Practices of Business Intelligence

自然語言處理
(Natural Language Processing)

1071BI11
MI4 (M2084) (2888)
Wed, 7, 8 (14:10-16:00) (B217)

Min-Yuh Day
Assistant Professor
Dept. of Information Management, Tamkang University

http://mail.tku.edu.tw/myday/
2018-12-12
<table>
<thead>
<tr>
<th>週次 (Week)</th>
<th>日期 (Date)</th>
<th>內容 (Subject/Topics)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2018/09/12</td>
<td>商業智慧實務課程介紹 (Course Orientation for Practices of Business Intelligence)</td>
</tr>
<tr>
<td>2</td>
<td>2018/09/19</td>
<td>商業智慧、分析與資料科學 (Business Intelligence, Analytics, and Data Science)</td>
</tr>
<tr>
<td>3</td>
<td>2018/09/26</td>
<td>人工智慧、大數據與雲端運算 (ABC: AI, Big Data, and Cloud Computing)</td>
</tr>
<tr>
<td>4</td>
<td>2018/10/03</td>
<td>描述性分析I：數據的性質、統計模型與可視化 (Descriptive Analytics I: Nature of Data, Statistical Modeling, and Visualization)</td>
</tr>
<tr>
<td>5</td>
<td>2018/10/10</td>
<td>國慶紀念日 (放假一天) (National Day) (Day off)</td>
</tr>
<tr>
<td>6</td>
<td>2018/10/17</td>
<td>描述性分析II：商業智慧與資料倉儲 (Descriptive Analytics II: Business Intelligence and Data Warehousing)</td>
</tr>
</tbody>
</table>
課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)
7 2018/10/24 預測性分析 I：資料探勘流程、方法與演算法
   (Predictive Analytics I: Data Mining Process, Methods, and Algorithms)
8 2018/10/31 預測性分析 II：文本、網路與社群媒體分析
   (Predictive Analytics II: Text, Web, and Social Media Analytics)
9 2018/11/07 期中報告 (Midterm Project Report)
10 2018/11/14 期中考試 (Midterm Exam)
11 2018/11/21 處方性分析：最佳化與模擬
   (Prescriptive Analytics: Optimization and Simulation)
12 2018/11/28 社會網絡分析
   (Social Network Analysis)
週次 (Week) 日期 (Date) 內容 (Subject/Topics)
13 2018/12/05 機器學習與深度學習
   (Machine Learning and Deep Learning)
14 2018/12/12 自然語言處理
   (Natural Language Processing)
15 2018/12/19 AI交談機器人與對話式商務
   (AI Chatbots and Conversational Commerce)
16 2018/12/26 商業分析的未來趨勢、隱私與管理考量
   (Future Trends, Privacy and Managerial Considerations in Analytics)
17 2019/01/02 期末報告 (Final Project Presentation)
18 2019/01/09 期末考試 (Final Exam)
Business Intelligence (BI)

1. Introduction to BI and Data Science
2. Descriptive Analytics
3. Predictive Analytics
4. Prescriptive Analytics
5. Big Data Analytics
6. Future Trends
Natural Language Processing (NLP)
Outline

• Natural Language Processing (NLP)
• NLP Libraries and Tools
• NLP and Text Analytics with Python
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. ... ”

Example of Opinion: review segment on iPhone

“(1) I bought an iPhone a few days ago.
(2) It was such a nice phone.
(3) The touch screen 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. ...”

Text Analytics and Text Mining

Text Mining “Knowledge Discovery in Textual Data”

- Document Matching
- Link Analysis
- Information Retrieval
- Search Engines
- POS Tagging
- Lemmatization
- Word Disambiguation

Text Analytics

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

- Data Mining
  - Classification
  - Clustering
  - Association

- Natural Language Processing

Statistics

Machine Learning

Management Science

Artificial Intelligence

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.
Text Mining Technologies

- Statistics
- Database Systems
- Natural Language Processing
- Information Retrieval
- Machine Learning
- Applications
- Pattern Recognition
- Visualization
- Algorithms
- High-performance Computing

Adapted from: Jiawei Han and Micheline Kamber (2011), Data Mining: Concepts and Techniques, Third Edition, Elsevier
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

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

- word’s stem: am → am
- word’s lemma: am → be
- having → hav
- having → have
Text Summarization

**Topic Modeling**

---

** 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 Andersson 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 Araceli Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

---


---

**Source:** Blei, David M. "Probabilistic topic models." Communications of the ACM 55, no. 4 (2012): 77-84.
Natural Language Processing (NLP)

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

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

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

Classical NLP

Deep Learning-based NLP

Dense Embeddings

obtained via word2vec, doc2vec, GloVe, etc.

Hidden Layers

Output Units

Documents

Language Detection

Documents

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

[Diagram showing the Modern NLP Pipeline]

Documents -> Language Detection -> Pre-processing

EN: Tokenize -> POS Tagging -> ... -> Token Filtering
CN: Tokenize -> POS Tagging -> ... -> Token Filtering

Pre-processed Documents

Pre-processed Documents -> Build Vocabulary

Bag-of-Words & Vectorization -> Machine Learning

Word Embeddings: word2vec, doc2vec, GloVe

(Deep) Neural Network

Task / Output

Classification
Sentiment Analysis
Entity Extraction
Topic Modeling
Similarity

Modern NLP Pipeline

Documents → Language Detection → Preprocessing → Modeling

EN: Preprocessing → Modeling
ZH: Preprocessing → Preprocessing → Modeling

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

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

Documents → 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

Jacob Devlin    Ming-Wei Chang    Kenton Lee    Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

BERT

Bidirectional Encoder Representations from Transformers

Pre-training model architectures

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

(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&lt;sub&gt;BASE&lt;/sub&gt;</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&lt;sub&gt;LARGE&lt;/sub&gt;</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

An example of Text Mining

Analyze Text
- Information Extraction
- Classification
- Summarization
- Clustering

Retrieve and preprocess document

Document Collection

Overview of Information Extraction based Text Mining Framework

The Three-Step/Task Text Mining Process

**Task 1: Establish the Corpus**
- Collect and organize the domain-specific unstructured data.
- The inputs to the process include a variety of relevant unstructured (and semi-structured) data sources such as text, XML, HTML, etc.

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

**Task 3: Extract Knowledge**
- Discover novel patterns from the T-D matrix.
- The output of Task 3 is a number of problem-specific classification, association, clustering models and visualizations.

## Term–Document Matrix

| Terms | Investment Risk | Project Management | Software Engineering | Development | SAP | ...
|-------|----------------|-------------------|---------------------|-------------|-----|------
| Documents |     |                   |                     |             |     |      |
| Document 1 | 1  |                   |                     | 1           |     |      |
| Document 2 |     | 1                 |                     |             |     |      |
| Document 3 |     |                   | 3                   |             |     |      |
| Document 4 |     | 1                 |                     |             |     |      |
| Document 5 |     |                   |                     | 2           | 1   |      |
| Document 6 | 1  |                   |                     |             |     | 1    |
| ...       |     |                   |                     |             |     |      |

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

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/
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 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'), ('title', 'NN'), ('of', 'IN'), ...]
blob.noun_phrases # WordList(['title', '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
- Documentation: [http://polyglot.readthedocs.org](http://polyglot.readthedocs.org)

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/](https://polyglot.readthedocs.io/)
scikit-learn

Home  Installation  Documentation  Examples

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, ...
— Examples

Regression
Predicting a continuous-valued attribute associated with an object.
Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ...
— Examples

Clustering
Automatic grouping of similar objects into sets.
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ...
— Examples

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.

Part-of-Speech:

```
1 Stanford NNP University NNP is VBZ located JJ in IN California NNP .
PRP VBZ DT JJ NN .
2 It is a great NN university .
```
Stanford University is located in California. It is a great university.

Named Entity Recognition:

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 University is located in California. It is a great university.
Collapsed dependencies:

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

Collapsed CC-processed dependencies:

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

Visualisation provided using the brat visualisation/annotation software.

Copyright © 2011, Stanford University, All Rights Reserved.

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

### Stanford CoreNLP XML Output

#### Document

#### Document Info

#### Sentences

**Sentence #1**

<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></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></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></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></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></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></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></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)))))) (.) )
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.

<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>PERO</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>PERO</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>PERO</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>PERO</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>PERO</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>.</td>
<td>70</td>
<td>71</td>
<td>.</td>
<td>O</td>
<td></td>
<td>PERO</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>Sentence</th>
<th>Head</th>
<th>Text</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 (gov)</td>
<td>Stanford University</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>It</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>a great university</td>
<td></td>
</tr>
</tbody>
</table>
Stanford University is located in California. It is a great university.

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

Copyright © 2011, Stanford University, All Rights Reserved.
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. Microsoft, 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.

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

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

Stanford Named Entity Tagger

Classifier: english.muc.7class.distsim.crf.ser.gz

Output Format: slashTags

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.

Submit Clear

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

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

Stanford Named Entity Tagger

Classifier: english.all.3class.distsim.crfsurfer.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 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

Copyright © 2011, Stanford University, All Rights Reserved.
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

Classifier: english.muc.7class.distsim.crf.ser.gz

Classifier: english.all.3class.distsim.crf.ser.gz
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 reclaimed 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 Sheridan @CNNTech May 2, 2014 5:46 PM ET

NEW YORK - LRB CNNMoney RRB 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 - LRB OMSFT RRB, 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.
CKIP 中研院中文斷詞系統
http://ckipsvr.iis.sinica.edu.tw/

歐巴馬是美國的一位總統

文章的文字檔
摘取未知詞過程
包含未知詞的斷詞標記結果
未知詞列表

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。

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。

http://udn.com/news/story/7323/1653437-%E8%8E%8E%E5%A3%AB%E6%AF%94%E4%BA%9E%E5%9C%A8%E6%B7%A1%E6%B1%9F-%E9%81%87%E8%A6%8B%E8%B3%BD%E8%90%AC%E6%8F%90%E6%96%AF
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「唐吉訏德」作者賽萬提斯逝世之日。英專起家の淡江大學舉辦「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雅表示，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。

日文系學生賴家郁說，原本只是喜歡「打籃球」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。
中文斷詞系統

http://ckipsvr.iis.sinica.edu.tw/
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。
中文斷詞系統

• CKIP
• Jieba
• THULAC
• CoreNLP
• pyLTP
• SnowNLP
• PyNLPIR
• HanLP
Vector Representations of Words

Word Embeddings

Word2Vec

GloVe
Modern NLP Pipeline

Pre-processing

Documents

Language Detection

EN

CN

Tokenize

POS Tagging

... (Continues)

Token Filtering

Pre-processed Documents

Documents

Build Vocabulary

Pre-processed Documents

Bag-of-Words & Vectorization

Word Embeddings

- word2vec
- doc2vec
- GloVe

Machine Learning

(Deep) Neural Network

Task / Output

- Classification
- Sentiment Analysis
- Entity Extraction
- Topic Modeling
- Similarity

Facebook Research FastText

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

Facebook Research 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/
# 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/
# 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>
NLP and Text Analytics with Python
Python in Google Colab

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

```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.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)
```
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_k: 1K, 2K, ... 15K, 20K, 25K, ... 90K, all)

Normalization mode

samplewise None featurewise

Embeddings

S/W < 15K

Yes

Fine-tuned pre-trained embedding

Frozen pre-trained embedding

Embeddings learned from scratch

No

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

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' =

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The</strong></td>
<td>1</td>
<td>[0, 1, 0, 0, 0, 0, 0, 0],</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>mouse</strong></td>
<td>2</td>
<td>[0, 0, 1, 0, 0, 0, 0, 0],</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ran</strong></td>
<td>3</td>
<td>[0, 0, 0, 1, 0, 0, 0, 0],</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>up</strong></td>
<td>4</td>
<td>[0, 0, 0, 0, 1, 0, 0, 0],</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>the</strong></td>
<td>1</td>
<td>[0, 1, 0, 0, 0, 0, 0, 0],</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>clock</strong></td>
<td>5</td>
<td>[0, 0, 0, 0, 0, 0, 1, 0] ]</td>
<td></td>
<td></td>
<td></td>
<td></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 mouse ran down

The mouse ran up the clock

Word embeddings

The
mouse
ran
up
clock
down

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

[[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]]
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 ]
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_words: 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

https://colab.research.google.com/drive/1FEG6DnGwfwUbeo4zJ1zTunjMqf2RkCrT
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

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)

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
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
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='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
http://www.nltk.org/
```
## Python Jieba

“结巴”中文分词

[GitHub, Inc. (US)](https://github.com/fxsjy/jieba)

**fxsjy / jieba**

<table>
<thead>
<tr>
<th>Branch</th>
<th>Issues</th>
<th>Pull requests</th>
<th>Projects</th>
<th>Wiki</th>
<th>Pulse</th>
<th>Graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>master</td>
<td>226</td>
<td>14</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Code**

- **485 commits**
- **2 branches**
- **23 releases**
- **31 contributors**

**MIT**

- **fxsjy committed on GitHub** Merge pull request #382 from huntzhan/master
- **extra_dict** update to v0.33
- **jieba** Bugfix for HMM=False in parallelism.
- **test** Bugfix for HMM=False in parallelism.
- **.gitattributes** first commit
- **.gitignore** update jieba3k
- **Changelog** version change 0.38
- **LICENSE** add a license file
- **MANIFEST.in** include Changelog & README.md in the distribution package
- **README.md** Update README.md

[https://github.com/fxsjy/jieba](https://github.com/fxsjy/jieba)
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
words = jieba.cut(sentence)

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)
p = ['.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)
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
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))
wordpos = pseg.cut(sentence)
result = 
for word, pos in wordpos:
    print(word + ' (' + pos + '))'
    result = result + ' ' + word + ' (' + pos + '))'
print(result.strip())
```

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)
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
  - 结巴中文斷詞台灣繁體版本
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

Source: https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/basic_text_classification.ipynb
Summary

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

• Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson.
• Rajesh Arumugam (2018), Hands-On Natural Language Processing with Python: A practical guide to applying deep learning architectures to your NLP applications, Packt.
• Christopher D. Manning and Hinrich Schütze (1999), Foundations of Statistical Natural Language Processing, The MIT Press.
• Nitin Hardeniya (2015), NLTK Essentials, Packt.
• Bing Liu (2009), Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer.