science

Machine Learning
- Computer Science

Other Disciplines

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Natural Language Processing (NLP)

• Natural language processing (NLP) is an important component of text mining and is a subfield of artificial intelligence and computational linguistics.

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Natural Language Processing (NLP) and Text Mining

- Raw text
- Sentence Segmentation
- Tokenization
- Part-of-Speech (POS)
- Stop word removal
- Stemming / Lemmatization
- Dependency Parser
- String Metrics & Matching

Word’s stem:
- am → am
- having → hav

Word’s lemma:
- am → be
- having → have

Source: Nitin Hardeniya (2015), NLTK Essentials, Packt Publishing; Florian Leitner (2015), Text mining - from Bayes rule to dependency parsing
Text Summarization

Text Input

Pre-processing

Text Structure Analysis

Word Segmentation

Occurrence Statistic

POS Tagging

Keyword Extraction

Weigh Words & Sentences

Sentences Selection

Rough Summary Generation

Smoothing

Summary Output

Dictionary / Thesaurus

Topic Modeling

**Topic Modeling**


**Topics**
- gene 0.04
- dna 0.02
- genetic 0.01
-...
- life 0.02
- evolve 0.01
- organism 0.01
-...
- brain 0.04
- neuron 0.02
- nerve 0.01
-...
- data 0.02
- number 0.02
- computer 0.01
-...

**Documents**

**Seeking Life’s Bare (Genetic) Necessities**

*COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here, two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today’s organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn’t be enough. Although the numbers don’t match precisely, those predictions are not all that far apart,” especially in comparison to the 75,000 genes in the human genome, notes Siv Anderson, of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. It may be a way of organizing any newly sequenced genome,” explains Arcahy Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an


SCIENCE • VOL. 272 • 24 MAY 1996

**Topic proportions and assignments**

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
Question Answering System

Question (XML)

Question Analysis

Document Retrieval

Answer Extraction

Answer Generation

Answer Validation

Answer (XML)

Complex Essay
Simple Essay
True-or-False
Factoid
Slot-Filling
Unique

JA&EN Translator
Stanford CoreNLP
Lucene
Wikipedia
Machine Learning

Stanford CoreNLP
JA&EN Translator
Wikipedia
Lucene
Complex Essay
Simple Essay
True-or-False
Factoid
Slot-Filling
Unique
Natural Language Processing (NLP)

- Part-of-speech tagging
- Text segmentation
- Word sense disambiguation
- Syntactic ambiguity
- Imperfect or irregular input
- Speech acts
NLP Tasks

• Question answering
• Automatic summarization
• Natural language generation
• Natural language understanding
• Machine translation
• Foreign language reading
• Foreign language writing.
• Speech recognition
• Text-to-speech
• Text proofing
• Optical character recognition
NLP

Classical NLP

Documents → Language Detection → Pre-processing → Modeling → Output

- Pre-processing: Tokenization (English), Part-of-Speech Tagging (English), Stopword Removal (EN), ...
- Modeling: Feature Extraction (EN), Modeling (English), Inference (English), ...
- Output: Sentiment, Classification, Entity Extraction, Translation, Topic Modeling, ...

Deep Learning-based NLP

Documents → Preprocessing → Dense Embeddings → Hidden Layers → Output Units

- Dense Embeddings: obtained via word2vec, doc2vec, GloVe, etc.
- Output Units: Sentiment, Classification, Entity Extraction, Translation, Topic Modeling, ...

Source: http://blog.aylien.com/leveraging-deep-learning-for-multilingual/
Modern NLP Pipeline

Pre-processing

Documents → Language Detection → Pre-processed Documents

EN
Tokenize → POS Tagging → ... → Token Filtering

CN
Tokenize → POS Tagging → ... → Token Filtering

Build Vocabulary → Pre-processed Documents

Bag-of-Words & Vectorization → Machine Learning

Word Embeddings:
- word2vec
- doc2vec
- GloVe

Task / Output
- Classification
- Sentiment Analysis
- Entity Extraction
- Topic Modeling
- Similarity

Modern NLP Pipeline

[Diagram showing a flow of documents through language detection, preprocessing, and modeling stages, with tasks and outputs such as classification, sentiment analysis, entity extraction, topic modeling, and document similarity.

Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/}
Deep Learning NLP

Preprocessing → Dense Word Embeddings → Deep Neural Network

Task / Output
- Classification
- Sentiment Analysis
- Entity Extraction
- Topic Modeling
- Document Similarity

Pre-generated Lookup OR Generated in 1st level of NeuralNet

Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT uses a bidirectional Transformer.
OpenAI GPT uses a left-to-right Transformer.
ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks.
Among three, only BERT representations are jointly conditioned on both left and right context in all layers.

The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

BERT Sequence-level tasks

(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

BERT Token-level tasks

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

General Language Understanding Evaluation (GLUE) benchmark

GLUE Test results

<table>
<thead>
<tr>
<th>System</th>
<th>MNLI-(m/mm)</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST-2</th>
<th>CoLA</th>
<th>STS-B</th>
<th>MRPC</th>
<th>RTE</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>392k</td>
<td>363k</td>
<td>108k</td>
<td>67k</td>
<td>8.5k</td>
<td>5.7k</td>
<td>3.5k</td>
<td>2.5k</td>
<td></td>
</tr>
<tr>
<td>Pre-OpenAI SOTA</td>
<td>80.6/80.1</td>
<td>66.1</td>
<td>82.3</td>
<td>93.2</td>
<td>35.0</td>
<td>81.0</td>
<td>86.0</td>
<td>61.7</td>
<td>74.0</td>
</tr>
<tr>
<td>BiLSTM+ELMo+Attn</td>
<td>76.4/76.1</td>
<td>64.8</td>
<td>79.9</td>
<td>90.4</td>
<td>36.0</td>
<td>73.3</td>
<td>84.9</td>
<td>56.8</td>
<td>71.0</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>82.1/81.4</td>
<td>70.3</td>
<td>88.1</td>
<td>91.3</td>
<td>45.4</td>
<td>80.0</td>
<td>82.3</td>
<td>56.0</td>
<td>75.2</td>
</tr>
<tr>
<td>BERT_BASE</td>
<td>84.6/83.4</td>
<td>71.2</td>
<td>90.1</td>
<td>93.5</td>
<td>52.1</td>
<td>85.8</td>
<td>88.9</td>
<td>66.4</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT_LARGE</td>
<td>86.7/85.9</td>
<td>72.1</td>
<td>91.1</td>
<td>94.9</td>
<td>60.5</td>
<td>86.5</td>
<td>89.3</td>
<td>70.1</td>
<td>81.9</td>
</tr>
</tbody>
</table>

**MNLI**: Multi-Genre Natural Language Inference  
**QQP**: Quora Question Pairs  
**QNLI**: Question Natural Language Inference  
**SST-2**: The Stanford Sentiment Treebank  
**CoLA**: The Corpus of Linguistic Acceptability  
**STS-B**: The Semantic Textual Similarity Benchmark  
**MRPC**: Microsoft Research Paraphrase Corpus  
**RTE**: Recognizing Textual Entailment

A High-Level Depiction of DeepQA Architecture

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Text Analytics and Text Mining
Text Analytics

• **Text Analytics =**
  Information Retrieval +
  Information Extraction +
  Data Mining +
  Web Mining

• **Text Analytics =**
  Information Retrieval +
  Text Mining

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), *Business Intelligence, Analytics, and Data Science: A Managerial Perspective*, 4th Edition, Pearson
Text mining

- Text Data Mining
- Knowledge Discovery in Textual Databases
Application Areas of Text Mining

- Information extraction
- Topic tracking
- Summarization
- Categorization
- Clustering
- Concept linking
- Question answering

Text-Based Deception-Detection Process

1. Statements Transcribed for Processing
2. Cues Extracted & Selected
3. Text Processing Software-Generated Quantified Cues
4. Classification Models Trained and Tested on Quantified Cues
5. Statements Labeled as Truthful or Deceptive by Law Enforcement

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Multilevel Analysis of Text for Gene/Protein Interaction Identification

... expression of Bcl-2 is correlated with insufficient white blood cell death and activation of p53.

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Context Diagram for the Text Mining Process

- Unstructured data (text)
- Structured data (databases)
- Extract knowledge from available data sources
- Software/hardware limitations
- Privacy issues
- Linguistic limitations
- Domain expertise
- Tools and techniques
- Context-specific knowledge

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
The Three-Step/Task Text Mining Process

Task 1: Establish the Corpus:
Collect and organize the domain-specific unstructured data

Task 2: Create the Term-Document Matrix:
Introduce structure to the corpus

Task 3: Extract Knowledge:
Discover novel patterns from the T-D matrix

The inputs to the process include a variety of relevant unstructured (and semi-structured) data sources such as text, XML, HTML, etc.

The output of Task 1 is a collection of documents in some digitized format for computer processing.

The output of Task 2 is a flat file called a term-document matrix where the cells are populated with the term frequencies.

The output of Task 3 is a number of problem-specific classification, association, clustering models and visualizations.

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
# Term–Document Matrix

<table>
<thead>
<tr>
<th>Documents</th>
<th>Investment Risk</th>
<th>Project Management</th>
<th>Software Engineering</th>
<th>Development</th>
<th>SAP</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document 1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document 2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document 3</td>
<td>1</td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document 4</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document 5</td>
<td>1</td>
<td>2</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document 6</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), *Business Intelligence, Analytics, and Data Science: A Managerial Perspective*, 4th Edition, Pearson
Emotions

Love

Joy

Surprise

Anger

Sadness

Fear

Example of Opinion: review segment on iPhone

“I bought an iPhone a few days ago. It was such a nice phone. The touch screen was really cool. The voice quality was clear too. However, my mother was mad with me as I did not tell her before I bought it. She also thought the phone was too expensive, and wanted me to return it to the shop. ... ”

“(1) I bought an ___________ a few days ago.
(2) It was such a nice phone.
(3) The ___________ was really cool.
(4) The voice quality was clear too.
(5) However, my mother was mad with me as I did not tell her before I bought it.
(6) She also thought the phone was too expensive, and wanted me to return it to the shop. ...”
A Multistep Process to Sentiment Analysis

1. Calculate the O–S Polarity
2. Calculate the N–P Polarity of the sentiment
3. Identify the target for the sentiment
4. Record the Polarity, Strength, and the Target of the sentiment

Step 4: Tabulate & aggregate the sentiment analysis results

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Sentiment Analysis

Sentiment Classification Techniques

Sentiment Analysis

- Machine Learning Approach
  - Supervised Learning
  - Unsupervised Learning

- Lexicon-based Approach
  - Dictionary-based Approach
  - Corpus-based Approach

- Corpus-based Approach
  - Statistical
  - Semantic

- Probabilistic Classifiers
  - Decision Tree Classifiers
  - Linear Classifiers
  - Rule-based Classifiers

- Deep Learning
  - Support Vector Machine (SVM)
  - Neural Network (NN)
  - Naïve Bayes (NB)
  - Bayesian Network (BN)
  - Maximum Entropy (ME)

P–N Polarity and S–O Polarity Relationship

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Taxonomy of Web Mining

Web Mining

- **Data Mining**
  - **Web Content Mining**
    - Source: unstructured textual content of the Web pages (usually in HTML format)
  - **Web Structure Mining**
    - Source: the unified resource locator (URL) links contained in the Web pages

- **Text Mining**
  - **Web Usage Mining**
    - Source: the detailed description of a Web site’s visits (sequence of clicks by sessions)

**Search Engines**
- Page Rank
- Information Retrieval

**Sentiment Analysis**
- Social Network Analysis

**Semantic Webs**
- Social Media Analytics
- Customer Analytics

**Web Analytics**
- Search Engine Optimization
- Marketing Attribution
- Clickstream Analysis
- Weblog Analysis
- 360 Customer View
- Voice of the Customer

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Structure of a Typical Internet Search Engine

Web Usage Mining
(Web Analytics)

• **Web usage mining (Web analytics)** is the extraction of useful information from data generated through Web page visits and transactions.

• **Clickstream Analysis**
Extraction of Knowledge from Web Usage Data

Preprocess Data
- Collecting
- Merging
- Cleaning
- Structuring
  - Identify users
  - Identify sessions
  - Identify page views
  - Identify visits

Extract Knowledge
- Usage patterns
- User profiles
- Page profiles
- Visit profiles
- Customer profiles
- Customer value

How to better the data
How to improve the Web site
How to increase the customer value

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Social Analytics

• Social analytics is defined as monitoring, analyzing, measuring and interpreting digital interactions and relationships of people, topics, ideas and content.

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Branches of Social Analytics

Social Analytics

- Social Network Analysis (SNA)
- Social Media Analytics

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
Text Mining Technologies
Text Mining (TM)

Natural Language Processing (NLP)
Steven Struhl (2015), Practical Text Analytics: Interpreting Text and Unstructured Data for Business Intelligence (Marketing Science), Kogan Page

http://www.amazon.com/Practical-Text-Analytics-Interpreting-Unstructured/dp/0749474017
Text Mining Concepts

• 85-90 percent of all corporate data is in some kind of unstructured form (e.g., text)
• Unstructured corporate data is doubling in size every 18 months
• Tapping into these information sources is not an option, but a need to stay competitive

• Answer: text mining
  – A semi-automated process of extracting knowledge from unstructured data sources
  – a.k.a. text data mining or knowledge discovery in textual databases

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems
Text mining

Text Data Mining

Intelligent Text Analysis

Knowledge-Discovery in Text (KDT)

Text Mining
(text data mining)

the process of deriving high-quality information from text

http://en.wikipedia.org/wiki/Text_mining
Text Mining: the process of extracting interesting and non-trivial information and knowledge from unstructured text.

Text Mining: discovery by computer of new, previously unknown information, by automatically extracting information from different written resources.

An example of Text Mining

- Document Collection
  - Retrieve and preprocess document
- Analyze Text
  - Information Extraction
  - Classification
  - Summarization
  - Clustering
- Knowledge Management Information System

Overview of Information Extraction based Text Mining Framework

Text Data Mining

Text → Information Extraction → DB → Data Mining → Rule

NLP Libraries and Tools
Natural Language Processing with Python
– Analyzing Text with the Natural Language Toolkit

Steven Bird, Ewan Klein, and Edward Loper

This version of the NLTK book is updated for Python 3 and NLTK 3. The first edition of the book, published by O'Reilly, is available at http://nltk.org/book_1ed/. (There are currently no plans for a second edition of the book.)

0. Preface
1. Language Processing and Python
2. Accessing Text Corpora and Lexical Resources
3. Processing Raw Text
4. Writing Structured Programs
5. Categorizing and Tagging Words (minor fixes still required)
6. Learning to Classify Text
7. Extracting Information from Text
8. Analyzing Sentence Structure
9. Building Feature Based Grammars
10. Analyzing the Meaning of Sentences (minor fixes still required)
11. Managing Linguistic Data (minor fixes still required)
12. Afterword: Facing the Language Challenge

Bibliography
Term Index

This book is made available under the terms of the Creative Commons Attribution Noncommercial No-Derivative-Works 3.0 US License. Please post any questions about the materials to the nltk-users mailing list. Please report any errors on the issue tracker.

http://www.nltk.org/book/
Industrial-Strength Natural Language Processing in Python

**Fastest in the world**

spaCy excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython. Independent research has confirmed that spaCy is the fastest in the world. If your application needs to process entire web dumps, spaCy is the library you want to be using.

**Get things done**

spaCy is designed to help you do real work — to build real products, or gather real insights. The library respects your time, and tries to avoid wasting it. It's easy to install, and its API is simple and productive. I like to think of spaCy as the Ruby on Rails of Natural Language Processing.

**Deep learning**

spaCy is the best way to prepare text for deep learning. It interoperates seamlessly with TensorFlow, Keras, Scikit-Learn, Gensim and the rest of Python's awesome AI ecosystem. spaCy helps you connect the statistical models trained by these libraries to the rest of your application.

[https://spacy.io/](https://spacy.io/)
gensim

gensim

Gensim is a FREE Python library

- Scalable statistical semantics
- Analyze plain-text documents for semantic structure
- Retrieve semantically similar documents

https://radimrehurek.com/gensim/
TextBlob: Simplified Text Processing

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

```python
from textblob import TextBlob

text = 'The titular threat of The Blob has always struck me as the ultimate movie monster: an insatiably hungry, amoeba-like mass able to penetrate virtually any safeguard, capable of—as a doomed doctor chillingly describes it—"assimilating flesh on contact. Snide comparisons to gelatin be damned, it’s a concept with the most devastating of potential consequences, not unlike the grey goo scenario proposed by technological theorists fearful of artificial intelligence run rampant.'

blob = TextBlob(text)

blob.tags  # [('The', 'DT'), ('titular', 'JJ'), ('threat', 'NN'), ('of', 'IN'), ...]

blob.noun_phrases  # WordList(['titular threat', 'blob', 'ultimate movie monster', 'amoeba-like mass', ...])

for sentence in blob.sentences:
    print(sentence.sentiment.polarity)
# 0.060
```

https://textblob.readthedocs.io
Welcome to polyglot's documentation!

polyglot

Polyglot is a natural language pipeline that supports massive multilingual applications.

- Free software: GPLv3 license

Features

- Tokenization (165 Languages)
- Language detection (196 Languages)
- Named Entity Recognition (40 Languages)
- Part of Speech Tagging (16 Languages)
- Sentiment Analysis (136 Languages)
- Word Embeddings (137 Languages)
- Morphological analysis (135 Languages)
- Transliteration (69 Languages)

https://polyglot.readthedocs.io/
scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification
Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest, ...

Regression
Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, ridge regression, Lasso, ...

Clustering
Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, ...

Dimensionality reduction
Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Model selection
Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning

Preprocessing
Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.

Modules: preprocessing, feature extraction.

http://scikit-learn.org/
The Stanford NLP Group makes parts of our Natural Language Processing software available to everyone. These are statistical NLP toolkits for various major computational linguistics problems. They can be incorporated into applications with human language technology needs.

All the software we distribute here is written in Java. All recent distributions require Oracle Java 6+ or OpenJDK 7+. Distribution packages include components for command-line invocation, jar files, a Java API, and source code. A number of helpful people have extended our work with bindings or translations for other languages. As a result, much of this software can also easily be used from Python (or Jython), Ruby, Perl, Javascript, and F# or other .NET languages.

Supported software distributions

This code is being developed, and we try to answer questions and fix bugs on a best-effort basis.

All these software distributions are open source, licensed under the GNU General Public License (v2 or later). Note that this is the full GPL, which allows many free uses, but does not allow its incorporation into any type of distributed proprietary software, even in part or in translation. Commercial licensing is also available; please contact us if you are interested.

Stanford CoreNLP
An integrated suite of natural language processing tools for English and (mainland) Chinese in Java, including tokenization, part-of-speech tagging, named entity recognition, parsing, and coreference. See also: Stanford Deterministic Coreference Resolution, and the online CoreNLP demo, and the CoreNLP FAQ.

Stanford Parser
Implementations of probabilistic natural language parsers in Java: highly optimized PCFG and dependency parsers, a lexicalized PCFG parser, and a deep learning reranker. See also: Online parser demo, the Stanford Dependencies page, and Parser FAQ.

Stanford POS Tagger
A maximum-entropy (CMM) part-of-speech (POS) tagger for English,
Stanford University is located in California. It is a great university.

Part-of-Speech:

1. Stanford University is located in California.
2. It is a great university.

Named Entity Recognition:

1. Stanford University is located in California.
2. It is a great university.

Coreference:

1. Stanford University is located in California.
2. It is a great university.
Stanford University is located in California. It is a great university.

Stanford CoreNLP

http://nlp.stanford.edu:8080/corenlp/process
Stanford University is located in California. It is a great university.
Stanford University is located in California. It is a great university.
Stanford University is located in California. It is a great university.
Stanford CoreNLP

http://nlp.stanford.edu:8080/corenlp/process
Stanford University is located in California. It is a great university.
Stanford University is located in California. It is a great university.
Stanford University is located in California. It is a great university.

Sentence #2

<table>
<thead>
<tr>
<th>Id</th>
<th>Word</th>
<th>Lemma</th>
<th>Char begin</th>
<th>Char end</th>
<th>POS</th>
<th>NER</th>
<th>Normalized NER</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>It</td>
<td>it</td>
<td>46</td>
<td>48</td>
<td>PRP</td>
<td>O</td>
<td></td>
<td>PER0</td>
</tr>
<tr>
<td>2</td>
<td>is</td>
<td>be</td>
<td>49</td>
<td>51</td>
<td>VBZ</td>
<td>O</td>
<td></td>
<td>PER0</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>a</td>
<td>52</td>
<td>53</td>
<td>DT</td>
<td>O</td>
<td></td>
<td>PER0</td>
</tr>
<tr>
<td>4</td>
<td>great</td>
<td>great</td>
<td>54</td>
<td>59</td>
<td>JJ</td>
<td>O</td>
<td></td>
<td>PER0</td>
</tr>
<tr>
<td>5</td>
<td>university</td>
<td>university</td>
<td>60</td>
<td>70</td>
<td>NN</td>
<td>O</td>
<td></td>
<td>PER0</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>.</td>
<td>70</td>
<td>71</td>
<td>.</td>
<td>O</td>
<td></td>
<td>PER0</td>
</tr>
</tbody>
</table>

Parse tree:

(ROOT (S (NP (PRP It)) (VP (VBZ is) (NP (DT a) (JJ great) (NN university))) (.) .)))
Stanford University is located in California. It is a great university.
<table>
<thead>
<tr>
<th>Id</th>
<th>Word</th>
<th>Lemma</th>
<th>Char begin</th>
<th>Char end</th>
<th>POS</th>
<th>NER</th>
<th>Normalized NER</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stanford</td>
<td>Stanford</td>
<td>0</td>
<td>8</td>
<td>NNP</td>
<td>ORGANIZATION</td>
<td>ORGANIZATION</td>
<td>PER0</td>
</tr>
<tr>
<td>2</td>
<td>University</td>
<td>University</td>
<td>9</td>
<td>19</td>
<td>NNP</td>
<td>ORGANIZATION</td>
<td>ORGANIZATION</td>
<td>PER0</td>
</tr>
<tr>
<td>3</td>
<td>is</td>
<td>be</td>
<td>20</td>
<td>22</td>
<td>VBZ</td>
<td>O</td>
<td>O</td>
<td>PER0</td>
</tr>
<tr>
<td>4</td>
<td>located</td>
<td>located</td>
<td>23</td>
<td>30</td>
<td>JJ</td>
<td>O</td>
<td>O</td>
<td>PER0</td>
</tr>
<tr>
<td>5</td>
<td>in</td>
<td>in</td>
<td>31</td>
<td>33</td>
<td>IN</td>
<td>O</td>
<td>O</td>
<td>PER0</td>
</tr>
<tr>
<td>6</td>
<td>California</td>
<td>California</td>
<td>34</td>
<td>44</td>
<td>NNP</td>
<td>LOCATION</td>
<td>LOCATION</td>
<td>PER0</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>.</td>
<td>44</td>
<td>45</td>
<td>.</td>
<td>O</td>
<td>O</td>
<td>PER0</td>
</tr>
</tbody>
</table>

Parse tree
(ROOT (S (NP (NNP Stanford) (NNP University)) (VP (VBZ is) (ADJP (JJ located) (PP (IN in) (NP (NNP California)))))) ( . . ))

Uncollapsed dependencies

root ( ROOT-0, located-4 )
nn ( University-2, Stanford-1 )
nsubj ( located-4, University-2 )
cop ( located-4, is-3 )
prep ( located-4, in-5 )
pobj ( in-5, California-6 )

Collapsed dependencies

root ( ROOT-0, located-4 )
nn ( University-2, Stanford-1 )
nsubj ( located-4, University-2 )
cop ( located-4, is-3 )
prep_in ( located-4, California-6 )

Collapsed dependencies with CC processed

root ( ROOT-0, located-4 )
nn ( University-2, Stanford-1 )
nsubj ( located-4, University-2 )
cop ( located-4, is-3 )
prep_in ( located-4, California-6 )

Stanford CoreNLP

http://nlp.stanford.edu:8080/corenlp/process

Stanford University is located in California.
It is a great university.
Stanford University is located in California. It is a great university.
Bill Gates no longer Microsoft's biggest shareholder
By Patrick M. Sheridan  @CNNTech May 2, 2014: 5:46 PM ET

Bill Gates sold nearly 8 million shares of Microsoft over the past two days.

NEW YORK (CNNMoney)

For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft (MSFT, Fortune 500), bringing down his total to roughly 330 million.

That puts him behind Microsoft's former CEO Steve Ballmer who owns 333 million shares.
Related: Gates reclaims title of world's richest billionaire
Ballmer, who was Microsoft's CEO until earlier this year, was one of Gates' first hires.
It's a passing of the torch for Gates who has always been the largest single owner of his company's stock. Gates now spends his time and personal fortune helping run the Bill & Melinda Gates foundation.
The foundation has spent $28.3 billion fighting hunger and poverty since its inception back in 1997.
Bill Gates no longer Microsoft's biggest shareholder
By Patrick M. Sheridan @CNNTech May 2, 2014: 5:46 PM ET

Bill Gates sold nearly 8 million shares of Microsoft over the past two days.

New York (CNNMoney) — For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the last two days, Gates has sold nearly 8 million shares of Microsoft (MSFT, Fortune 500), bringing down his total to roughly 330 million. That puts him behind Microsoft's former CEO Steve Ballmer, who owns 333 million shares. Related: Gates reclaims title of world's richest billionaire Ballmer, who was Microsoft's CEO until earlier this year, was one of Gates' first hires. It's a passing of the torch for Gates who has always been the largest single owner of his company's stock. Gates now spends his time and personal fortune helping run the Bill & Melinda Gates foundation. The foundation has spent $28.3 billion fighting hunger and poverty since its inception back in 1997.

Potential tags:
LOCATION
TIME
PERSON
ORGANIZATION
MONEY
PERCENT
DATE

Copyright © 2011, Stanford University, All Rights Reserved.
Stanford Named Entity Tagger (NER)

http://nlp.stanford.edu:8080/ner/process

Bill Gates no longer Microsoft's biggest shareholder
By Patrick M. Sheridan @CNNTech May 2, 2014: 5:46 PM ET

Bill Gates sold nearly 8 million shares of Microsoft over the past two days.

NEW YORK (CNNMoney) -- For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft over the past two days.

For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft over the past two days.

NEW YORK (CNNMoney) -- For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft over the past two days.

For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft over the past two days.

For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft over the past two days.

For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft over the past two days.

For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft over the past two days.

Copyright © 2011, Stanford University, All Rights Reserved.
Stanford Named Entity Tagger (NER)

http://nlp.stanford.edu:8080/ner/process
Stanford Named Entity Tagger (NER)

http://nlp.stanford.edu:8080/ner/process

Bill Gates no longer Microsoft's biggest shareholder
By Patrick M. Sheridan @CNETtech May 2, 2014: 5:46 PM ET

Bill Gates sold nearly 8 million shares of Microsoft over the past two days.

Stanford Named Entity Tagger (NER)

http://nlp.stanford.edu:8080/ner/process

Stanford Named Entity Tagger

Classifier: english.conll.4class.distsim.crf.ser.gz

Output Format: highlighted

Preserve Spacing: yes

Please enter your text here:

Bill Gates no longer Microsoft's biggest shareholder
By Patrick M. Sheridan @CNNTech May 2, 2014: 5:46 PM ET

Bill Gates sold nearly 8 million shares of Microsoft over the past two days.

NEW YORK (CNNMoney) For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft (MSFT, Fortune 500), bringing down his total to roughly 330 million. That puts him behind Microsoft's former CEO Steve Ballmer, who owns 333 million shares. Related: Gates reclaim title of world's richest billionaire Ballmer, who was Microsoft's CEO until earlier this year, was one of Gates' first hires. It's a passing of the torch for Gates, who has always been the largest single owner of his company's stock. Gates now spends his time and personal fortune helping run the Bill & Melinda Gates foundation. The foundation has spent $28.3 billion fighting hunger and poverty since its inception back in 1997.
Bill Gates no longer Microsoft’s biggest shareholder
By Patrick M. Sheridan @CNNTech May 2, 2014: 5:46 PM ET

Bill Gates sold nearly 8 million shares of Microsoft over the past two
days.

NEW YORK (CNNMoney) For the first time in Microsoft’s history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft (MSFT, Fortune 500), bringing down his total to roughly 330 million. That puts him behind Microsoft’s former CEO Steve Ballmer who owns 333 million shares. Related: Gates reclaims title of world’s richest billionaire Ballmer, who was Microsoft’s CEO until earlier this year, was one of Gates’ first hires. It’s a passing of the torch for Gates who has always been the largest single owner of his company’s stock. Gates now spends his time and personal fortune helping run the Bill & Melinda Gates Foundation. The foundation has spent $28.3 billion fighting hunger and poverty since its inception back in 1997.

Potential tags:
LOCATION
ORGANIZATION
PERSON
Bill Gates no longer Microsoft's biggest shareholder By Patrick M. Sheridan @CNNTech May 2, 2014: 5:46 PM ET Bill Gates sold nearly 8 million shares of Microsoft over the past two days. NEW YORK (CNNMoney)
For the first time in Microsoft's history, founder Bill Gates is no longer its largest individual shareholder. In the past two days, Gates has sold nearly 8 million shares of Microsoft (MSFT, Fortune 500), bringing down his total to roughly 330 million. That puts him behind Microsoft's former CEO Steve Ballmer who owns 333 million shares. Related: Gates reclaims title of world’s richest billionaire Ballmer, who was Microsoft's CEO until earlier this year, was one of Gates' first hires. It's a passing of the torch for Gates who has always been the largest single owner of his company's stock. Gates now spends his time and personal fortune helping run the Bill & Melinda Gates foundation. The foundation has spent $28.3 billion fighting hunger and poverty since its inception back in 1997.

Potential tags:
- LOCATION
- TIME
- PERSON
- ORGANIZATION
- MONEY
- PERCENT
- DATE
Bill Gates no longer <ORGANIZATION>Microsoft</ORGANIZATION>'s biggest shareholder By <PERSON>Patrick M. Sheridan</PERSON> @CNNTech <DATE>May 2, 2014</DATE>: 5:46 PM ET Bill Gates sold nearly 8 million shares of <ORGANIZATION>Microsoft</ORGANIZATION> over the past two days. <LOCATION>NEW YORK</LOCATION> (CNNMoney) For the first time in <ORGANIZATION>Microsoft</ORGANIZATION>'s history, founder <PERSON>Bill Gates</PERSON> is no longer its largest individual shareholder. In the <DATE>past two days</DATE>, Gates has sold nearly 8 million shares of <ORGANIZATION>Microsoft</ORGANIZATION> (<ORGANIZATION>MSFT</ORGANIZATION>, Fortune 500), bringing down his total to roughly 330 million. That puts him behind <ORGANIZATION>Microsoft</ORGANIZATION>'s former CEO <PERSON>Steve Ballmer</PERSON> who owns 333 million shares. Related: Gates reclaims title of world's richest billionaire <PERSON>Ballmer</PERSON>, who was <ORGANIZATION>Microsoft</ORGANIZATION>'s CEO until <DATE>earlier this year</DATE>, was one of Gates' first hires. It's a passing of the torch for Gates who has always been the largest single owner of his company's stock. Gates now spends his time and personal fortune helping run the <ORGANIZATION>Bill & Melinda Gates</ORGANIZATION> foundation. The foundation has spent <MONEY>$28.3 billion</MONEY> fighting hunger and poverty since its inception back in <DATE>1997</DATE>.
歐巴馬(Nb) 是(SHI) 美國(Nc) 的(DE) 一(Neu) 位(Nf) 總統(Na)
莎士比亞在淡江 遇見賽萬提斯
2016-04-26 02:27 聯合報 記者徐葳倫／淡水報導

分享4月23日是「世界閱讀日」，也是英國大文豪莎士比亞的生日與忌日，及「唐吉訶德」作者賽萬提斯逝世之日。英專起家的淡江大學舉辦「當莎士比亞遇見賽萬提斯」活動，規畫主題書展、彩繪活動，並添購新書，拉近學生與經典文學的距離。

首波登場的「主題書展」，展出2大文豪經典作品的原著、各種譯本以及DVD、電子書等數位化資料，校方也添購許多新書，吸引學生「搶鮮」閱讀經典名作。現場還規畫「彩繪大師」，讓學生發揮創意，畫出五彩繽紛的莎士比亞和賽萬提斯人像。

英語系四年級學生陳彥伶說，讀英語系接觸莎士比亞作品，但過去沒有舉辦書展時，這些作品都放在圖書館8樓，現在搬到1樓大廳陳列，不僅有很多莎士比亞、賽萬提斯的經典新書，還可藉由電子書、電影理解兩位作家，是以前沒有過的體驗。

英語系四年級學生鄭少淮表示，莎士比亞的「馬克白」、「羅密歐與茱麗葉」都已經讀過很多次，從經典文學中理解不同城市、國家的文化。

日文系學生賴喬郁說，原本只是喜歡塗鴉才來參加活動，後來才知道畫的是2個大文豪，接觸他們的作品，文學經典「原來離我這麼近」。

淡江大學外語學院院長陳小雀表示，莎士比亞的「to be, or not to be; that is the question」，賽萬提斯的「看得越多，行得越遠，書讀得越多，知識就越廣博」，都是來自文學的名言，校方希望用最簡單的方式，讓學生知道「文學不難」，就在你我身邊。
自 2014/01/06 起，本斷詞系統已經處理過 28270134 篇文章

莎士比亞在淡江 遇見賽萬提斯
2016-04-26 02:27 聯合報 記者徐蒼倫 / 淡水報導

分享4月23日是「世界閱讀日」，也是英國大文豪莎士比亞的生日與忌日，及「唐吉訶德」作者賽萬提斯逝世之日。英專起家的淡江大學舉辦「當莎士比亞遇見賽萬提斯」活動，規畫主題書展、彩繪活動，並添購新書，拉近學生與經典文學的距離。

首波登場的「主題書展」，展出2大文豪經典作品的原著、各種譯本以及DVD、電子書等數位化資料，校方也添購許多新書，吸引學生「搶鮮」閱讀經典名作。現場還規畫「彩繪大師」，讓學生發揮創意，畫出五彩繽紛的莎士比亞和賽萬提斯人像。

英語系四年級學生陳彥伶說，讀英語系接觸莎士比亞作品，但過去沒有舉辦書展時，這些作品都放在圖書館8樓，現在搬到1樓大廳陳列，不僅有很多莎士比亞、賽萬提斯的經典新書，還可藉由電子書、電影理解兩位作家，是以前沒有過體驗。

英語系四年級學生鄭少儀表示，莎士比亞的「馬克白」、「羅密歐與茱麗葉」都已經讀過很多次，從經典文學中理解不同城市、國家的文化。

日文系學生賴喬郁說，原本只是喜歡壁鴨才來參加活動，後來才知道畫的是2個大文豪，接觸他們的作品，文學經典「原來離我這麼近」。

淡江大學外語學院院長陳小雀強調，莎士比亞的「to be, or not to be; that is the question」，賽萬提斯的「看得越多，行得越遠；書讀得越多，知識就越廣博」，都是來自文學的名言，校方希望用最簡單的方式，讓學生知道「文學不難」，就在你我身邊。
中文斷詞系統

莎士比亞 (Nb) 在 (P) 淡江 (Nb) 遇見 (VC) 賽萬提 (Nb) 斯 (Nep) 2016 (Neu) - (FW) 04 (Neu) - (FW) 2602 (Neu) : (COLONCATEGORY)

27 (Neu) 聯合報 (Nb) 記者 (Na) 徐เซ통 (Nb) 淡水 (Nc) 報導 (Na) 分享 (VJ) 4月 (Nd) 23日 (Nd) 是 (SHI) 「 (PARENTHESISCATEGORY)」

也 (D) 是 (SHI) 英國 (Nc) 大 (V1) 文豪 (Na) 莎士比亞 (Nb) 的 (DE) 生日 (Na) 與 (Caa) 思 (Na) 「 (PARENTHESISCATEGORY)」

及 (Caa) 「 (PARENTHESISCATEGORY) 唐吉訶德 (Nb)」 「 (PARENTHESISCATEGORY) 作者 (Na) 賽萬提 (Nb) 斯 (Nep) 逝世 (V1) 之 (DE) 日 (Na)」

英 (Nc) 專 (D) 起 (Caa) 的 (DE) 淡江 (Nb) 大學 (Nc) 舉辦 (VC) 「 (PARENTHESISCATEGORY) 當 (P) 莎士比亞 (Nb) 遇見 (VC) 賽萬提 (Nb)」

規畫 (VC) 主題 (Na) 書展 (Na) 「 (PAUSECATEGORY) 彩繪 (VC) 活動 (Na)」 「 (COMMACATEGORY)」

並 (Cbb) 添購 (VC) 新書 (Na) 「 (COMMACATEGORY)」

拉近 (VC) 學生 (Na) 與 (Caa) 經典 (Na) 文學 (Na) 的 (DE) 距離 (Na)」 「 (PERIODCATEGORY)」

首 (Nes) 波 (Nf) 登場 (VA) 的 (T) 「 (PARENTHESISCATEGORY) 主題 (Na) 書展 (Na)」 「 (PARENTHESISCATEGORY)」 「 (COMMACATEGORY)」

展出 (VC) 2 (Neu) 大 (V1) 文豪 (Na) 經典 (Na) 作品 (Na) 的 (DE) 原著 (Na) 「 (PAUSECATEGORY) 各 (Nes) 種 (Nf) 譯本 (Na) 以及 (Caa)」

校方 (Na) 也 (D) 添購 (VC) 許多 (Nes) 新書 (Na) 「 (COMMACATEGORY)」

吸引 (VJ) 學生 (Na) 「 (PARENTHESISCATEGORY) 搶鮮 (Na)」 「 (PARENTHESISCATEGORY)」 閱讀 (VC) 經典 (Na) 名作 (Na)」 「 (PERIODCATEGORY)」

現場 (Nc) 造 (D) 規畫 (VC) 「 (PARENTHESISCATEGORY) 彩繪 (VC) 大師 (Na)」 「 (PARENTHESISCATEGORY)」 「 (COMMACATEGORY)」

讓 (VL) 學生 (Na) 發揮 (VJ) 創意 (Na) 「 (COMMACATEGORY)」

畫出 (VC) 五彩綿紛 (V1) 的 (DE) 莎士比亞 (Nb) 和 (Caa) 賽萬提 (Nb) 斯人 (Na) 像 (VC)」 「 (PERIODCATEGORY)」

英語系 (Nc) 四年級 (Na) 學生 (Na) 陳彥伶 (Nb) 說 (VE) 「 (COMMACATEGORY)」

讓 (VC) 英語系 (Nc) 接觸 (VC) 莎士比亞 (Nb) 作品 (Na) 「 (COMMACATEGORY)」

但 (Cbb) 過去 (Nd) 沒有 (D) 舉辦 (VC) 書展 (Na) 時 (Ng) 「 (COMMACATEGORY)」

這些 (Nes) 作品 (Na) 都 (D) 放 (VC) 在 (P) 圖書館 (Nc) 八樓 (Nc) 「 (COMMACATEGORY)」
莎士比亞在淡江 遇見賽萬提斯
2016-04-26 02:27 聯合報 記者徐葳倫／淡水報導

分享4月23日是「世界閱讀日」，也是英國大文豪莎士比亞的生日與忌日，及「唐吉訶德」作者賽萬提斯逝世之日。英專起家的淡水大學舉辦「當莎士比亞遇見賽萬提斯」活動，規畫主題書展、彩繪活動，並添購新書，拉近學生與經典文學的距離。

莎士比亞(Nb) 在(P) 淡江(Nb) 遇見(VC) 賽萬提(Nb) 斯(Nep) 2016(Neu) -(FW) 04(Neu) -(FW) 26 02(Neu) :(COLONCATEGORY)
27(Neu) 聯合報(Nb) 記者(Na) 徐葳倫(Nb) 淡水(Nc) 報導(Na) 分享(VJ) 4月(Nd) 23日(Nd) 是(SHI) 「(PARENTHESISCATEGORY) 世界(Nc) 閱讀日(Na) 」
(PARENTHESISCATEGORY) ，(COMMACATEGORY)
也(D) 是(SHI) 英國(Nc) 大(VH) 文豪(Na) 莎士比亞(Nb) 的(DE) 生日(Na) 與(Caa) 忌日(Na) ，(COMMACATEGORY)
及(Caa) 「(PARENTHESISCATEGORY) 唐吉訶德(Nb) 」(PARENTHESISCATEGORY) 作者(Na) 賽萬提(Nb) 斯(Nep) 逝世(VH) 之(DE) 日(Na) 。(PERIODCATEGORY)
英(Nc) 專(D) 起家(VA) 的(DE) 淡江(Nb) 大學(Nc) 舉辦(VC) 「
(PARENTHESISCATEGORY) 當(P) 莎士比亞(Nb) 遇見(VC) 賽萬提(Nb) 斯(Nep) 」
(PARENTHESISCATEGORY) 活動(Na) ，(COMMACATEGORY)
規畫(VC) 主題(Na) 書展(Na) ，(PAUSECATEGORY) 彩繪(VC) 活動(Na) ，
(COMMACATEGORY)
並(Cbb) 添購(VC) 新書(Na) ，(COMMACATEGORY)
拉近(VC) 學生(Na) 與(Caa) 經典(Na) 文學(Na) 的(DE) 距離(Na) 。(PERIODCATEGORY)
Vector Representations of Words

Word Embeddings

Word2Vec

GloVe
Modern NLP Pipeline

[Diagram of NLP pipeline]

Documents → Language Detection → Pre-processing → Tokenize → POS Tagging → ... → Token Filtering → Pre-processed Documents

Documents → Build Vocabulary → Bag-of-Words & Vectorization → Machine Learning → (Deep) Neural Network → Task / Output

Word Embeddings: word2vec, doc2vec, GloVe

Tasks / Outputs: Classification, Sentiment Analysis, Entity Extraction, Topic Modeling, Similarity

Pre-trained word vectors

Word2Vec

wiki.zh.vec (861MB)

332647 word

300 vec

Pre-trained word vectors for 90 languages, trained on Wikipedia using fastText.

These vectors in dimension 300 were obtained using the skip-gram model with default parameters.

https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

FastText

Word2Vec: wiki.zh.vec

(861MB) (332647 word 300 vec)

https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md
Word Embeddings in LSTM RNN

Time Expanded LSTM Network

LSTM Internal States

Word Embeddings

Input Question

Is this person dancing?

Fixed length question vector encoded by the LSTM

Source: https://avisingh599.github.io/deeplearning/visual-qa/
自然語言處理與資訊檢索研究資源

http://mail.tku.edu.tw/myday/resources/

淡江大學資訊管理學系
(Department of Information Management, Tamkang University)
自然語言處理與資訊檢索研究資源
(Resources of Natural Language Processing and Information Retrieval)

1. 中央研究院CKIP中文斷詞系統
   授權單位：中央研究院詞庫小組
   授權金額：免費授權學術使用。
   授權日期：2011.03.31。
   CKIP: http://ckipsvr.iis.sinica.edu.tw/

2. 「中央研究院中英雙語詞網」(The Academia Sinica Bilingual Wordnet)
   「中央研究院中英雙語詞網」(The Academia Sinica Bilingual Wordnet)，
   授權「淡江大學資訊管理學系」(Department of Information Management, Tamkang University)學術使用。
   授權單位：中央研究院，中華民國計算語言學學會
   授權金額：「中央研究院中英雙語詞網」(The Academia Sinica Bilingual Wordnet)國內非營利機構(1-10人使用)非會員：NT$61,000元，
   授權日期：2011.05.16。
   Sinica BOW: http://bow.ling.sinica.edu.tw/
自然語言處理與資訊檢索研究資源

http://mail.tku.edu.tw/myday/resources/

3. 開放式中研院專名問答系統 (OpenASQA)
   授權單位：中央研究院資訊科學研究所智慧型代理人系統實驗室
   授權金額：免費授權學術使用。
   授權日期：2011.05.05。
   ASQA: http://asqa.iis.sinica.edu.tw/
自然語言處理與資訊檢索研究資源

http://mail.tku.edu.tw/myday/resources/

4. 哈工大資訊檢索研究中心(HIT-CIR)語言技術平臺

語料資源
哈工大資訊檢索研究中心漢語依存樹庫〔HIT-CIR Chinese Dependency Treebank〕
哈工大資訊檢索研究中心同義詞詞林擴展版〔HIT-CIR Tongyici Cilin (Extended)〕

語言處理模組

斷句 (SplitSentence: Sentence Splitting)
詞法分析 (IRLAS: Lexical Analysis System)
基於SVMTool的詞性標注 (PosTag: Part-of-speech Tagging)
命名實體識別 (NER: Named Entity Recognition)
基於動態局部優化的依存句法分析 (Parser: Dependency Parsing)
基於圖的依存句法分析 (GParser: Graph-based DP)
全文詞義消歧 (WSD: Word Sense Disambiguation)
淺層語義標注模組 (SRL: hallow Semantics Labeling)

資料表示

語言技術置標語言 (LTML: Language Technology Markup Language)

視覺化工具

LTML視覺化XSL

授權單位：哈工大資訊檢索研究中心(HIT-CIR)
授權金額：免費授權學術使用。
授權日期：2011.05.03。
HIT IR: http://ir.hit.edu.cn/
# NLP Tools: spaCy vs. NLTK

<table>
<thead>
<tr>
<th>Feature</th>
<th>SPACY</th>
<th>SYNTAXNET</th>
<th>NLTK</th>
<th>CORENLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy installation</td>
<td>🟢</td>
<td>🟥</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Python API</td>
<td>🟢</td>
<td>🟥</td>
<td>🟢</td>
<td>🟥</td>
</tr>
<tr>
<td>Multi-language support</td>
<td>🏷</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Tokenization</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Part-of-speech tagging</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Sentence segmentation</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Dependency parsing</td>
<td>🟢</td>
<td>🟢</td>
<td>🟥</td>
<td>🟢</td>
</tr>
<tr>
<td>Entity Recognition</td>
<td>🟢</td>
<td>🟥</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Integrated word vectors</td>
<td>🟢</td>
<td>🟥</td>
<td>🟥</td>
<td>🶹</td>
</tr>
<tr>
<td>Sentiment analysis</td>
<td>🟥</td>
<td>🟥</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>Coreference resolution</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🟢</td>
</tr>
</tbody>
</table>

Source: [https://spacy.io/docs/api/](https://spacy.io/docs/api/)
Natural Language Processing (NLP) spaCy

1. Tokenization
2. Part-of-speech tagging
3. Sentence segmentation
4. Dependency parsing
5. Entity Recognition
6. Integrated word vectors
7. Sentiment analysis
8. Coreference resolution

Source: https://spacy.io/docs/api/
### spaCy: Fastest Syntactic Parser

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>LANGUAGE</th>
<th>ACCURACY</th>
<th>SPEED (WPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>spaCy</td>
<td>Cython</td>
<td>91.8</td>
<td>13,963</td>
</tr>
<tr>
<td>ClearNLP</td>
<td>Java</td>
<td>91.7</td>
<td>10,271</td>
</tr>
<tr>
<td>CoreNLP</td>
<td>Java</td>
<td>89.6</td>
<td>8,602</td>
</tr>
<tr>
<td>MATE</td>
<td>Java</td>
<td>92.5</td>
<td>550</td>
</tr>
<tr>
<td>Turbo</td>
<td>C++</td>
<td>92.4</td>
<td>349</td>
</tr>
</tbody>
</table>

Source: https://spacy.io/docs/api/
## Processing Speed of NLP libraries

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>ABSOLUTE (MS PER DOC)</th>
<th>RELATIVE (TO SPACY)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOKENIZE</td>
<td>TAG</td>
</tr>
<tr>
<td>spaCy</td>
<td>0.2ms</td>
<td>1ms</td>
</tr>
<tr>
<td>CoreNLP</td>
<td>2ms</td>
<td>10ms</td>
</tr>
<tr>
<td>ZPar</td>
<td>1ms</td>
<td>8ms</td>
</tr>
<tr>
<td>NLTK</td>
<td>4ms</td>
<td>443ms</td>
</tr>
</tbody>
</table>

Source: [https://spacy.io/docs/api/](https://spacy.io/docs/api/)
# Google SyntaxNet (2016): Best Syntactic Dependency Parsing Accuracy

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>NEWS</th>
<th>WEB</th>
<th>QUESTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>spaCy</td>
<td>92.8</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Parsey McParseface</td>
<td>94.15</td>
<td>89.08</td>
<td>94.77</td>
</tr>
<tr>
<td>Martins et al. (2013)</td>
<td>93.10</td>
<td>88.23</td>
<td>94.21</td>
</tr>
<tr>
<td>Zhang and McDonald (2014)</td>
<td>93.32</td>
<td>88.65</td>
<td>93.37</td>
</tr>
<tr>
<td>Weiss et al. (2015)</td>
<td>93.91</td>
<td>89.29</td>
<td>94.17</td>
</tr>
<tr>
<td>Andor et al. (2016)</td>
<td>94.44</td>
<td>90.17</td>
<td>95.40</td>
</tr>
</tbody>
</table>

Source: [https://spacy.io/docs/api/](https://spacy.io/docs/api/)
## Named Entity Recognition (NER)

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>PRECISION</th>
<th>RECALL</th>
<th>F-MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>spaCy</td>
<td>0.7240</td>
<td>0.6514</td>
<td>0.6858</td>
</tr>
<tr>
<td>CoreNLP</td>
<td>0.7914</td>
<td>0.7327</td>
<td>0.7609</td>
</tr>
<tr>
<td>NLTK</td>
<td>0.5136</td>
<td>0.6532</td>
<td>0.5750</td>
</tr>
<tr>
<td>LingPipe</td>
<td>0.5412</td>
<td>0.5357</td>
<td>0.5384</td>
</tr>
</tbody>
</table>
Natural Language Processing with Python
# Keras preprocessing text

```python
# keras.preprocessing.text Tokenizer
from keras.preprocessing.text import Tokenizer

# define 5 documents
docs = ['Well done!', 'Good work', 'Great effort', 'nice work', 'Excellent!']

# create the tokenizer
t = Tokenizer()

# fit the tokenizer on the documents
t.fit_on_texts(docs)

t.print('docs:', docs)

t.print('word_counts:', t.word_counts)

t.print('document_count:', t.document_count)

t.print('word_index:', t.word_index)

t.print('word_docs:', t.word_docs)

# integer encode documents
texts_to_matrix = t.texts_to_matrix(docs, mode='count')

print('texts_to_matrix:')

t.print(texts_to_matrix)
```

Using TensorFlow backend.

```
docs: ['Well done!', 'Good work', 'Great effort', 'nice work', 'Excellent!']
word_counts:OrderedDict([('well', 1), ('done', 1), ('good', 1), ('work', 2), ('great', 1), ('effort', 1), ('nice', 1), ('excellent', 1), 'document_count':5
word_index: {'work': 1, 'well': 2, 'done': 3, 'good': 4, 'great': 5, 'effort': 6, 'nice': 7, 'excellent': 8}
word_docs: {'done': 1, 'well': 1, 'work': 2, 'good': 1, 'great': 1, 'effort': 1, 'nice': 1, 'excellent': 1}

texts_to_matrix:
[[0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 1. 0.]
 [0. 1. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1.]]
```
Text Classification

Source: https://developers.google.com/machine-learning/guides/text-classification/
Text Classification Workflow

- Step 1: Gather Data
- Step 2: Explore Your Data
- Step 2.5: Choose a Model*
- Step 3: Prepare Your Data
- Step 4: Build, Train, and Evaluate Your Model
- Step 5: Tune Hyperparameters
- Step 6: Deploy Your Model

Source: https://developers.google.com/machine-learning/guides/text-classification/
Text Classification S/W<1500: N-gram

Text Classification S/W>=1500: Sequence

Select top_k features [freq]

min(top: 1K, 2K, ..., 15K, 20K, 25K, ..., 90K, all)

Normalization mode

samplewise None featurewise

Embeddings

S/W < 15K

Yes

Fine-tuned pre-trained embedding

No

Frozen pre-trained embedding

Embeddings learned from scratch

Build model

RNN stacked RNN CNN-RNN sepCNN CNN

Hyperparameter tuning

Step 2.5: Choose a Model

Samples/Words < 1500

150,000/100 = 1500

IMDb review dataset, the samples/words-per-sample ratio is ~ 144
Step 2.5: Choose a Model

Samples/Words < 15,000

1,500,000/100 = 15,000

Prepare model

Build model

SVM
MLP
GBDT

Embbedings

Yes S/W < 15K

No

Fine-tuned pre-trained embedding

Frozen pre-trained embedding

Embeddings learned from scratch

Build model

RNN
stacked RNN
CNN-RNN
sepCNN
CNN

Hyperparameter tuning

Step 3: Prepare Your Data

Texts:
T1: 'The mouse ran up the clock'
T2: 'The mouse ran down'

Token Index:
{'the': 1, 'mouse': 2, 'ran': 3, 'up': 4, 'clock': 5, 'down': 6}.
NOTE: 'the' occurs most frequently,
so the index value of 1 is assigned to it.
Some libraries reserve index 0 for unknown tokens,
as is the case here.

Sequence of token indexes:
T1: 'The mouse ran up the clock' =
[1, 2, 3, 4, 1, 5]
T1: 'The mouse ran down' =
[1, 2, 3, 6]
# One-hot encoding

'The mouse ran up the clock' =

<table>
<thead>
<tr>
<th>Word</th>
<th>Index</th>
<th>One-hot Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>1</td>
<td>[0, 1, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>mouse</td>
<td>2</td>
<td>[0, 0, 1, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>ran</td>
<td>3</td>
<td>[0, 0, 0, 1, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>up</td>
<td>4</td>
<td>[0, 0, 0, 0, 1, 0, 0, 0]</td>
</tr>
<tr>
<td>the</td>
<td>1</td>
<td>[0, 1, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>clock</td>
<td>5</td>
<td>[0, 0, 0, 0, 0, 1, 0, 0]</td>
</tr>
</tbody>
</table>

[0, 1, 2, 3, 4, 5, 6]
Word embeddings

Male-Female

Verb Tense

Country-Capital

Source: https://developers.google.com/machine-learning/guides/text-classification/step-3
Word embeddings

The mouse ran up the clock
- the: 1
- mouse: 2
- ran: 3
- up: 4
- clock: 5

The mouse ran down
down: 6

[1, 2, 3, 4, 1, 5]

Embedding layer (output dim = 4)

[[0.236, -0.141, 0.000, 0.045],
 [0.006, 0.652, 0.270, -0.556],
 [0.305, 0.569, -0.028, 0.496],
 [0.421, 0.195, -0.058, 0.477],
 [0.236, -0.141, 0.000, 0.045],
 [0.844, -0.001, 0.763, 0.201]]

[[0.236, -0.141, 0.000, 0.045],
 [0.006, 0.652, 0.270, -0.556],
 [0.305, 0.569, -0.028, 0.496],
 [0.466, -0.326, 0.884, 0.007]]

Source: https://developers.google.com/machine-learning/guides/text-classification/step-3
t1 = 'The mouse ran up the clock'
t2 = 'The mouse ran down'
s1 = t1.lower().split(' ')
s2 = t2.lower().split(' ')
terms = s1 + s2
sortedset = sorted(set(terms))
print('terms =', terms)
print('sortedset =', sortedset)
t1 = 'The mouse ran up the clock'
t2 = 'The mouse ran down'
s1 = t1.lower().split(' ')
s2 = t2.lower().split(' ')
terms = s1 + s2
print(terms)

tfdict = {}
for term in terms:
    if term not in tfdict:
        tfdict[term] = 1
    else:
        tfdict[term] += 1

a = []
for k,v in tfdict.items():
    a.append('{} , {}'.format(k, v))
print(a)

['the', 'mouse', 'ran', 'up', 'the', 'clock', 'the', 'mouse', 'ran', 'down']
['the', 3, 'mouse', 2, 'ran', 2, 'up', 1, 'clock', 1, 'down', 1]
```python
sorted_by_value_reverse = sorted(tfdict.items(), key=lambda kv: kv[1], reverse=True)

sorted_by_value_reverse_dict = dict(sorted_by_value_reverse)

id2word = {id: word for id, word in enumerate(sorted_by_value_reverse_dict)}

word2id = dict([(v, k) for (k, v) in id2word.items()])
```

```
sorted_by_value: [('up', 1), ('clock', 1), ('down', 1), ('mouse', 2), ('ran', 2), ('the', 3)]
sorted_by_value2: ['the', 'mouse', 'ran', 'up', 'clock', 'down']
sorted_by_value_reverse: [('the', 3), ('mouse', 2), ('ran', 2), ('up', 1), ('clock', 1), ('down', 1)]
sorted_by_value_reverse_dict {'the': 3, 'mouse': 2, 'ran': 2, 'up': 1, 'clock': 1, 'down': 1}
id2word {0: 'the', 1: 'mouse', 2: 'ran', 3: 'up', 4: 'clock', 5: 'down'}
word2id {'the': 0, 'mouse': 1, 'ran': 2, 'up': 3, 'clock': 4, 'down': 5}
len_word: 6
sorted_by_key: [('clock', 1), ('down', 1), ('mouse', 2), ('ran', 2), ('the', 3), ('up', 1)]
the, 3
mouse, 2
ran, 2
up, 1
clock, 1
down, 1
```
sorted_by_value = sorted(tfdict.items(), key=lambda kv: kv[1])
print('sorted_by_value: ', sorted_by_value)

sorted_by_value2 = sorted(tfdict, key=tfdict.get, reverse=True)
print('sorted_by_value2: ', sorted_by_value2)

sorted_by_value_reverse = sorted(tfdict.items(), key=lambda kv: kv[1], reverse=True)
print('sorted_by_value_reverse: ', sorted_by_value_reverse)

sorted_by_value_reverse_dict = dict(sorted_by_value_reverse)
print('sorted_by_value_reverse_dict: ', sorted_by_value_reverse_dict)

id2word = {id: word for id, word in enumerate(sorted_by_value_reverse_dict)}
print('id2word', id2word)

word2id = dict([(v, k) for (k, v) in id2word.items()])
print('word2id', word2id)

print('len_words: ', len(word2id))

sorted_by_key = sorted(tfdict.items(), key=lambda kv: kv[0])
print('sorted_by_key: ', sorted_by_key)

tfstring = '\n'.join(a)
print(tfstring)

tf = tfdict.get('mouse')
print(tf)
from keras.preprocessing.text import Tokenizer

docs = ['Well done!', 'Good work', 'Great effort', 'nice work', 'Excellent!']

# create the tokenizer
t = Tokenizer()
t.fit_on_texts(docs)

print('docs:', docs)
print('word_counts:', t.word_counts)
print('document_count:', t.document_count)
print('word_index:', t.word_index)
print('word_docs:', t.word_docs)

# integer encode documents
texts_to_matrix = t.texts_to_matrix(docs, mode='count')

print('texts_to_matrix:')
print(texts_to_matrix)
from keras.preprocessing.text import Tokenizer
# define 5 documents
docs = ['Well done!', 'Good work', 'Great effort', 'nice work', 'Excellent!']
# create the tokenizer
t = Tokenizer()
# fit the tokenizer on the documents
t.fit_on_texts(docs)
print('docs:', docs)
print('word_counts:', t.word_counts)
print('document_count:', t.document_count)
print('word_index:', t.word_index)
print('word_docs:', t.word_docs)
# integer encode documents
texts_to_matrix = t.texts_to_matrix(docs, mode='count')
print('texts_to_matrix:')
print(texts_to_matrix)
texts_to_matrix =
t.texts_to_matrix(docs, mode='count')

docs: ['Well done!', 'Good work', 'Great effort', 'nice work', 'Excellent!']
word_counts: OrderedDict([( 'well', 1), ( 'done', 1), ('good', 1), ( 'work', 2), ( 'great', 1), ( 'effort', 1), ( 'nice', 1), ( 'excellent', 1)])
document_count: 5
word_index: { 'work': 1, 'well': 2, 'done': 3, 'good': 4, 'great': 5, 'effort': 6, 'nice': 7, 'excellent': 8}
word_docs: { 'done': 1, 'well': 1, 'work': 2, 'good': 1, 'great': 1, 'effort': 1, 'nice': 1, 'excellent': 1}
texts_to_matrix:
[[0. 0. 1. 1. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 1. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1.]]

Source: https://machinelearningmastery.com/prepare-text-data-deep-learning-keras/
from keras.preprocessing.text import Tokenizer

# define 5 documents
docs = ['Well done!', 'Good work', 'Great effort', 'nice work', 'Excellent!']

# create the tokenizer
tokenizer = Tokenizer()
# fit the tokenizer on the documents
tokenizer.fit_on_texts(docs)
print('docs:', docs)
print('word_counts:', tokenizer.word_counts)
print('document_count:', tokenizer.document_count)
print('word_index:', tokenizer.word_index)
print('word_docs:', tokenizer.word_docs)

# integer encode documents
texts_to_matrix = tokenizer.texts_to_matrix(docs, mode='tfidf')
print('texts_to_matrix:')
print(texts_to_matrix)

Source: https://machinelearningmastery.com/prepare-text-data-deep-learning-keras/
Natural Language Toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The book is being updated for Python 3 and NLTK 3. (The original Python 2 version is still available at http://nltk.org/book_1ed.)

Some simple things you can do with NLTK

Tokenize and tag some text:

```python
>>> import nltk
```
<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
<th>Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>matplotlib</td>
<td>2.0.0</td>
<td>np111py36_0</td>
</tr>
<tr>
<td>mistune</td>
<td>0.7.3</td>
<td>py36_1</td>
</tr>
<tr>
<td>mkl</td>
<td>2017.0.1</td>
<td></td>
</tr>
<tr>
<td>mkl-service</td>
<td>1.1.2</td>
<td>py36_3</td>
</tr>
<tr>
<td>mpmath</td>
<td>0.19</td>
<td>py36_1</td>
</tr>
<tr>
<td>multipledispatch</td>
<td>0.4.9</td>
<td>py36_0</td>
</tr>
<tr>
<td>nbconvert</td>
<td>4.2.0</td>
<td>py36_0</td>
</tr>
<tr>
<td>nbformat</td>
<td>4.2.0</td>
<td>py36_0</td>
</tr>
<tr>
<td>networkx</td>
<td>1.11</td>
<td>py36_0</td>
</tr>
<tr>
<td>nltk</td>
<td>3.2.2</td>
<td>py36_0</td>
</tr>
<tr>
<td>nose</td>
<td>1.5.7</td>
<td>py36_1</td>
</tr>
<tr>
<td>notebook</td>
<td>4.3.1</td>
<td>py36_0</td>
</tr>
<tr>
<td>numba</td>
<td>0.30.1</td>
<td>np111py36_0</td>
</tr>
<tr>
<td>numexpr</td>
<td>2.6.1</td>
<td>np111py36_2</td>
</tr>
<tr>
<td>numpy</td>
<td>1.11.3</td>
<td>py36_0</td>
</tr>
<tr>
<td>numpydoc</td>
<td>0.6.0</td>
<td>py36_0</td>
</tr>
<tr>
<td>odo</td>
<td>0.5.0</td>
<td>py36_1</td>
</tr>
<tr>
<td>openpyxl</td>
<td>2.4.1</td>
<td>py36_0</td>
</tr>
<tr>
<td>openssl</td>
<td>1.0.2k</td>
<td></td>
</tr>
<tr>
<td>pandas</td>
<td>0.19.2</td>
<td>np111py36_1</td>
</tr>
<tr>
<td>pandas-datareader</td>
<td>0.2.1</td>
<td>py36_0</td>
</tr>
<tr>
<td>partd</td>
<td>0.3.7</td>
<td>py36_0</td>
</tr>
<tr>
<td>path.py</td>
<td>10.0</td>
<td>py36_0</td>
</tr>
<tr>
<td>pathlib2</td>
<td>2.2.0</td>
<td>py36_0</td>
</tr>
<tr>
<td>patsy</td>
<td>0.4.1</td>
<td>py36_0</td>
</tr>
<tr>
<td>pep8</td>
<td>1.7.0</td>
<td>py36_0</td>
</tr>
<tr>
<td>pexpect</td>
<td>4.2.1</td>
<td>py36_0</td>
</tr>
<tr>
<td>pickleshare</td>
<td>0.7.4</td>
<td>py36_0</td>
</tr>
<tr>
<td>pillow</td>
<td>4.0.0</td>
<td>py36_0</td>
</tr>
<tr>
<td>pip</td>
<td>9.0.1</td>
<td>py36_1</td>
</tr>
<tr>
<td>plotly</td>
<td>1.12.9</td>
<td>py36_0</td>
</tr>
<tr>
<td>ply</td>
<td>3.9</td>
<td>py36_0</td>
</tr>
<tr>
<td>prompt_toolkit</td>
<td>1.0.9</td>
<td>py36_0</td>
</tr>
<tr>
<td>psutil</td>
<td>5.0.1</td>
<td>py36_0</td>
</tr>
</tbody>
</table>
help('modules')
import nltk

Source: http://www.nltk.org/
import nltk
nltk.download()
import nltk
nltk.download()
```python
import nltk
nltk.download()
```

Source: [http://www.nltk.org/](http://www.nltk.org/)
nltk_data

- chunkers
- corpora
- grammars
- help
- models
- stemmers
- taggers
- tokenizers
At eight o'clock on Thursday morning Arthur didn't feel very good.
At eight o'clock on Thursday morning, Arthur didn't feel very good.
```python
import nltk
sentence = "At eight o'clock on Thursday morning Arthur didn't feel very good."
tokens = nltk.word_tokenize(sentence)
tokens

print(tokens)
```

```
In [1]: import nltk
   sentence = "At eight o'clock on Thursday morning Arthur didn't feel very good."
   tokens = nltk.word_tokenize(sentence)
   tokens
Out[1]: ['At', 'eight', 'o'clock', 'on', 'Thursday', 'morning', 'Arthur', 'did', 'n't', 'feel', 'very', 'good', '.']

In [2]: print(tokens)
   
   ['At', 'eight', 'o'clock', 'on', 'Thursday', 'morning', 'Arthur', 'did', 'n't', 'feel', 'very', 'good', '.']
```

Source: [http://www.nltk.org/](http://www.nltk.org/)
```
tagged = nltk.pos_tag(tokens)
tagged[0:6]
```

```
In [3]:  tagged = nltk.pos_tag(tokens)
tagged[0:6]

Out[3]:  [('At', 'IN'),
        ('eight', 'CD'),
        ('o'clock', 'NN'),
        ('on', 'IN'),
        ('Thursday', 'NNP'),
        ('morning', 'NN')]
```
tagged

In [4]: tagged

Out[4]: [('At', 'IN'), ('eight', 'CD'),
        ('o\'clock', 'NN'), ('on', 'IN'),
        ('Thursday', 'NNP'), ('morning', 'NN'),
        ('Arthur', 'NNP'), ('did', 'VBD'),
        ('n\'t', 'RB'), ('feel', 'VB'),
        ('very', 'RB'), ('good', 'JJ'), ('.', '.')]
At eight o'clock on Thursday morning
Arthur didn't feel very good.
entities = nltk.chunk.ne_chunk(tagged)

```
Tree('S', [('At', 'IN'), ('eight', 'CD'), ('o'clock', 'NN'), ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'), Tree('PERSON', [('Arthur', 'NNP')]), ('did', 'VBD'), ('n't', 'RB'), ('feel', 'VB'), ('very', 'RB'), ('good', 'JJ'), ('.', '.')])
```

Source: [http://www.nltk.org/](http://www.nltk.org/)
from nltk.corpus import treebank
t = treebank.parsed_sents('wsj_0001.mrg')[0]
t.draw()
wsj_0001.mrg
(S
  (NP-SBJ
    (NP (NNP Pierre) (NNP Vinken))
    (, ,)
    (ADJP
      (NP (CD 61) (NNS years))
      (JJ old))
    (, ,)
  )
  (VP (MD will)
    (VP (VB join)
      (NP (DT the) (NN board))
      (PP-CLR (IN as)
        (NP (DT a) (JJ nonexecutive) (NNN director))
        (NP-TMP (NNP Nov.) (CD 29)))))
  (.)
)

(S
  (NP-SBJ (NNP Mr.) (NNP Vinken))
  (VP (VBZ is)
    (NP-PRD
      (NP (NNN chairman))
      (PP (IN of)
        (NP
          (NP (NNP Elsevier) (NNP N.V.))
          (, ,)
          (NP (DT the) (NNP Dutch) (VBG publishing) (NNN group))))
    (.)
  ))
)
Pragmatic NLP

Pragmatic NLP - Live Demo

Dataset: CNN Facebook Posts 2012-2016

Source: https://data.world/martinchek/2012-2016-facebook-posts

In [1]:
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
matplotlib.style.use('ggplot')

import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from textblob import TextBlob

# Don't forget to fetch necessary models for TextBlob's NLTK hooks to function > 'python -m textblob.download_corpora'

import json
import multiprocessing
import re

In [2]:
fname_data = '/Volumes/SD/datasets/facebook-news/cnn-5550296508.csv-cnn-5550296508.csv'

1. Ingest Data

In [3]:
pd_data = pd.read_csv(fname_data, encoding='utf-16', na_values='NULL', quoting=1)

In [ ]:
pd_data.id = pd_data['id'].map(lambda x : x.replace('', ''))

https://github.com/fortiema/notebooks/blob/master/Pragmatic%20NLP.ipynb
**Python Jieba**

"结巴"中文分词

[GitHub link](https://github.com/fxsjy/jieba)

**Repository Details**

- **Issues**: 226
- **Pull requests**: 14
- **Projects**: 0
- **Wiki**:
- **Branch**: master

**Commit History**

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>extra_dict</td>
<td>update to v0.33</td>
<td>2 years ago</td>
</tr>
<tr>
<td>jieba</td>
<td>Bugfix for HMM=False in parallelism.</td>
<td>6 months ago</td>
</tr>
<tr>
<td>test</td>
<td>Bugfix for HMM=False in parallelism.</td>
<td>6 months ago</td>
</tr>
<tr>
<td>.gitattributes</td>
<td>first commit</td>
<td>4 years ago</td>
</tr>
<tr>
<td>.gitignore</td>
<td>update jieba3k</td>
<td>2 years ago</td>
</tr>
<tr>
<td>Changelog</td>
<td>version change 0.38</td>
<td>a year ago</td>
</tr>
<tr>
<td>LICENSE</td>
<td>add a license file</td>
<td>4 years ago</td>
</tr>
<tr>
<td>MANIFEST.in</td>
<td>include Changelog &amp; README.md in the distribution package</td>
<td>4 years ago</td>
</tr>
<tr>
<td>README.md</td>
<td>Update README.md</td>
<td>8 months ago</td>
</tr>
</tbody>
</table>
import jieba
import jieba.posseg as pseg
sentence = "銀行產業正在改變，金融機構欲挖角科技人才"
words = jieba.cut(sentence)
print(sentence)
print(" ".join(words))
wordspos = pseg.cut(sentence)
result = ''
for word, pos in wordspos:
    print(word + ' (' + pos + ')')
    result = result + ' ' + word + '(' + pos + ')
print(result.strip())
import jieba
cut = jieba.cut(sentence)

words = jieba.cut(sentence)
print(sentence)
print(' '.join(words))  # 銀行 產業 正在 改變 ， 金融 機構 欲 挖角 科技人才

wordspos = pseg.cut(sentence)
result = ''
for word, pos in wordspos:
    print(word + ' (' + pos + ')')
    result = result + ' ' + word + ' (' + pos + ')
print(result.strip())  # 銀行(n) 產業(n) 正在(t) 改變(v) ， (x) 金融(n) 機構(n) 欲(d) 挖角(n) 科技人才(n)

銀行 產業 正在 改變 ， 金融 機構 欲 挖角 科技人才
銀行 產業 正在 改變 ， 金融 機構 欲 挖角 科技人才
銀行(n) 產業(n) 正在(t) 改變(v) ， (x) 金融(n) 機構(n) 欲(d) 挖角(n) 科技人才(n)
Python Jieba “结巴”中文分词

- https://github.com/fxsjy/jieba
- jieba.set_dictionary('data/dict.txt.big')
  - #/anaconda/lib/python3.5/site-packages/jieba
  - dict.txt (5.4MB)(349,046)
  - dict.txt.big.txt (8.6MB)(584,429)
  - dict.txt.small.txt (1.6MB)(109,750)
  - dict.tw.txt (4.2MB)(308,431)
- https://github.com/ldkrsi/jieba-zh_TW
  - 結巴中文斷詞台灣繁體版本
AI and Deep Machine Learning

• Artificial Intelligence (AI)
  – AI is the broadest term, applying to any technique that enables computers to mimic human intelligence, using logic, if-then rules, decision trees, and machine learning (including deep learning).

• Machine Learning (ML)
  – The subset of AI that includes abstruse statistical techniques that enable machines to improve at tasks with experience. The category includes deep learning.

• Deep Learning (DL)
  – The subset of machine learning composed of algorithms that permit software to train itself to perform tasks, like speech and image recognition, by exposing multilayered neural networks to vast amounts of data.

TensorFlow NLP Examples

• Basic Text Classification  
  (Text Classification) (46 Seconds)  

• NMT with Attention  
  (20-30 minutes)  
Text Classification
IMDB Movie Reviews

https://colab.research.google.com/drive/1x16h1GhHsLIrLYtPCvCHaoO1W-i_gror

Copyright 2018 The TensorFlow Authors.
Licensed under the Apache License, Version 2.0 (the "License");
MIT License

Text classification with movie reviews
Download the IMDB dataset
Explore the data
- Convert the integers back to words
Prepare the data
Build the model
- Hidden units
- Loss function and optimizer
Create a validation set
Train the model
Evaluate the model

Text classification with movie reviews

This notebook classifies movie reviews as positive or negative using the text of the review. This is an example of binary—or two-class—classification, an important and widely applicable kind of machine learning problem.

We'll use the IMDB dataset that contains the text of 50,000 movie reviews from the Internet Movie Database. These are split into 25,000 reviews for training and 25,000 reviews for testing. The training and testing sets are balanced, meaning they contain an equal number of positive and negative reviews.

This notebook uses tf.keras, a high-level API to build and train models in TensorFlow. For a more advanced text classification tutorial using tf.keras, see the MLCC Text Classification Guide.
Summary

• Natural Language Processing (NLP)
• Text Analytics and Text Mining
• Natural Language Processing with Python
References

- Christopher D. Manning and Hinrich Schütze (1999), Foundations of Statistical Natural Language Processing, The MIT Press.
- Rajesh Arumugam (2018), Hands-On Natural Language Processing with Python: A practical guide to applying deep learning architectures to your NLP applications, Packt.
Q & A

Natural Language Processing (NLP)

(自然語言處理)

Time: 2018/11/23 (Fri) (13:10-15:00)
Place: 淡江大學商管學院 B206
Host: 鄭啟斌 教授 (淡江大學資管系碩士班 人工智能 課程)

Min-Yuh Day
戴敏育
Assistant Professor
專任助理教授

Dept. of Information Management, Tamkang University
淡江大學 資訊管理學系

http://mail.tku.edu.tw/myday/
2018-11-23