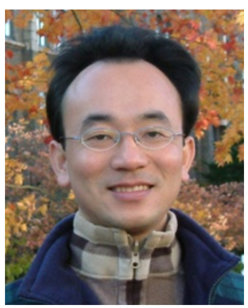
Tamkang University
淡江大學

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
淡江大學 資訊管理學系
http://mail. tku.edu.tw/myday/

2018-11-23

Outline

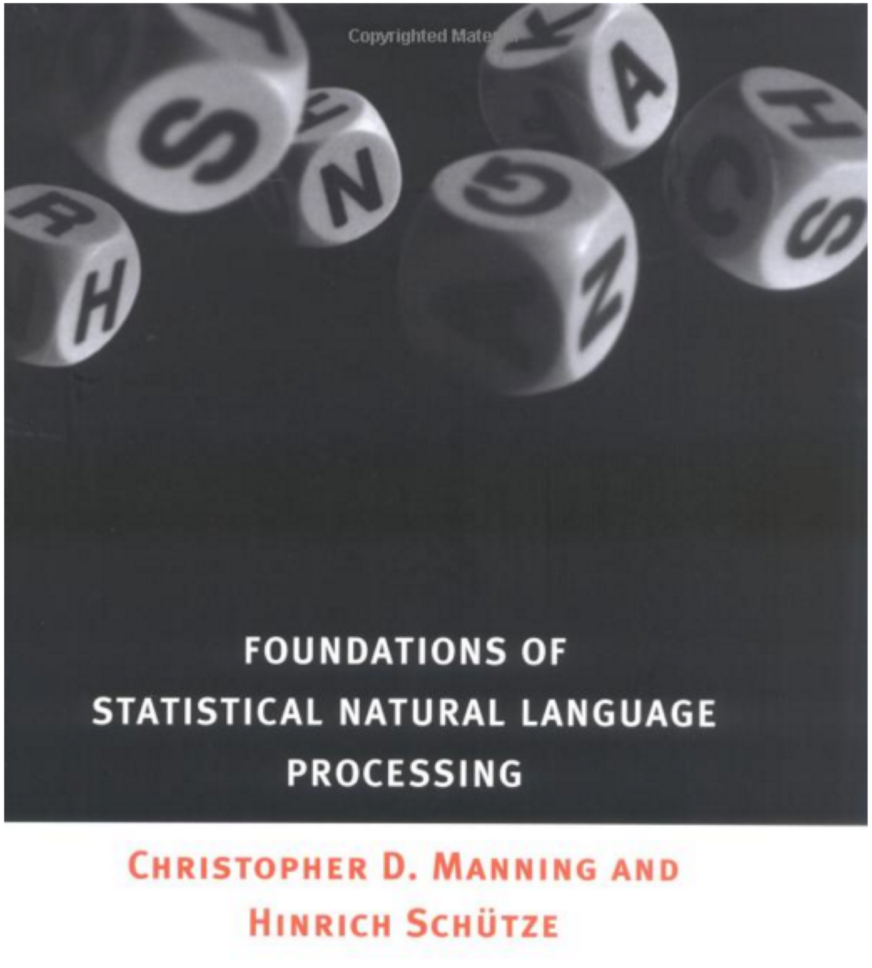
- Natural Language Processing (NLP)
- Text Analytics and Text Mining
- Natural Language Processing with Python

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

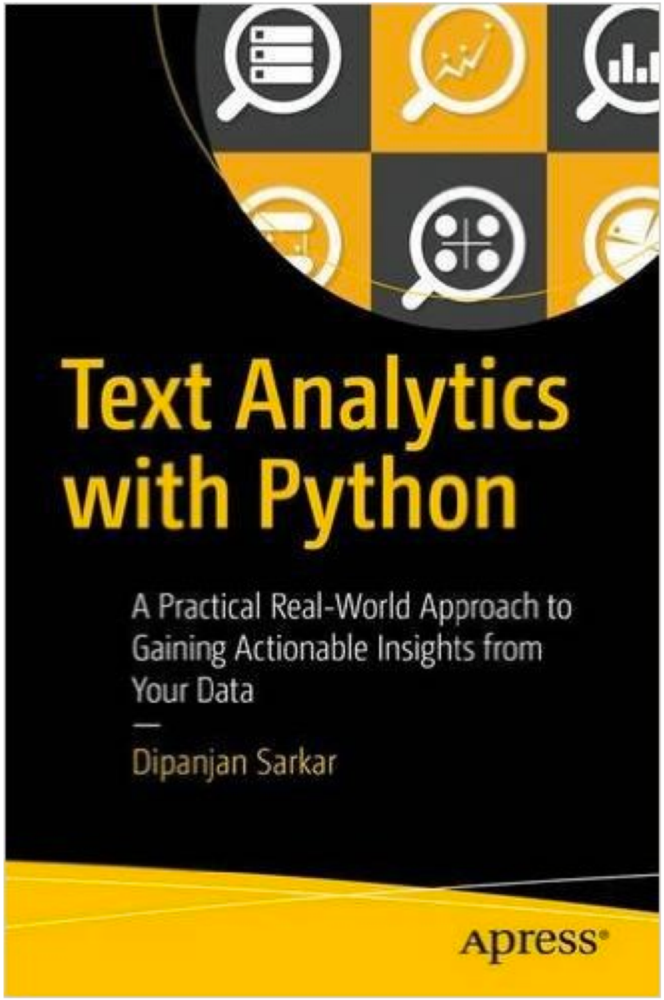


http://www.amazon.com/Foundations-Statistical-Natural-Language-Processing/dp/0262133601

Dipanjan Sarkar (2016),

Text Analytics with Python:

A Practical Real-World Approach to Gaining Actionable Insights from your Data, Apress

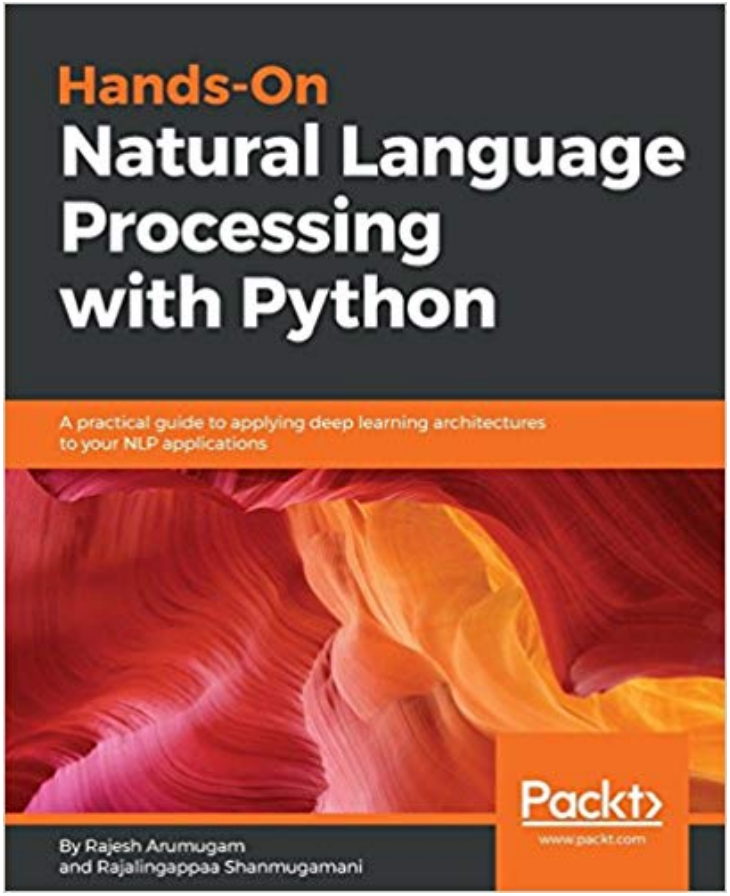


https://www.amazon.com/Text-Analytics-Python-Real-World-Actionable/dp/148422387X

Rajesh Arumugam (2018),

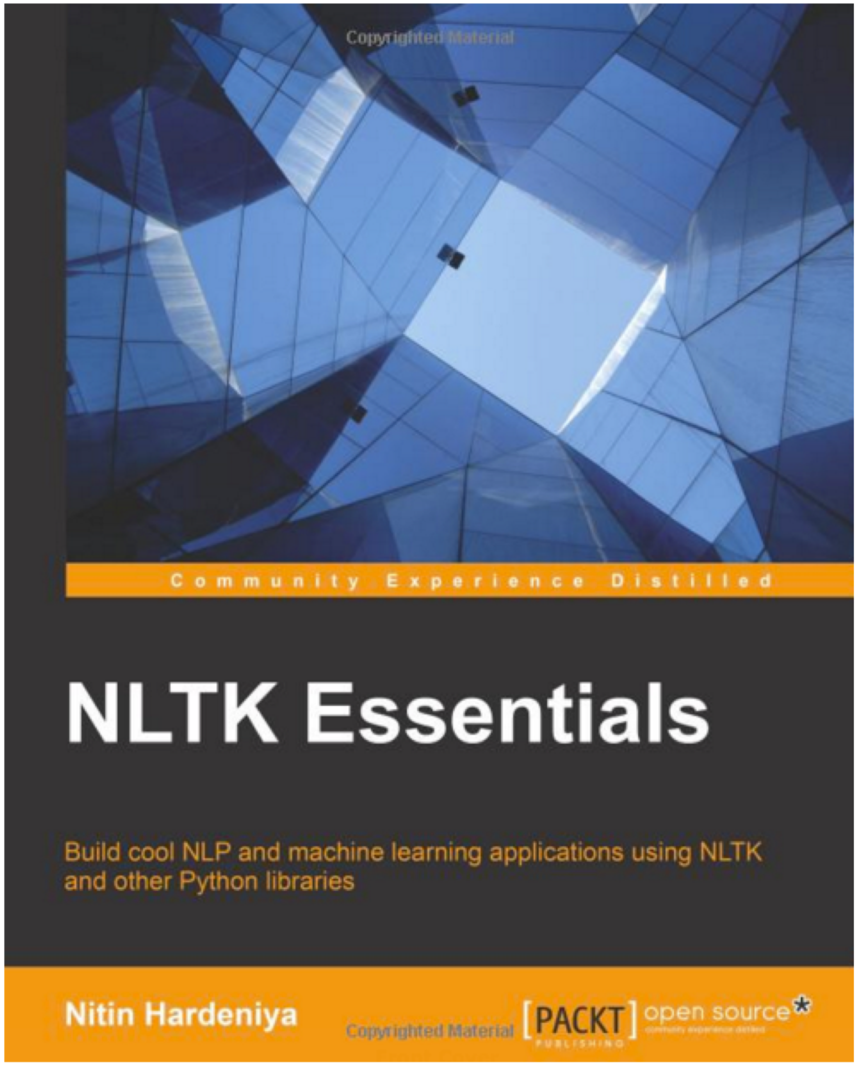
Hands-On Natural Language Processing with Python:

A practical guide to applying deep learning architectures to your NLP applications, Packt



https://www.amazon.com/Hands-Natural-Language-Processing-Python/dp/178913949X

Nitin Hardeniya (2015), NLTK Essentials, Packt Publishing

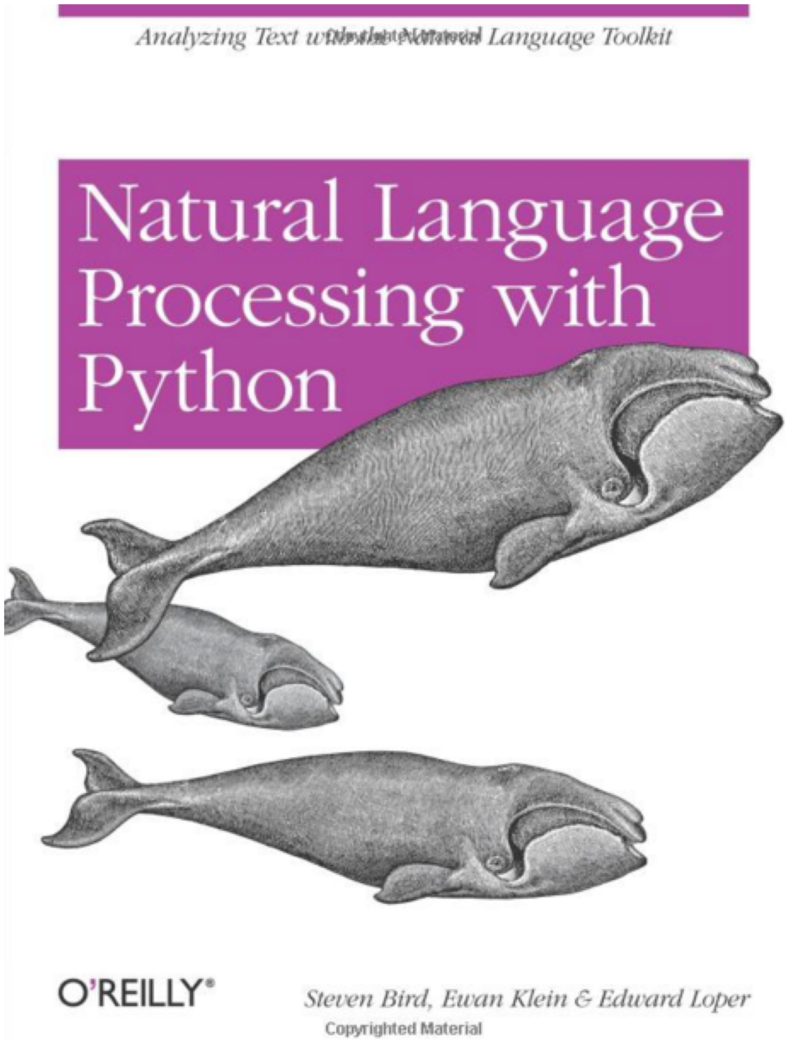


http://www.amazon.com/NLTK-Essentials-Nitin-Hardeniya/dp/1784396907

Steven Bird, Ewan Klein and Edward Loper (2009),

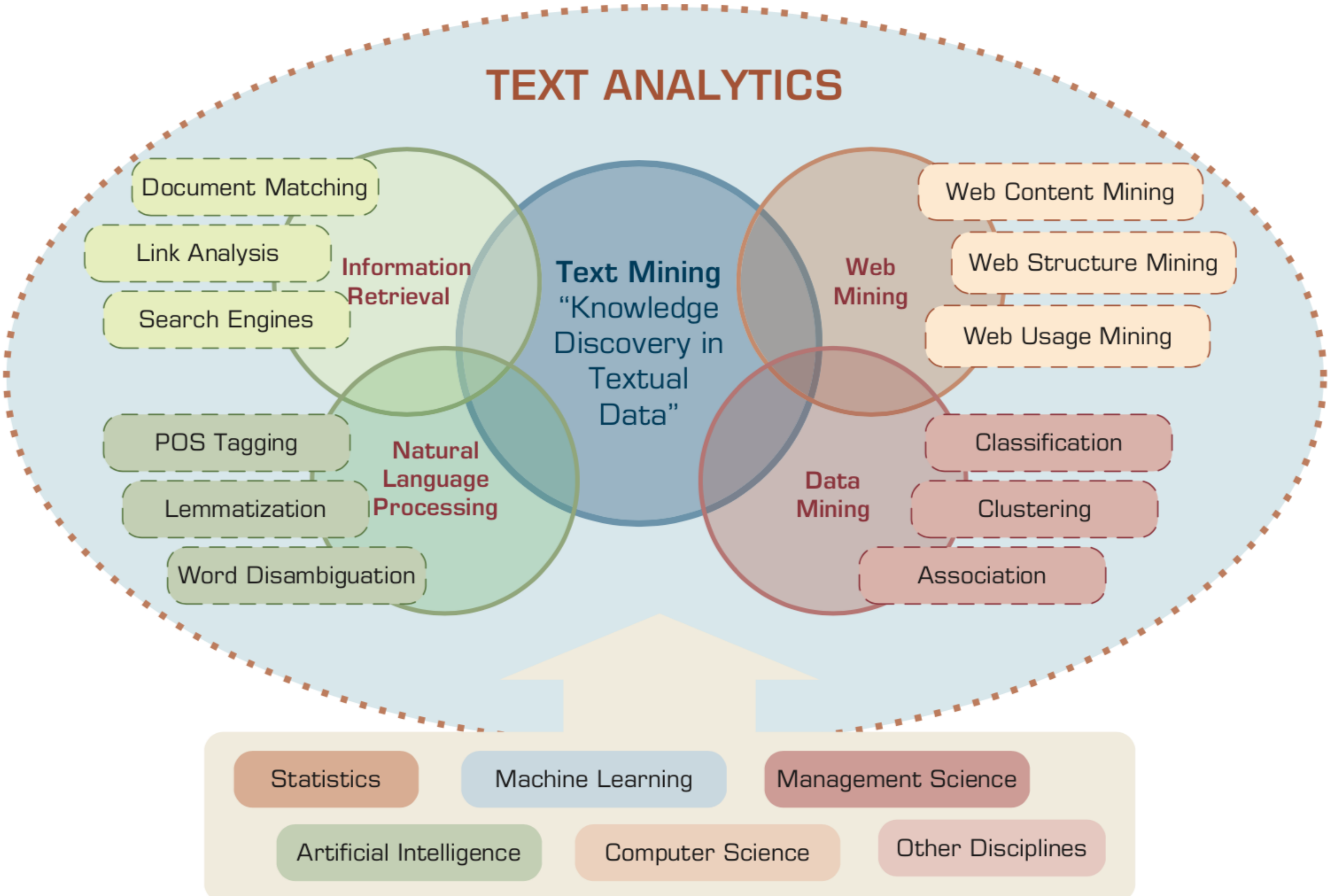
Natural Language Processing with Python,

O'Reilly Media



http://www.amazon.com/Natural-Language-Processing-Python-Steven/dp/0596516495

Text Analytics and Text Mining

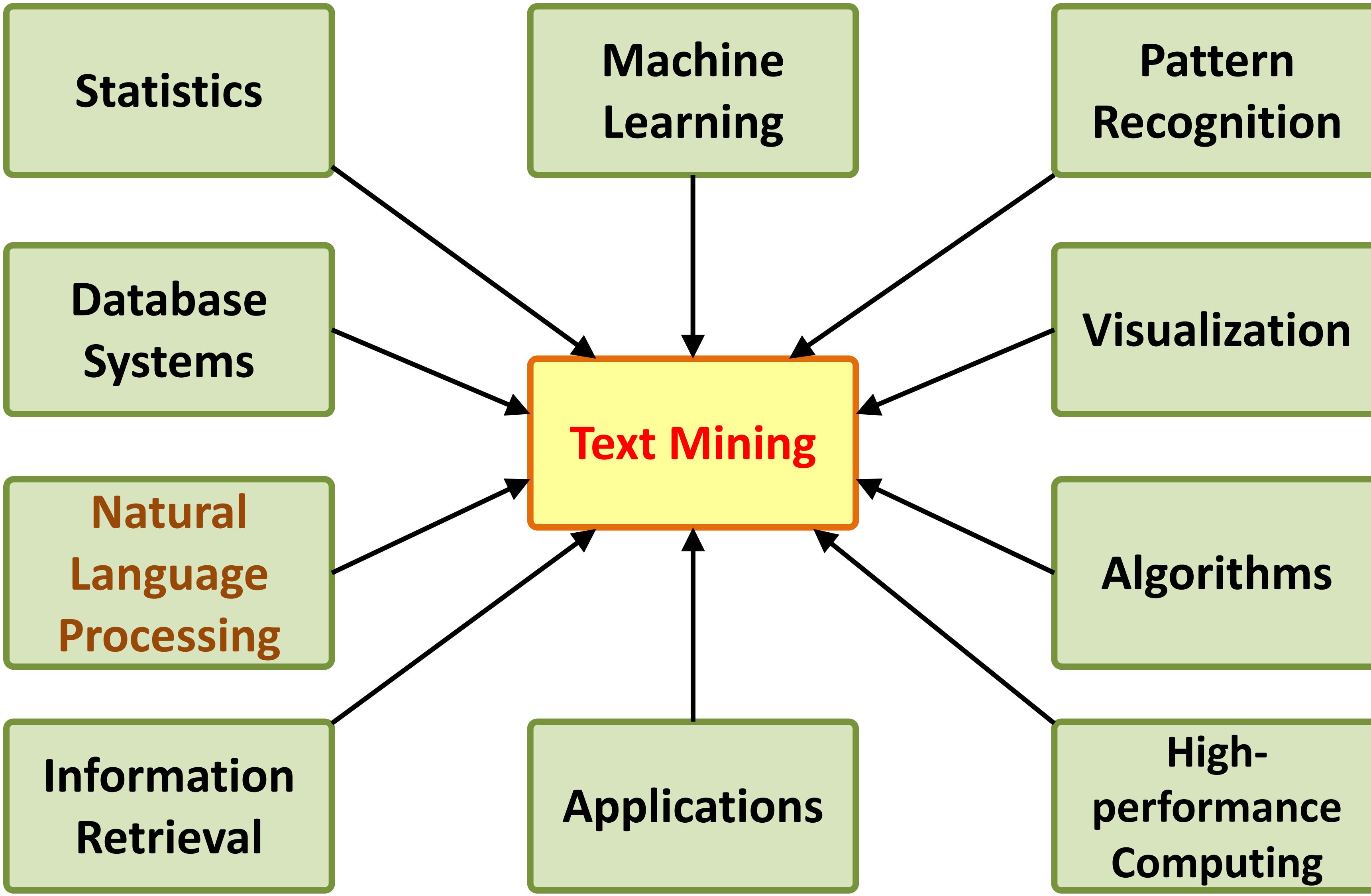


Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

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



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

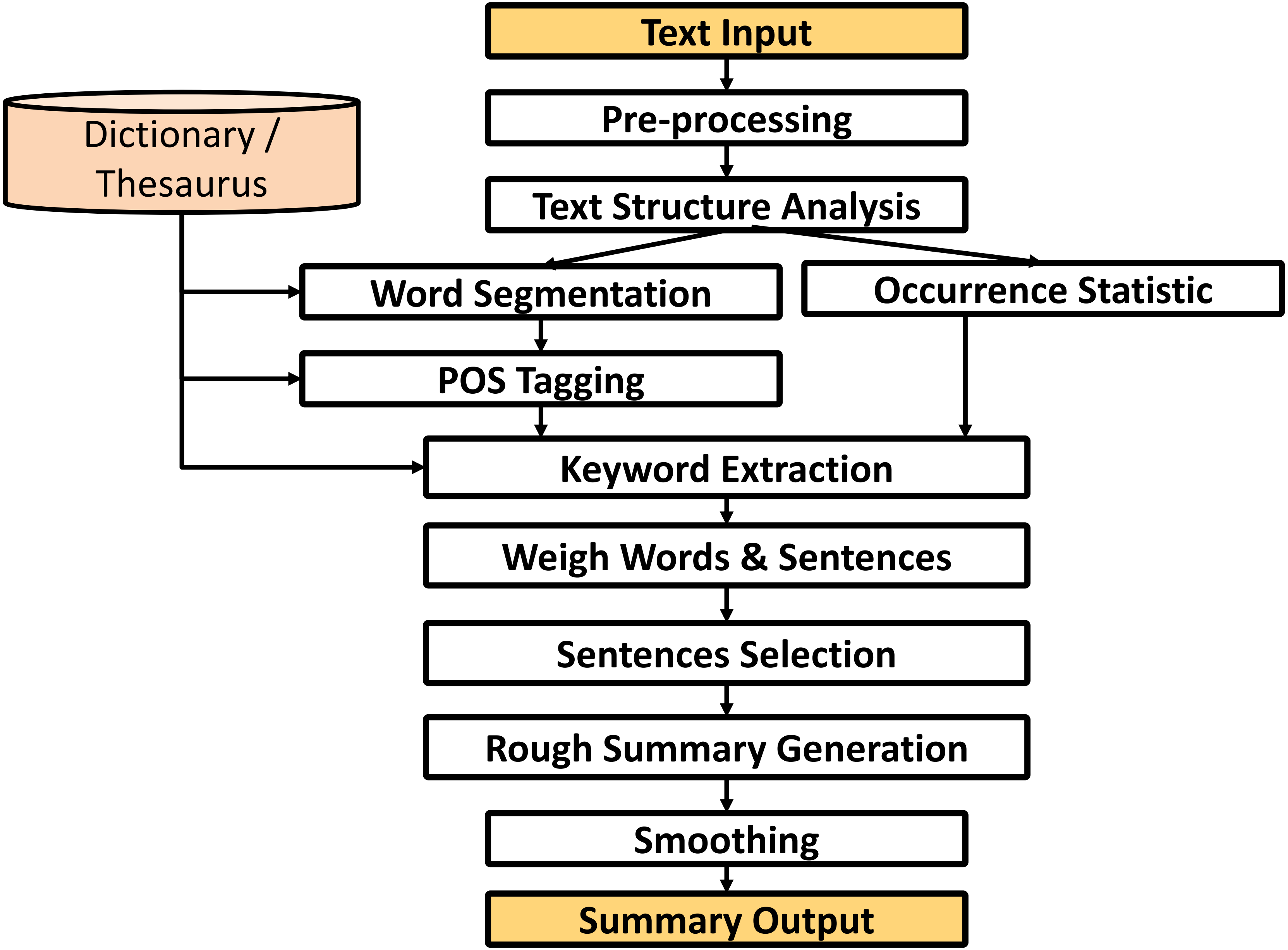
Natural Language Processing (NLP) and Text Mining

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

word's stem	word's lemma
am → am	am → be
having → hav	having → have

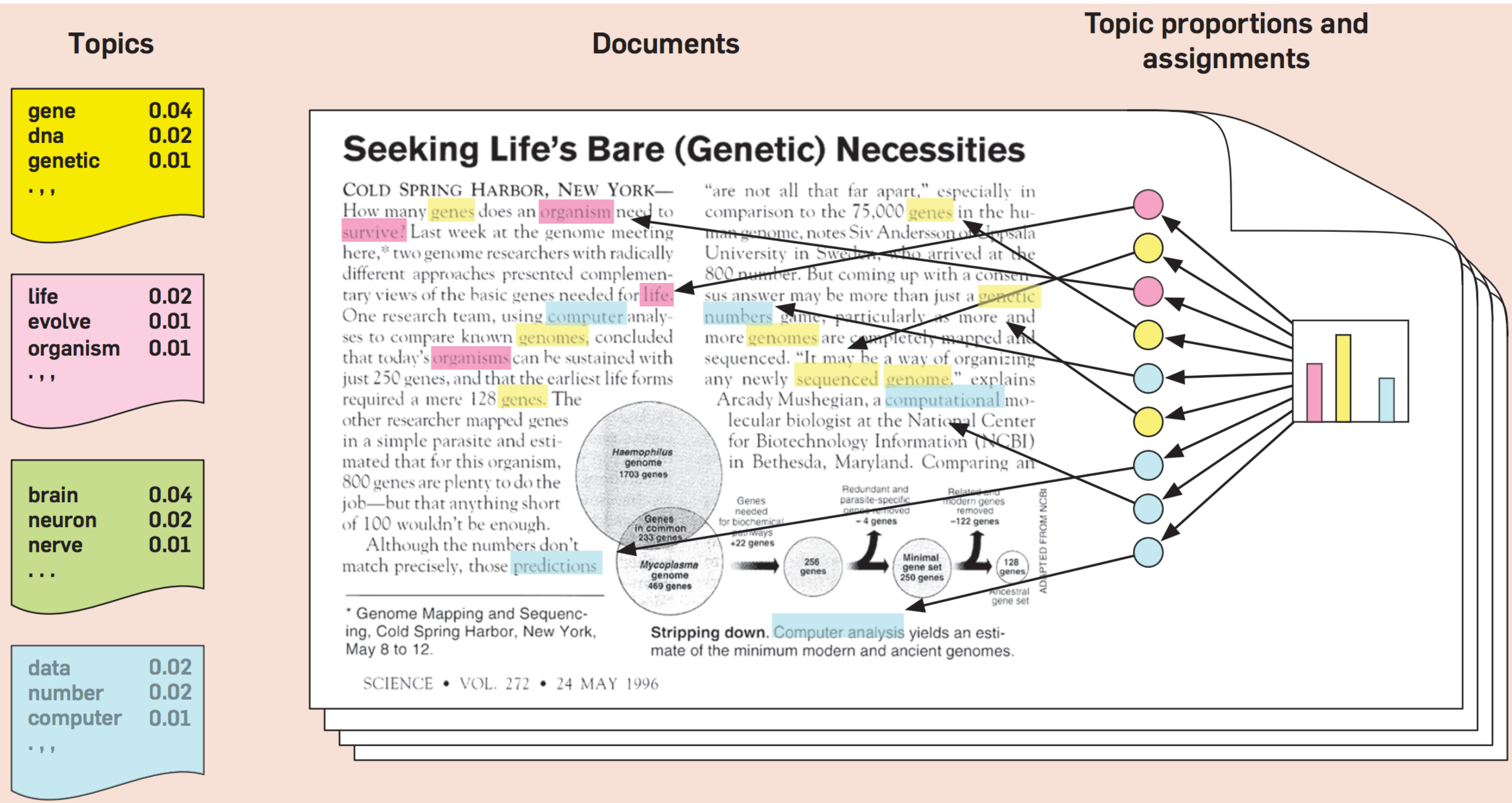
Source: Nitin Hardeniya (2015), NLTK Essentials, Packt Publishing; Florian Leitner (2015), Text mining - from Bayes rule to dependency parsing

Text Summarization



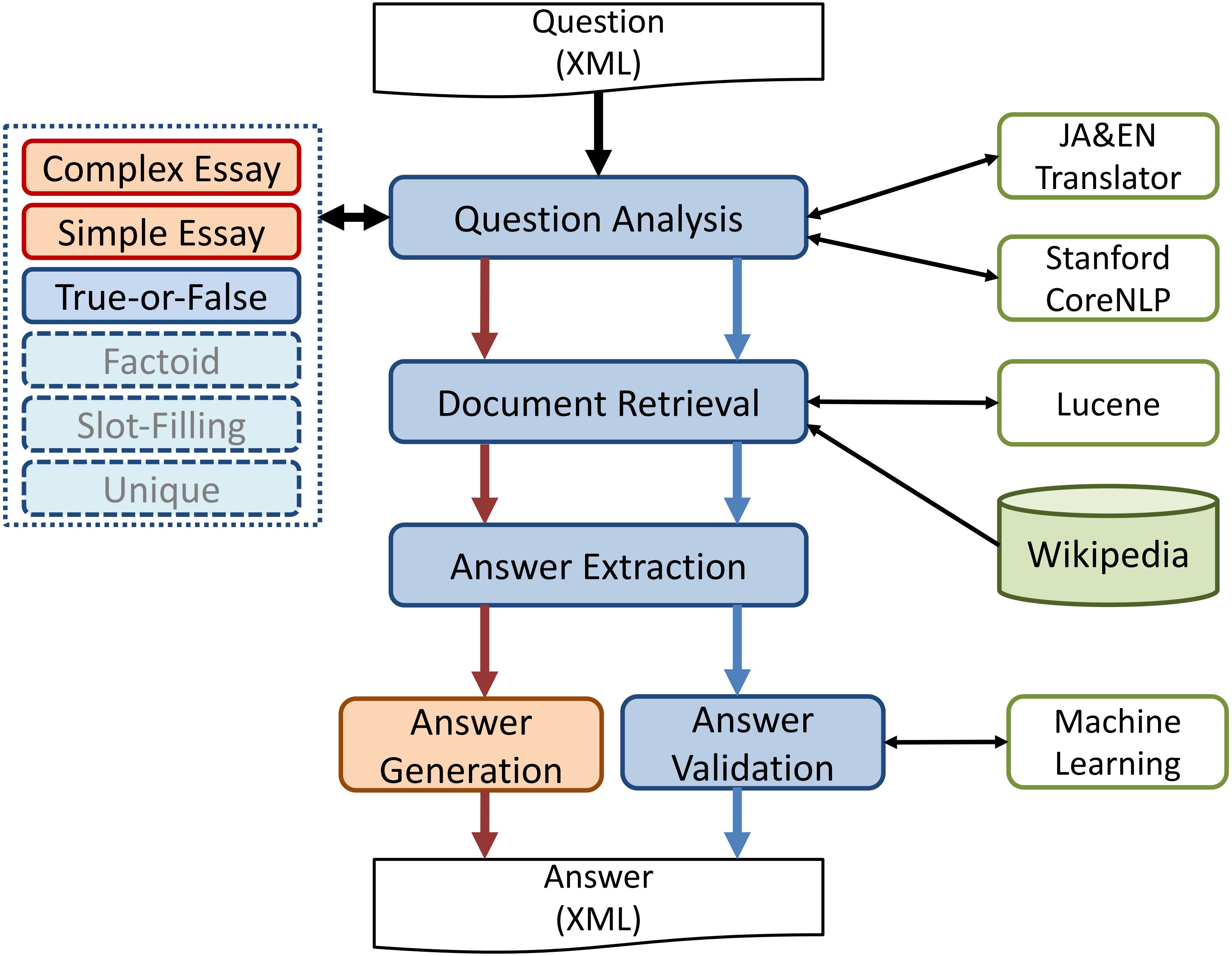
Source: Vishal Gupta and Gurpreet S. Lehal (2009), "A survey of text mining techniques and applications," Journal of emerging technologies in web intelligence, vol. 1, no. 1, pp. 60-76.

Topic Modeling



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

Question Answering System



Natural Language Processing (NLP)

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

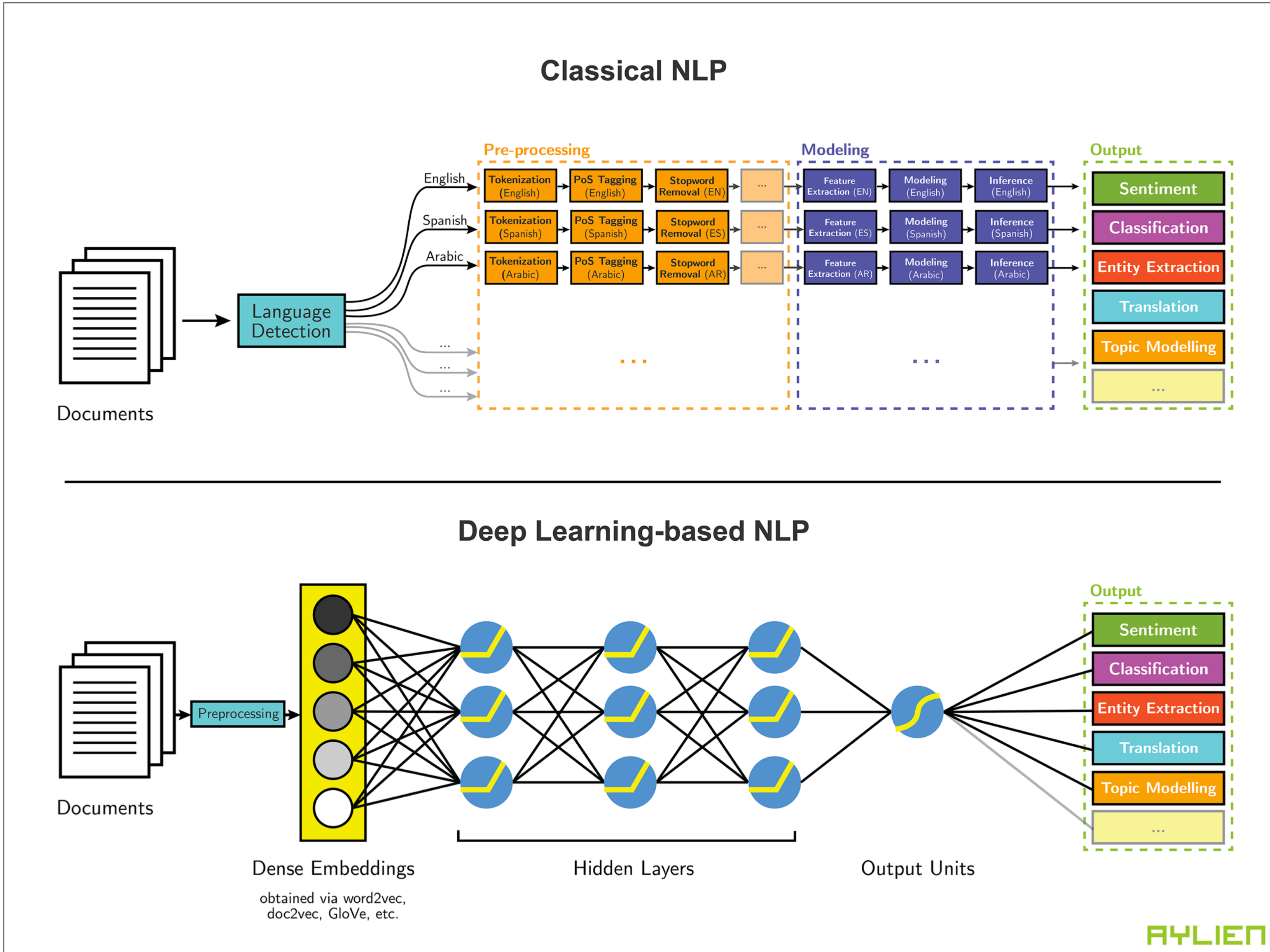
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

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

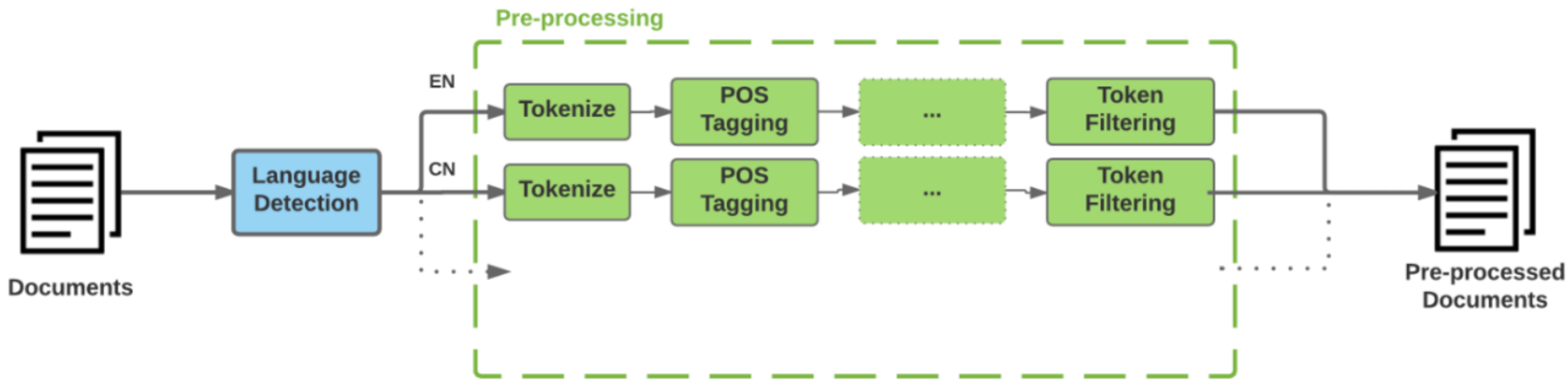
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

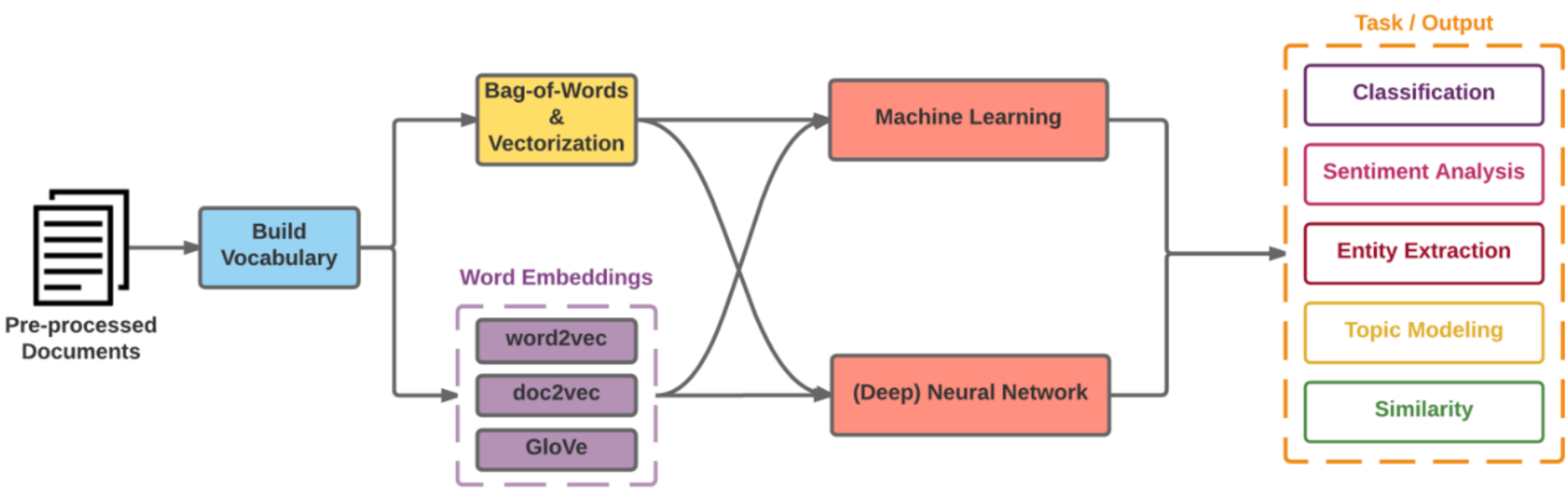
NLP



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

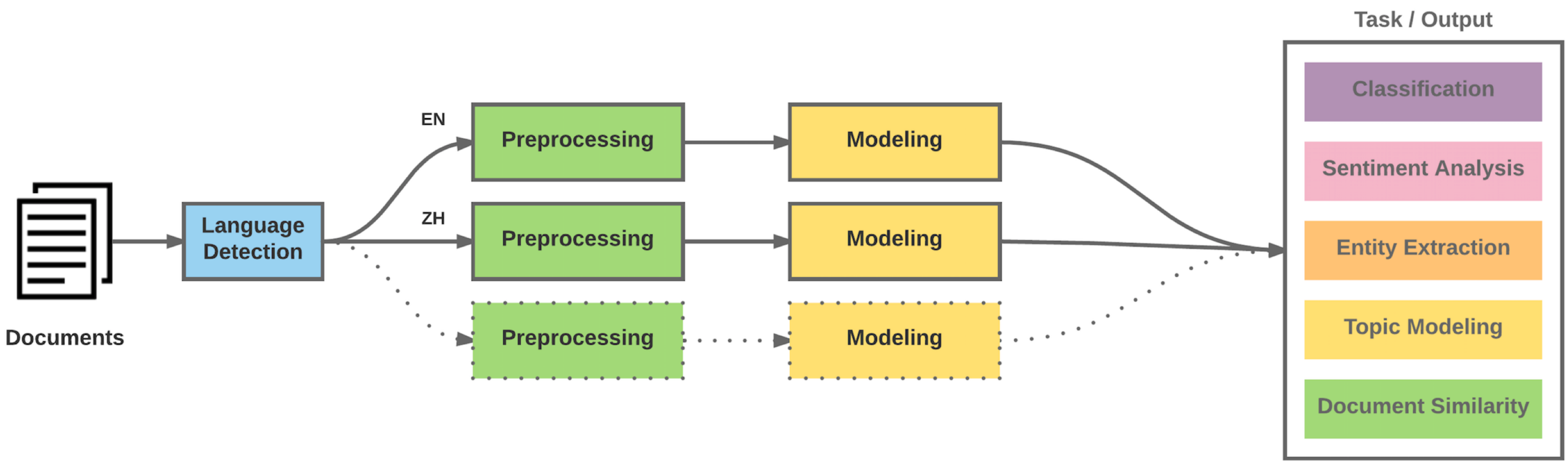
Modern NLP Pipeline





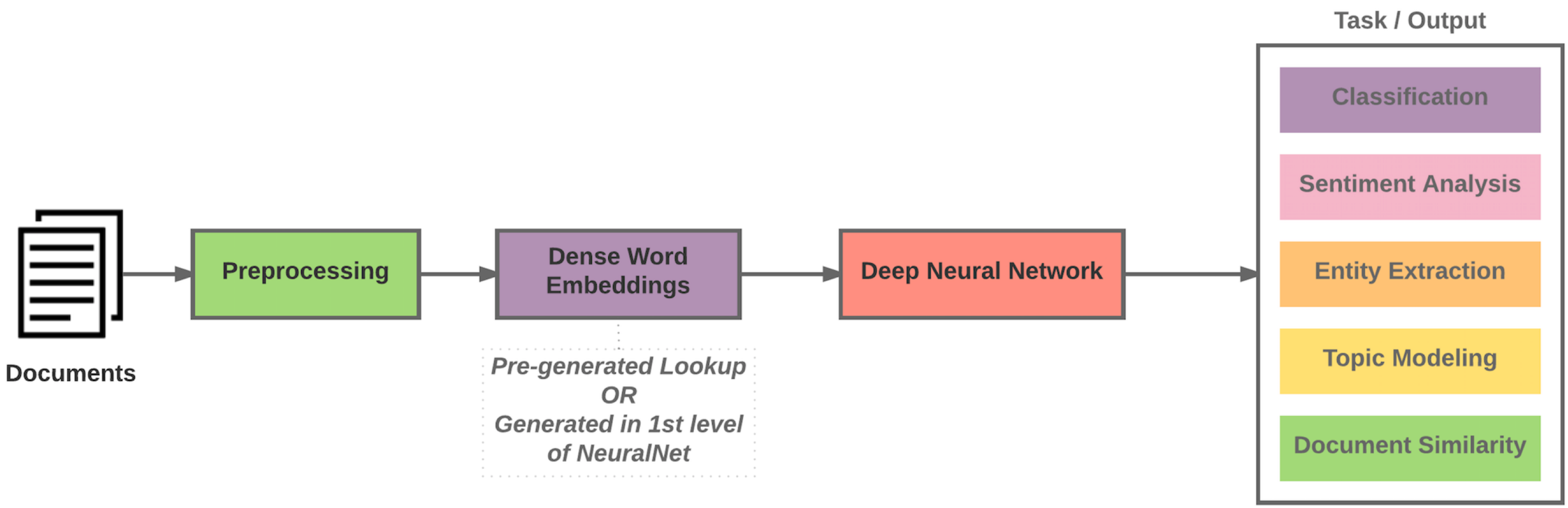
Source: https://github.com/fortiema/talks/blob/master/opendata2016sh/pragmatic-nlp-opendata2016sh.pdf

Modern NLP Pipeline



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

Deep Learning NLP



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

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

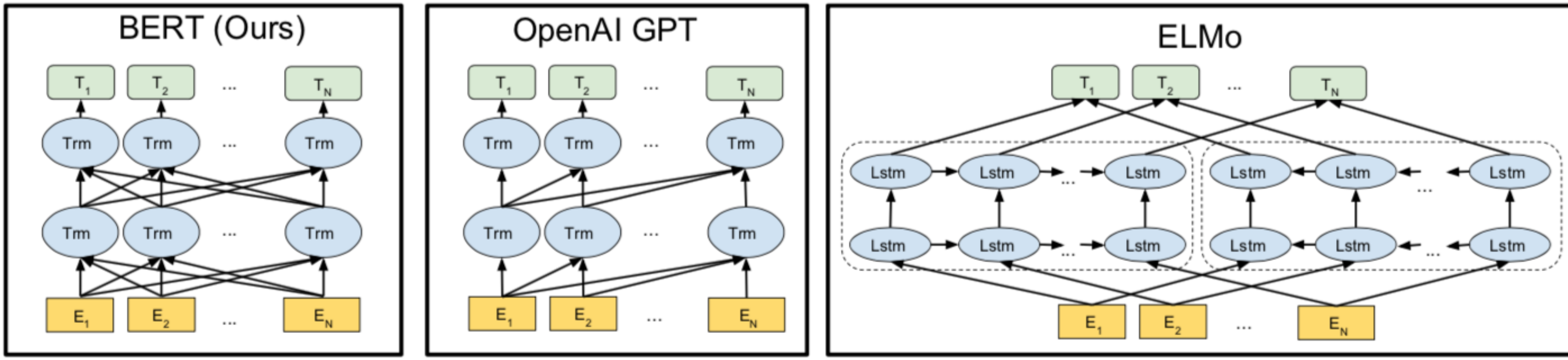
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

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

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

BERT

Bidirectional Encoder Representations from Transformers

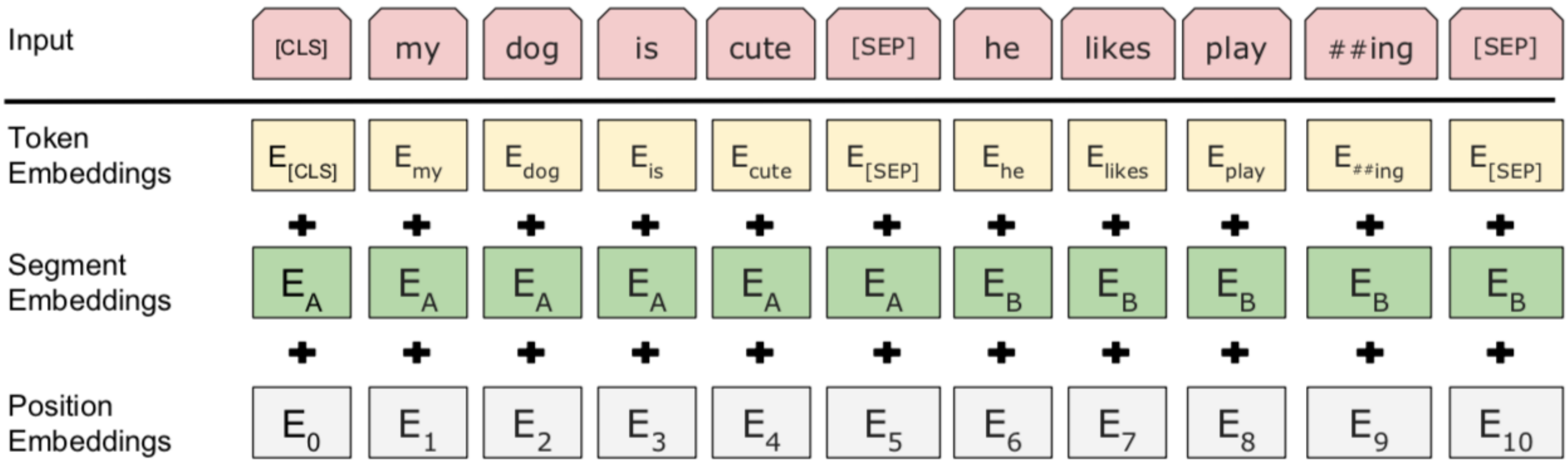


Pre-training model architectures

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.

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

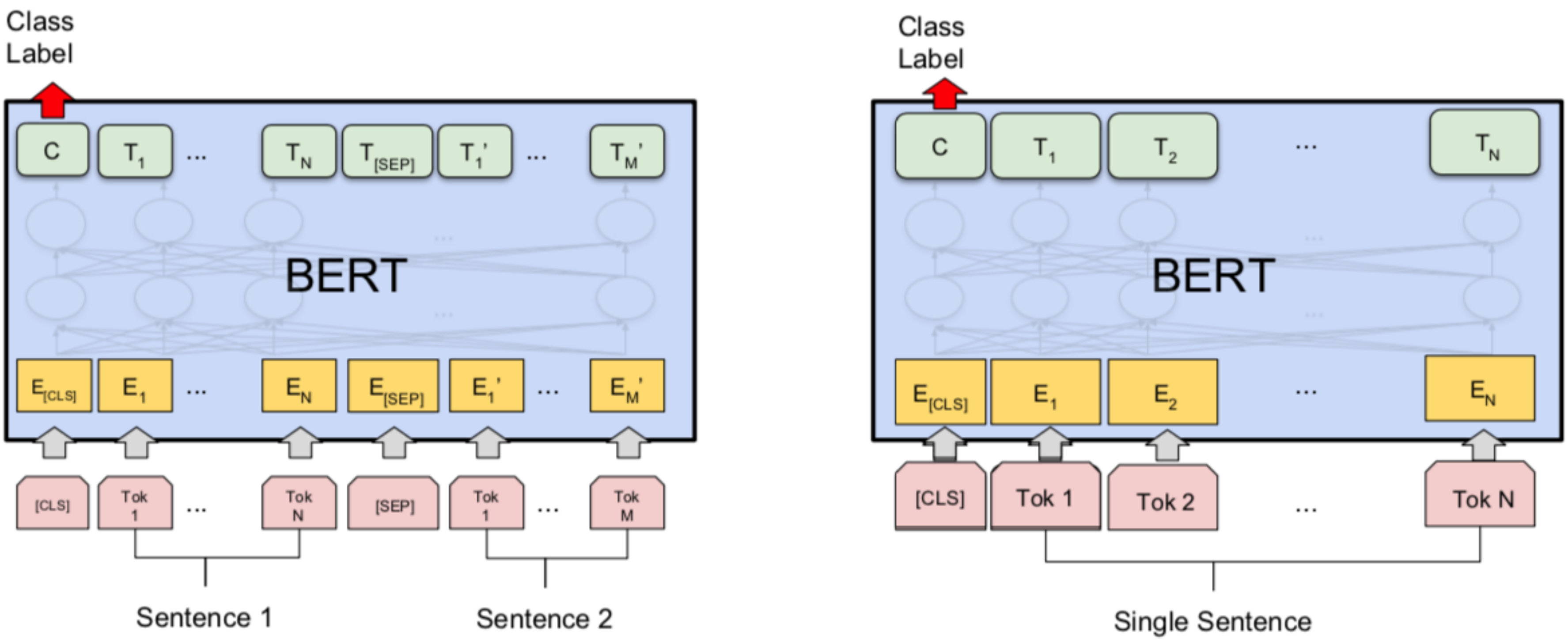
BERT input representation



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

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

BERT Sequence-level tasks

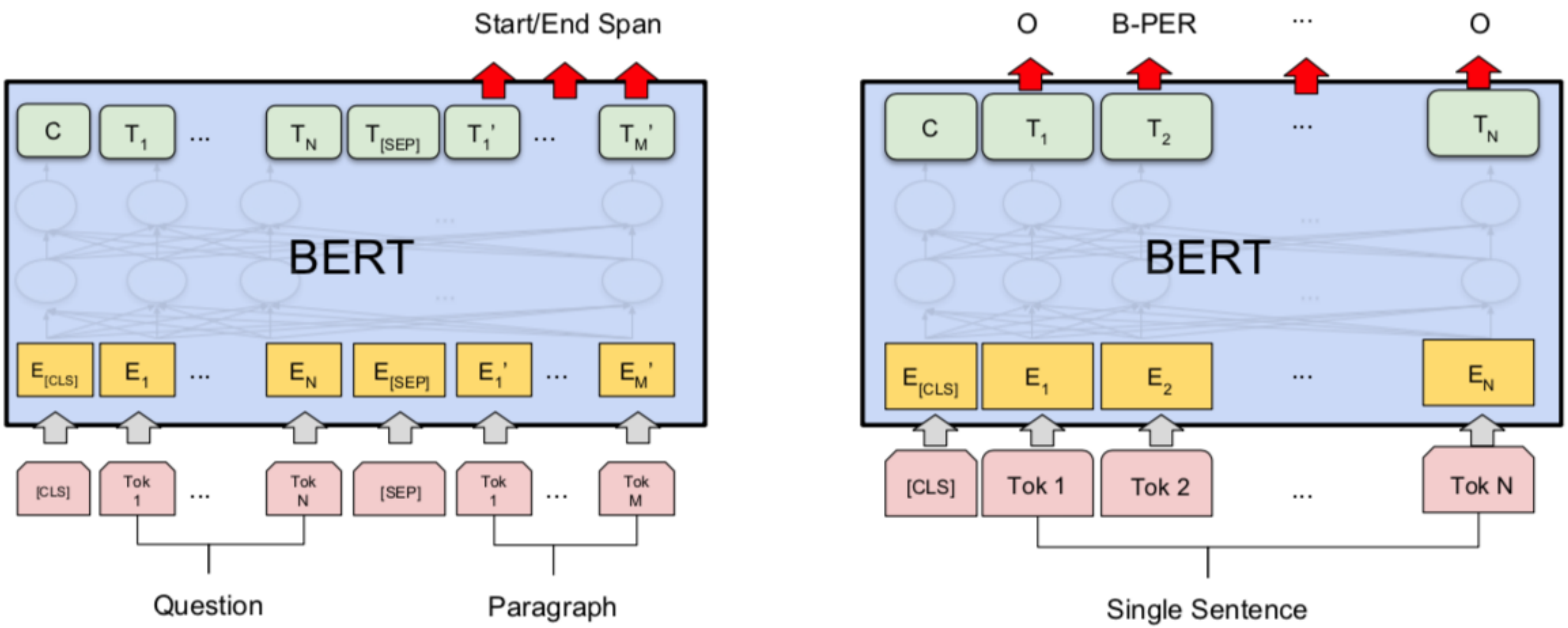


(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

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

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

BERT Token-level tasks



(c) Question Answering Tasks:
SQuAD v1.1

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

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

General Language Understanding Evaluation (GLUE) benchmark

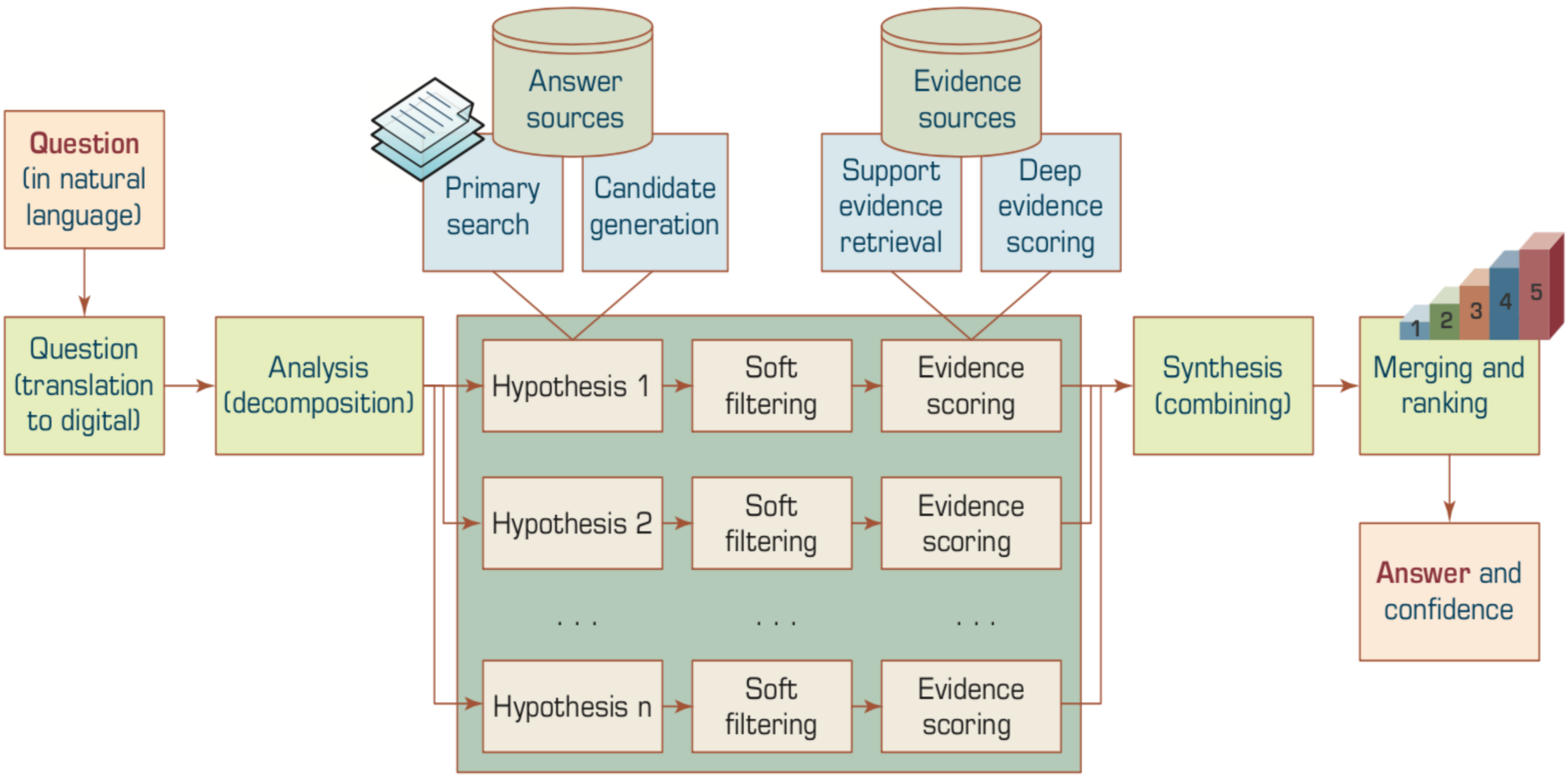
GLUE Test results

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

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

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

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

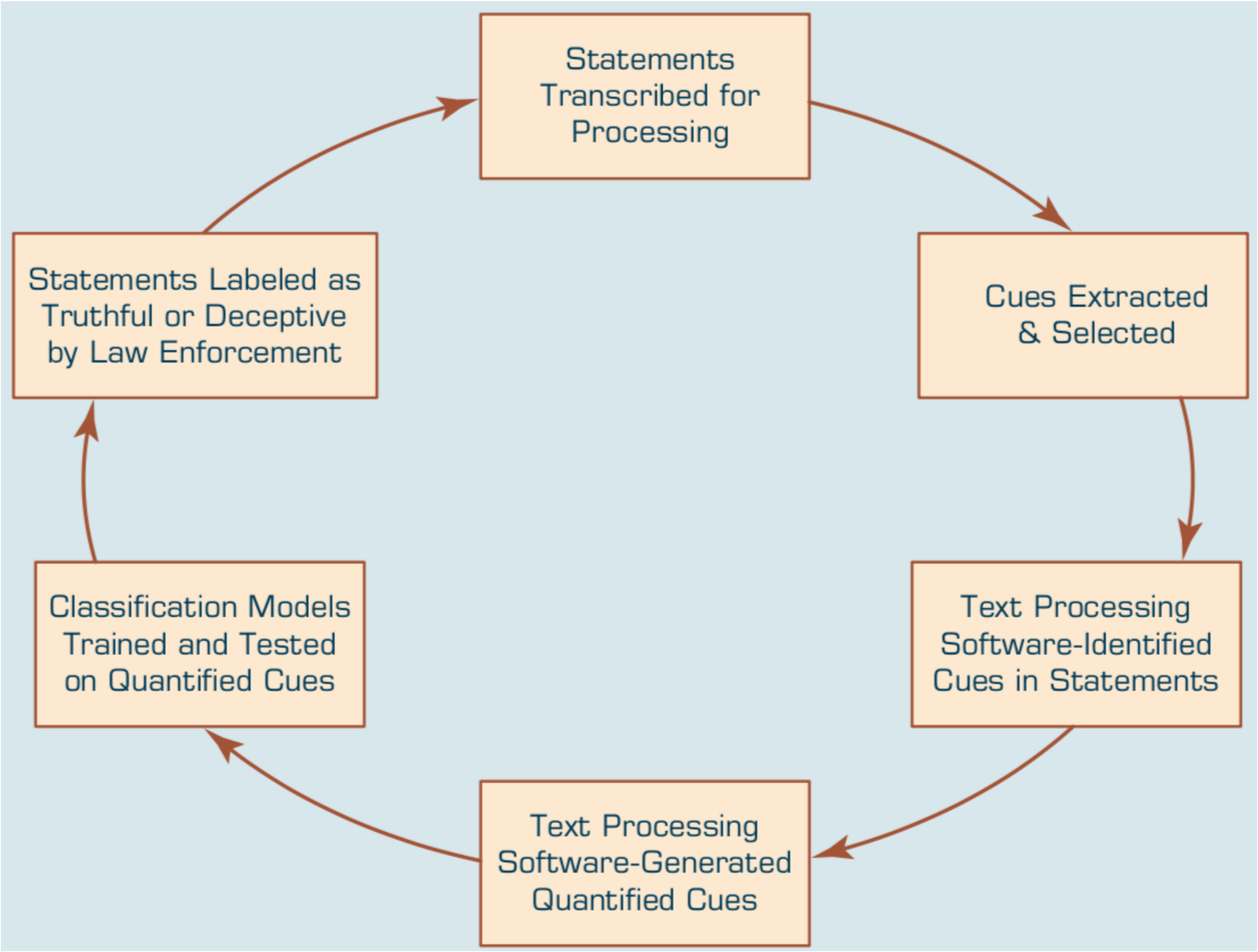
- Text Data Mining
- Knowledge Discovery in Textual Databases

Application Areas of Text Mining

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

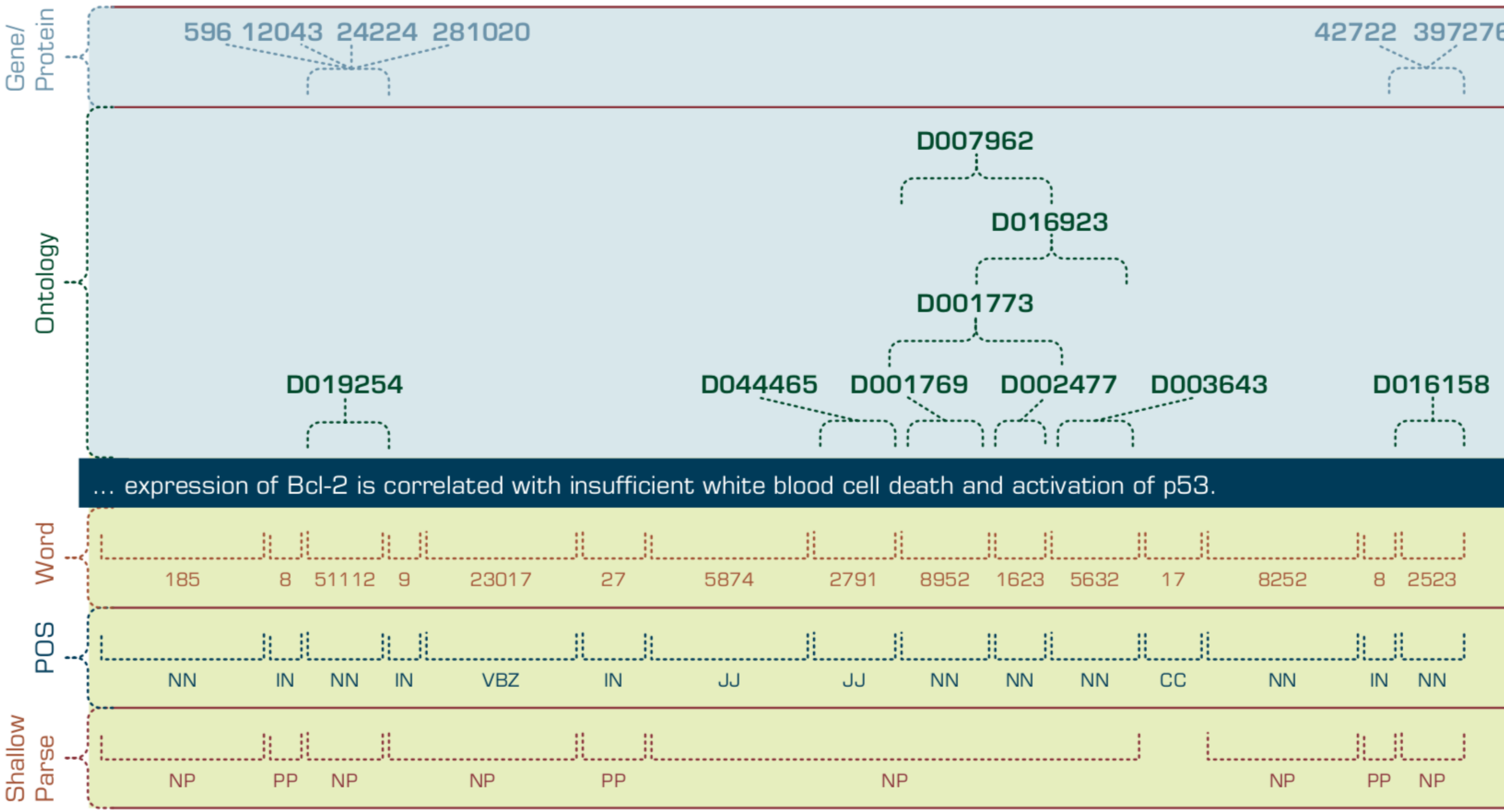
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Text-Based Deception-Detection Process



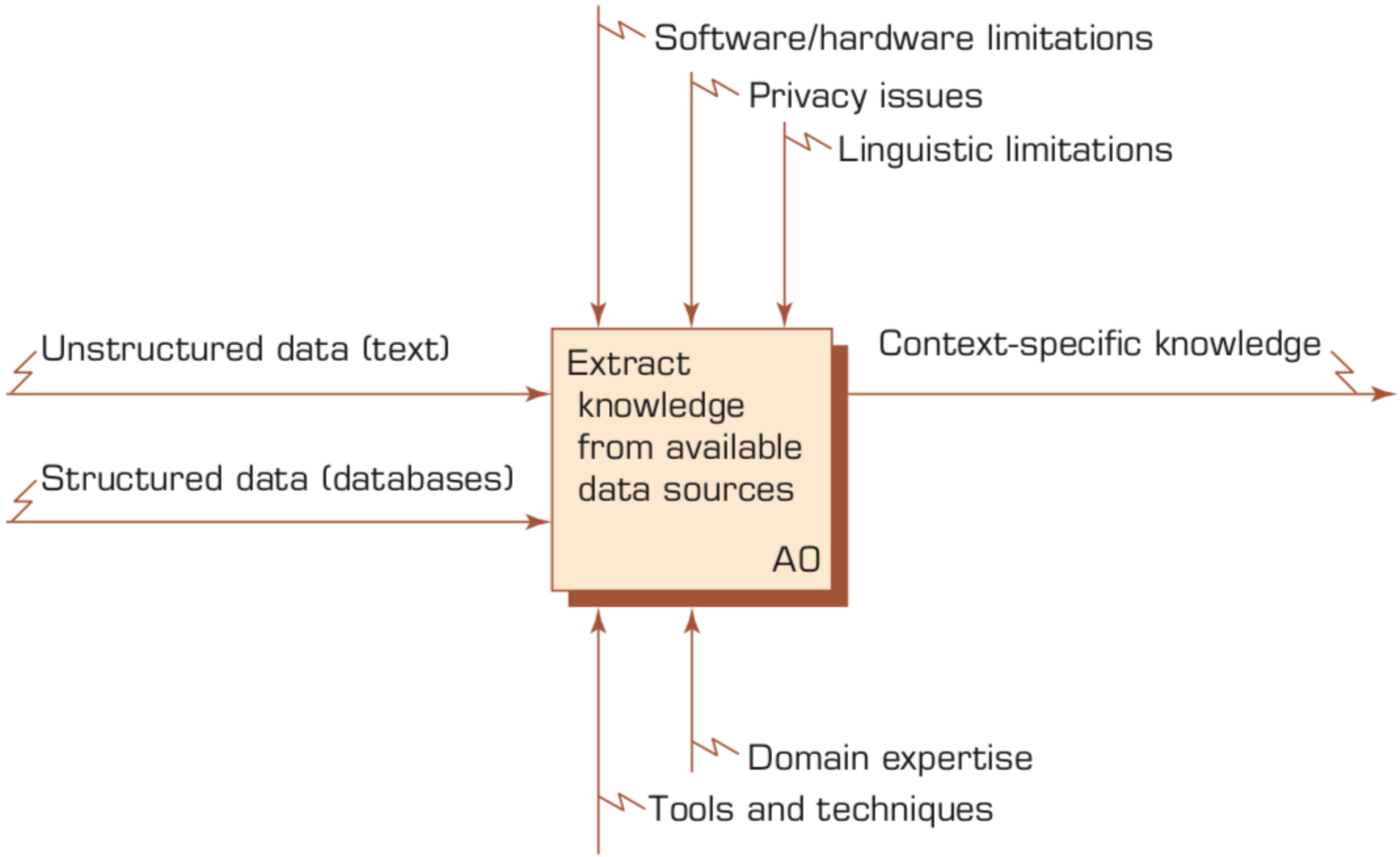
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Multilevel Analysis of Text for Gene/Protein Interaction Identification



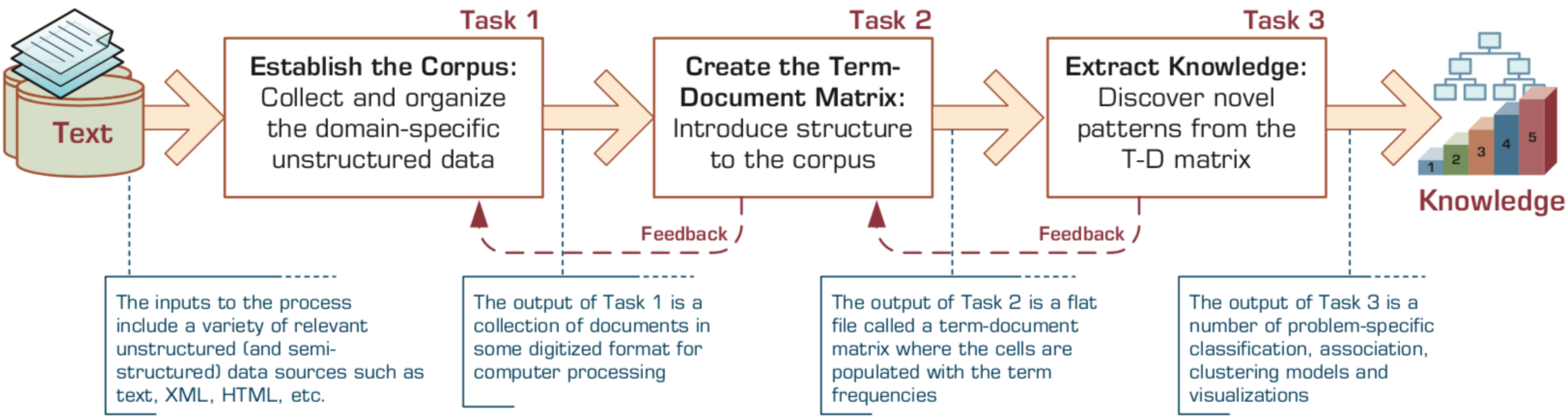
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Context Diagram for the Text Mining Process



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



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

Term–Document Matrix

Documents \ Terms	Investment Risk	Project Management	Software Engineering	Development	SAP	...
Document 1	1			1		
Document 2		1				
Document 3			3		1	
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

Emotions

Love

Joy

Surprise

Anger

Sadness

Fear

Source: Bing Liu (2011) , “Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data,” Springer, 2nd Edition,

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. … "

Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

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.

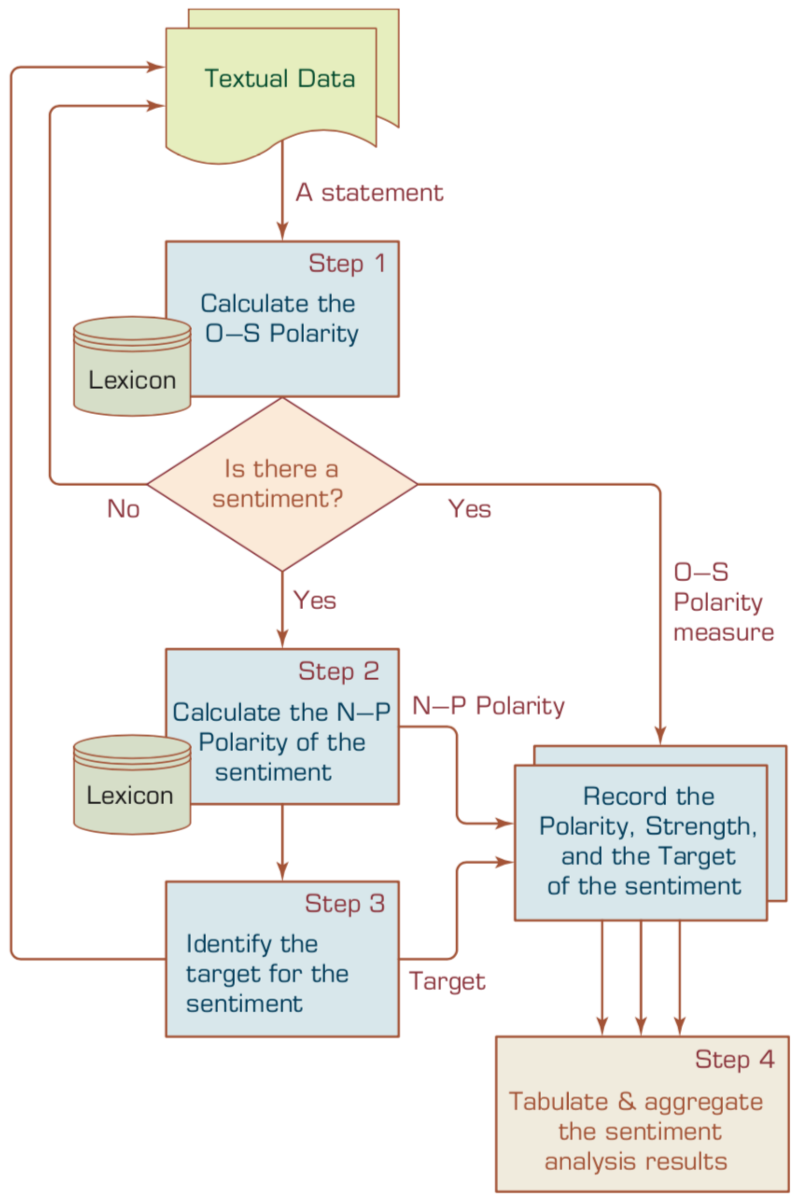
(6) She also thought the phone was too **expensive**, and wanted me to return it to the shop. … "

+Positive Opinion

-Negative Opinion

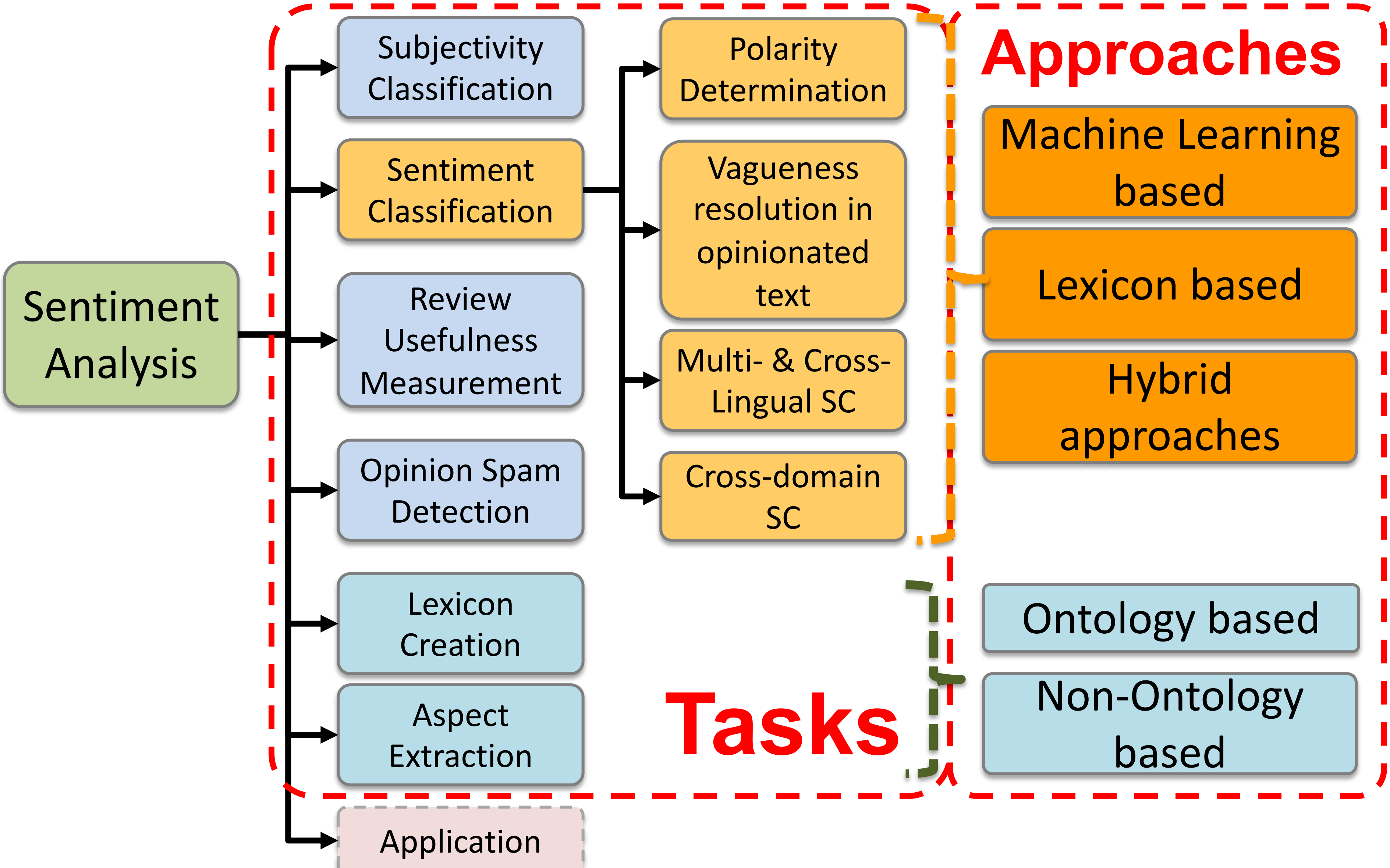
Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

A Multistep Process to Sentiment Analysis



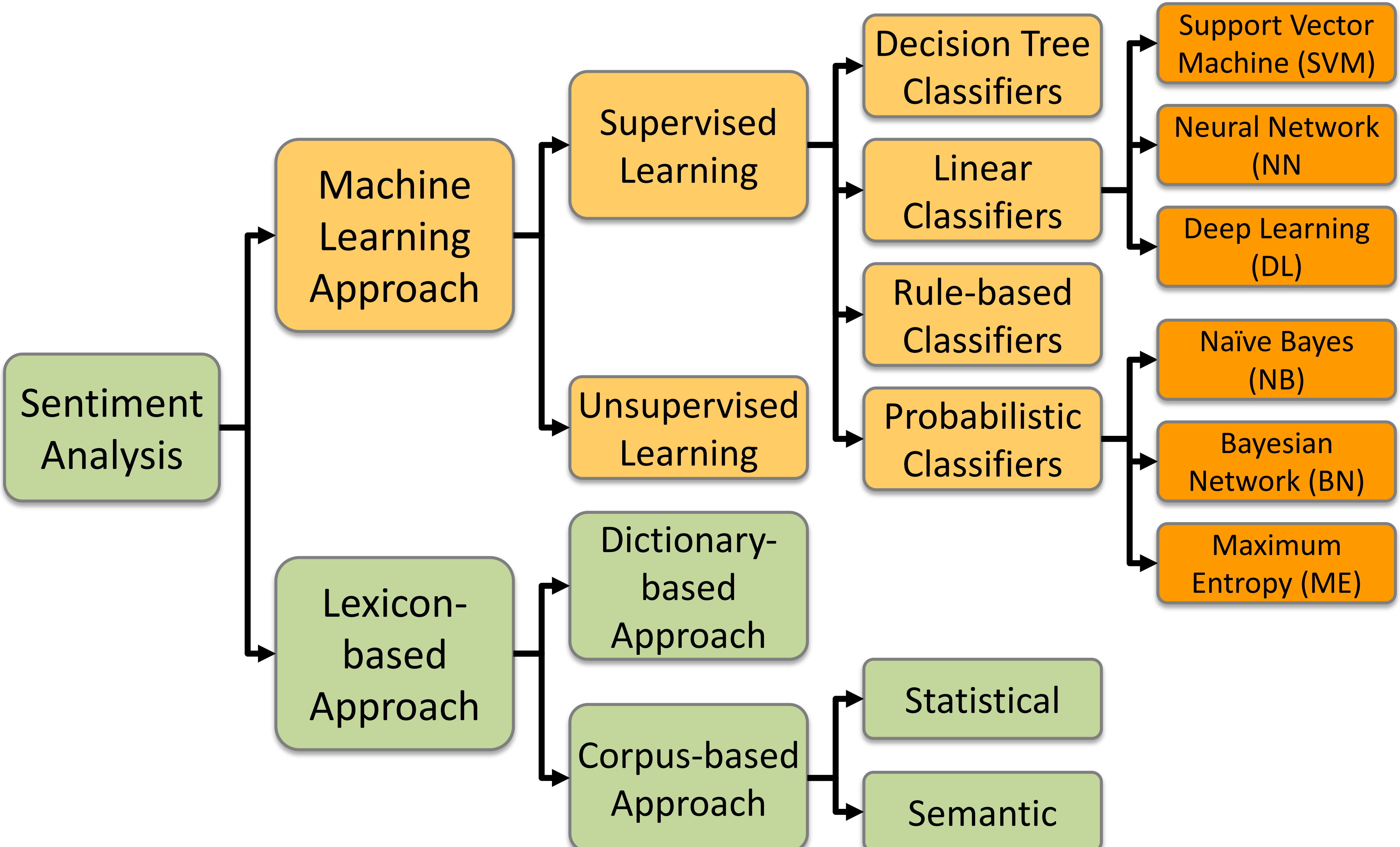
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Sentiment Analysis



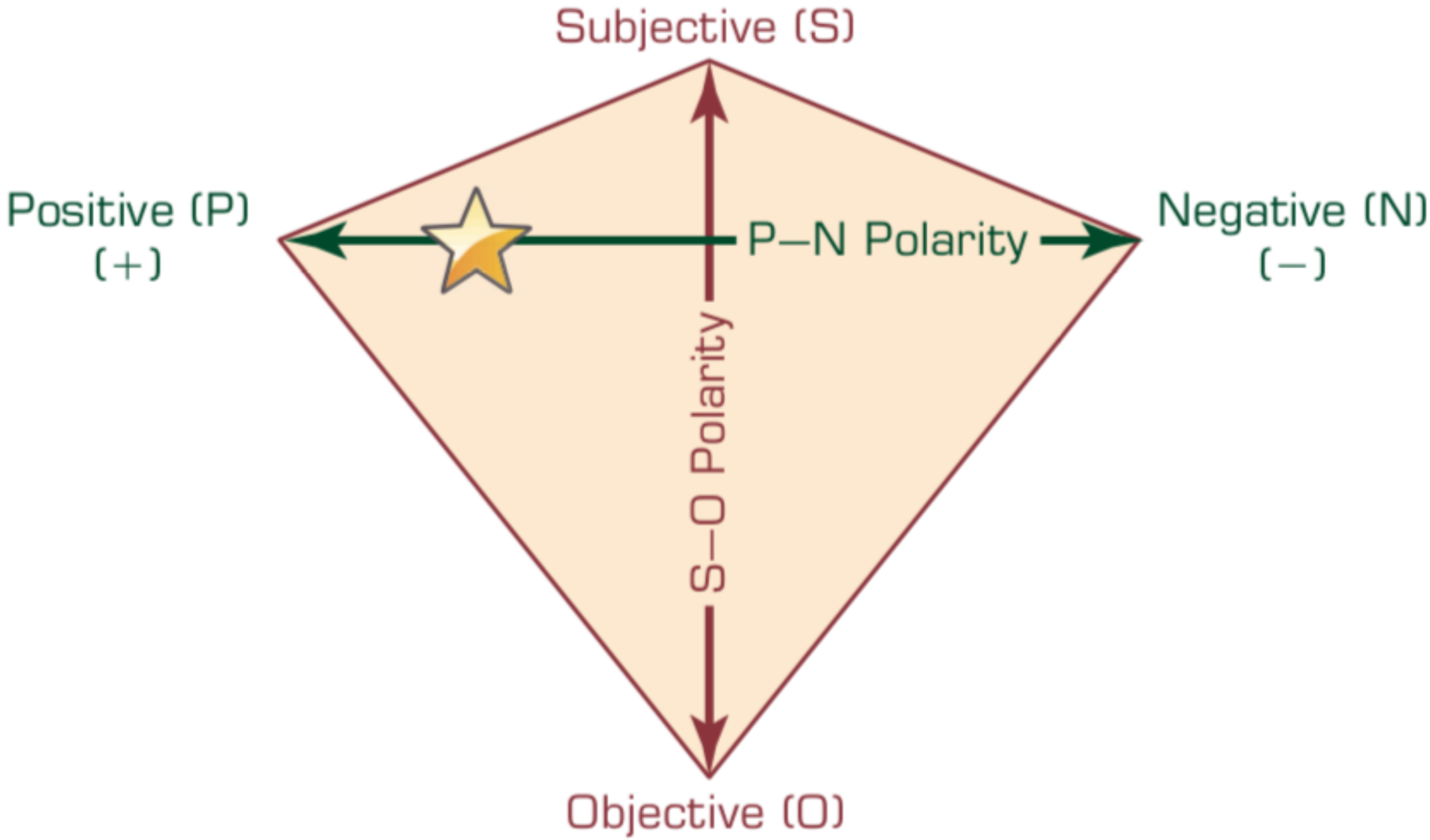
Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

Sentiment Classification Techniques



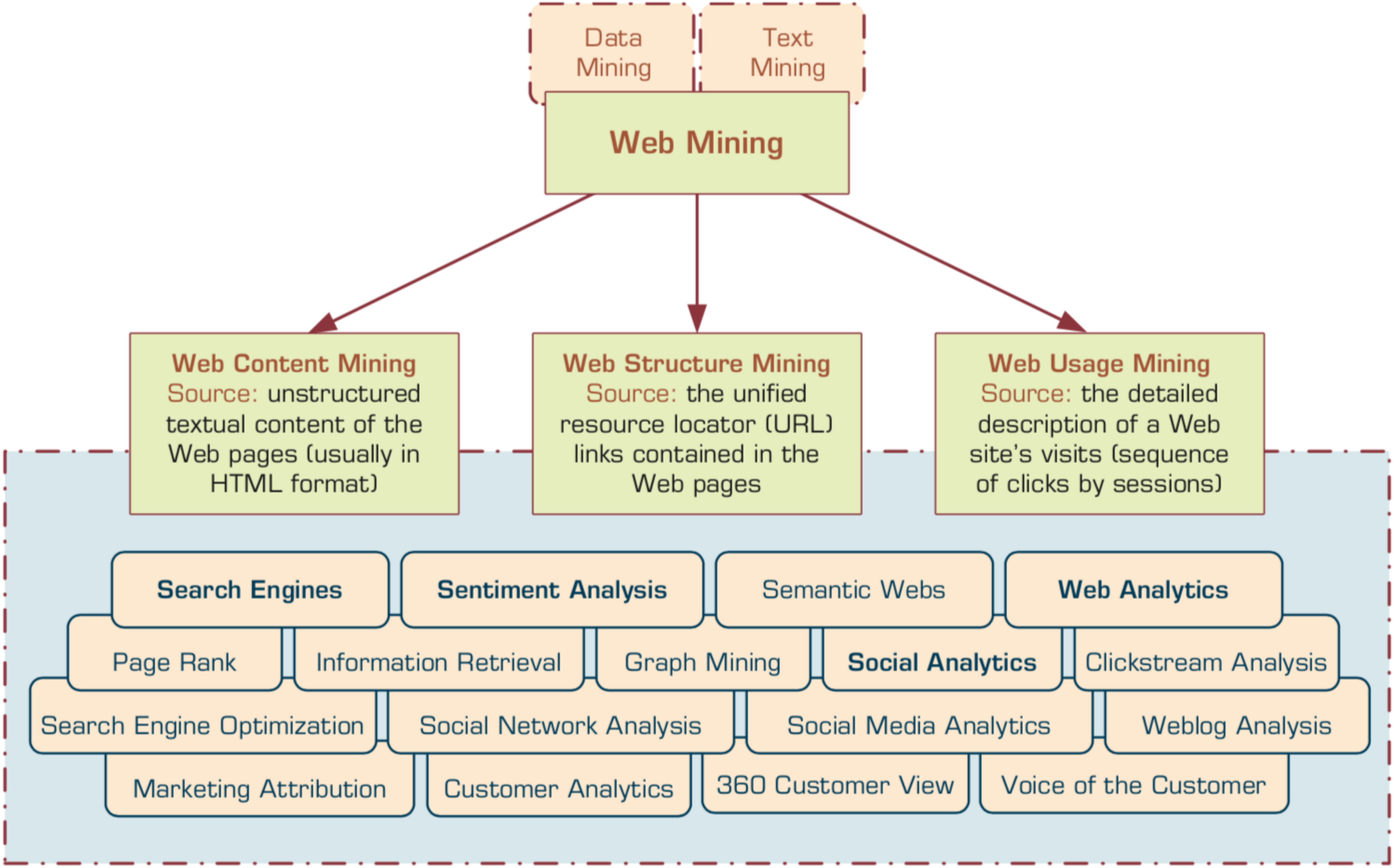
Source: Jesus Serrano-Guerrero, Jose A. Olivas, Francisco P. Romero, and Enrique Herrera-Viedma (2015), "Sentiment analysis: A review and comparative analysis of web services," Information Sciences, 311, pp. 18-38.

P–N Polarity and S–O Polarity Relationship



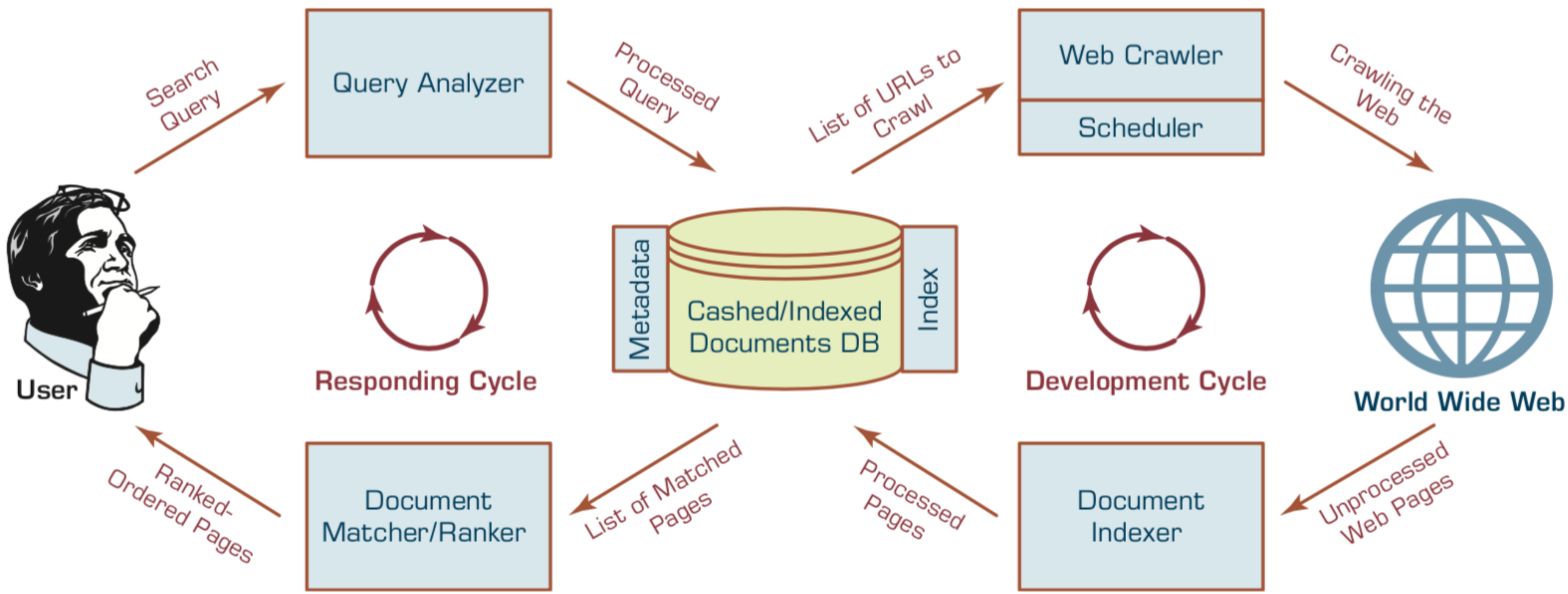
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Taxonomy of Web Mining



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

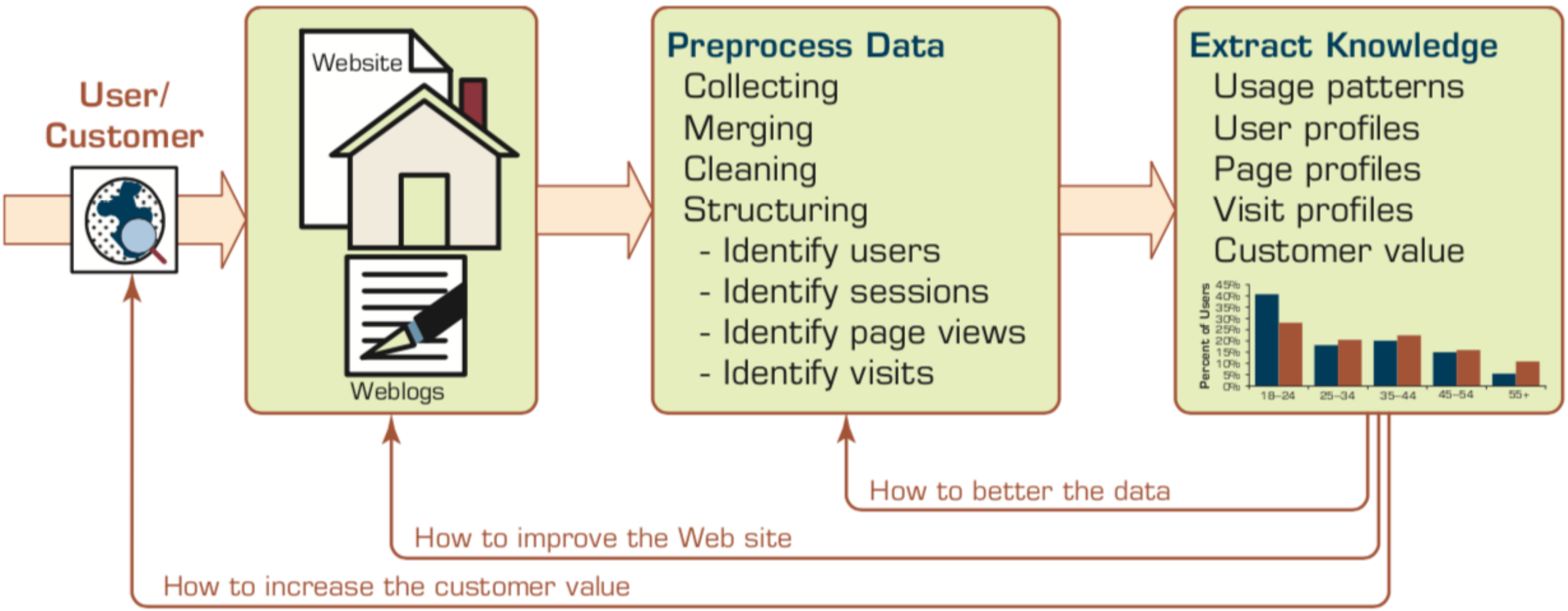


Web Usage Mining (Web Analytics)

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

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

Extraction of Knowledge from Web Usage Data

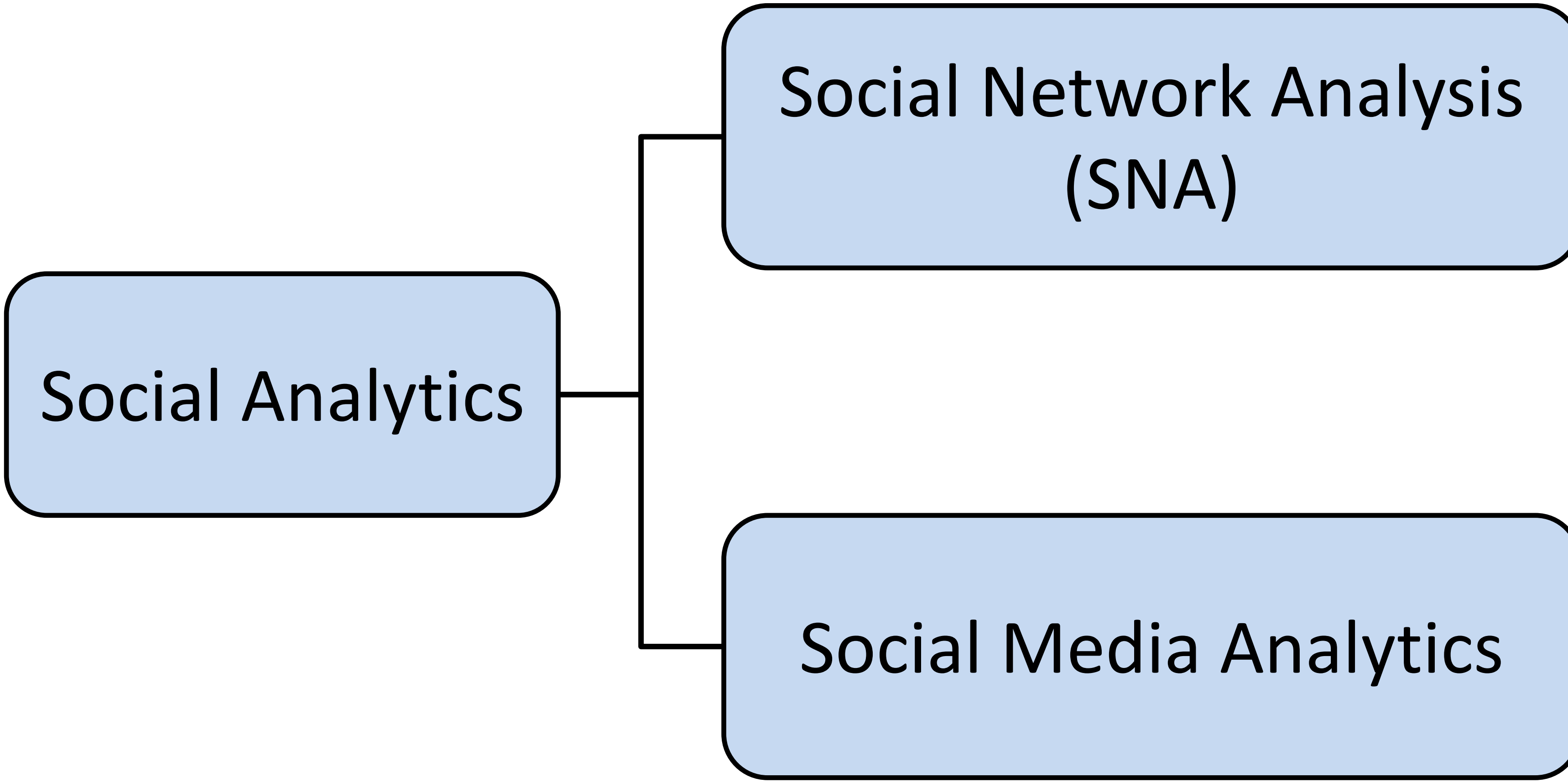


Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Social Analytics

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

Branches of Social Analytics



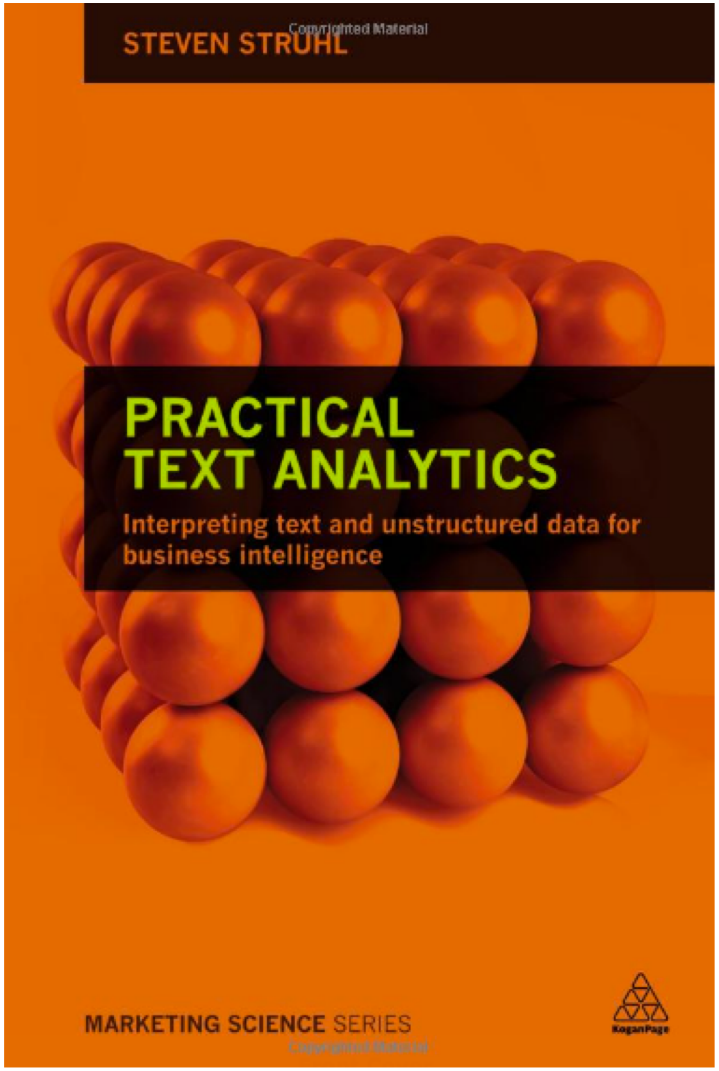
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

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)

Source: Vishal Gupta and Gurpreet S. Lehal (2009), "A survey of text mining techniques and applications," Journal of emerging technologies in web intelligence, vol. 1, no. 1, pp. 60-76.

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.

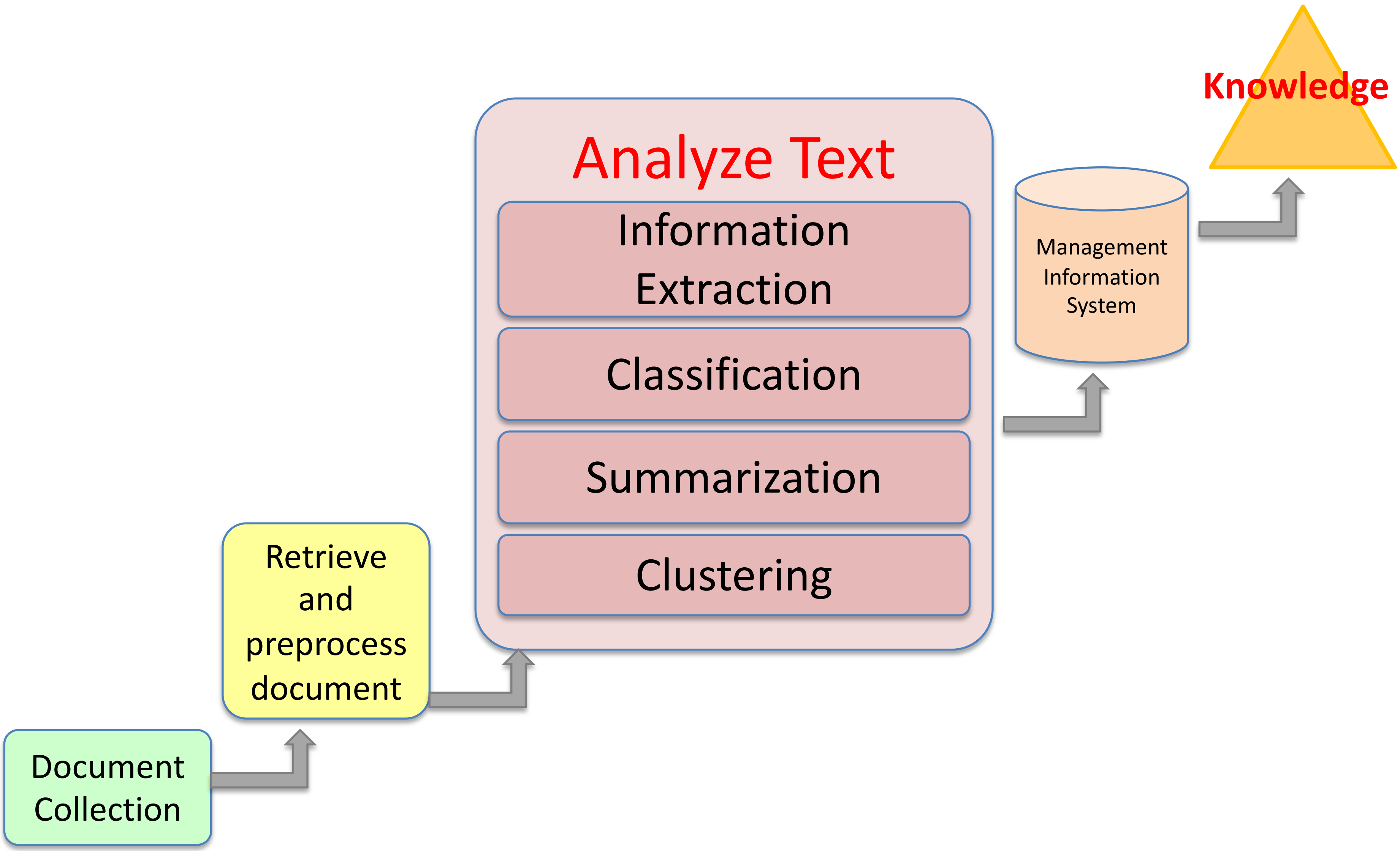
Source: Vishal Gupta and Gurpreet S. Lehal (2009), "A survey of text mining techniques and applications," Journal of emerging technologies in web intelligence, vol. 1, no. 1, pp. 60-76.

Text Mining:

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

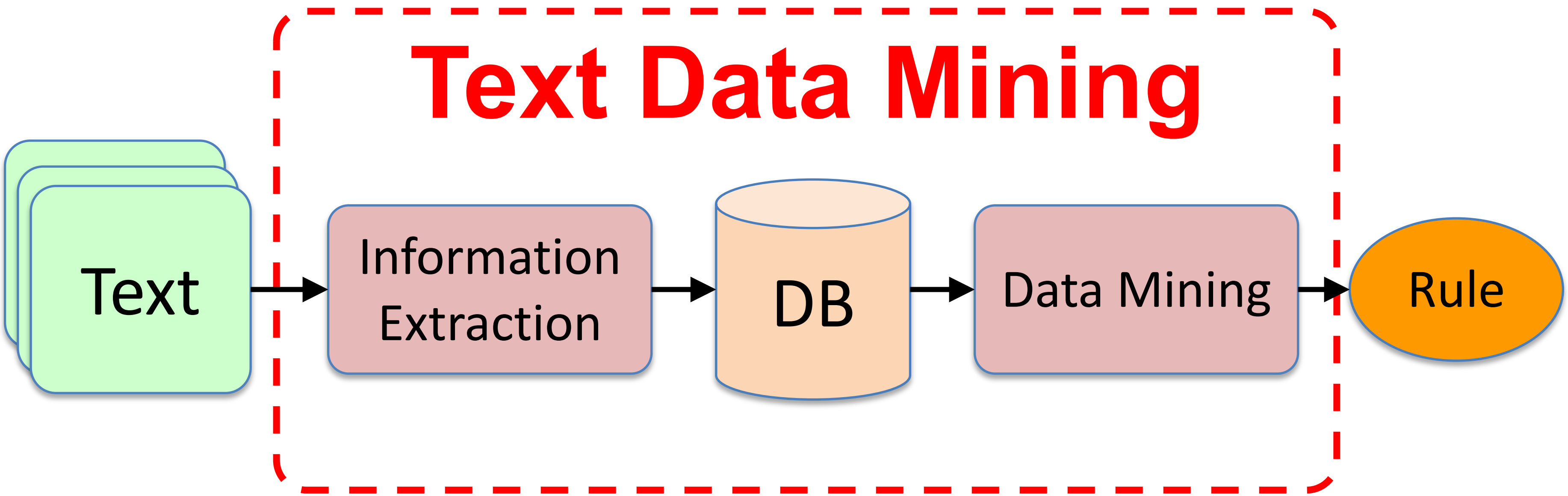
Source: Vishal Gupta and Gurpreet S. Lehal (2009), "A survey of text mining techniques and applications," Journal of emerging technologies in web intelligence, vol. 1, no. 1, pp. 60-76.

An example of Text Mining



Source: Vishal Gupta and Gurpreet S. Lehal (2009), "A survey of text mining techniques and applications," Journal of emerging technologies in web intelligence, vol. 1, no. 1, pp. 60-76.

Overview of Information Extraction based Text Mining Framework



Source: Vishal Gupta and Gurpreet S. Lehal (2009), "A survey of text mining techniques and applications," Journal of emerging technologies in web intelligence, vol. 1, no. 1, pp. 60-76.

NLP Libraries and Tools

Natural Language Processing with Python
– Analyzing Text with the Natural Language Toolkit

www.nltk.org/book/

Natural Language Processing with Python

NLTK

– 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.)

http://www.nltk.org/book/

spaCy

https://spacy.io/

gensim

https://radimrehurek.com/gensim/

TextBlob

TextBlob

Star 3,777

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

Useful Links

TextBlob @ PyPI
TextBlob @ GitHub
Issue Tracker

Stay Informed

Follow @sloria

Donate

If you find TextBlob useful,

TextBlob: Simplified Text Processing

Release v0.12.0. (Changelog)

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.

```
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

Polyglot

polyglot
latest

Search docs

- Installation
- Language Detection
- Tokenization
- Command Line Interface
- Downloading Models
- Word Embeddings
- Part of Speech Tagging
- Named Entity Extraction
- Morphological Analysis
- Transliteration
- Sentiment
- polyglot

Docs » Welcome to polyglot's documentation!

Edit on GitHub

Welcome to polyglot's documentation!

polyglot

downloads 17k/month pypi package 16.7.4 build passing docs passing

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

- Free software: GPLv3 license
- Documentation: 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/

scikit-learn

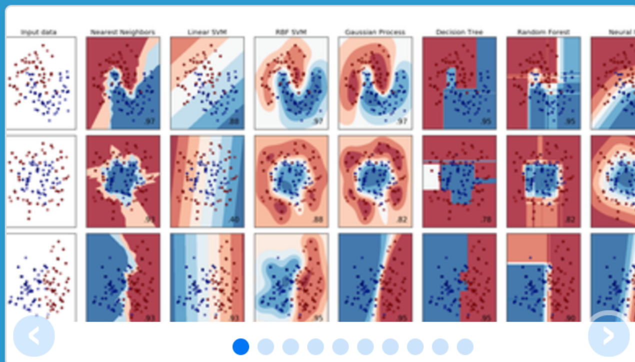
scikit learn

Home Installation Documentation Examples

powered by Google

Google Custom Search | Search

Fork me on GitHub



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/

http://nlp.stanford.edu/software/index.shtml

The Stanford Natural Language Processing Group

home · people · teaching · research · publications · software · events · local

Stanford NLP Software

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.

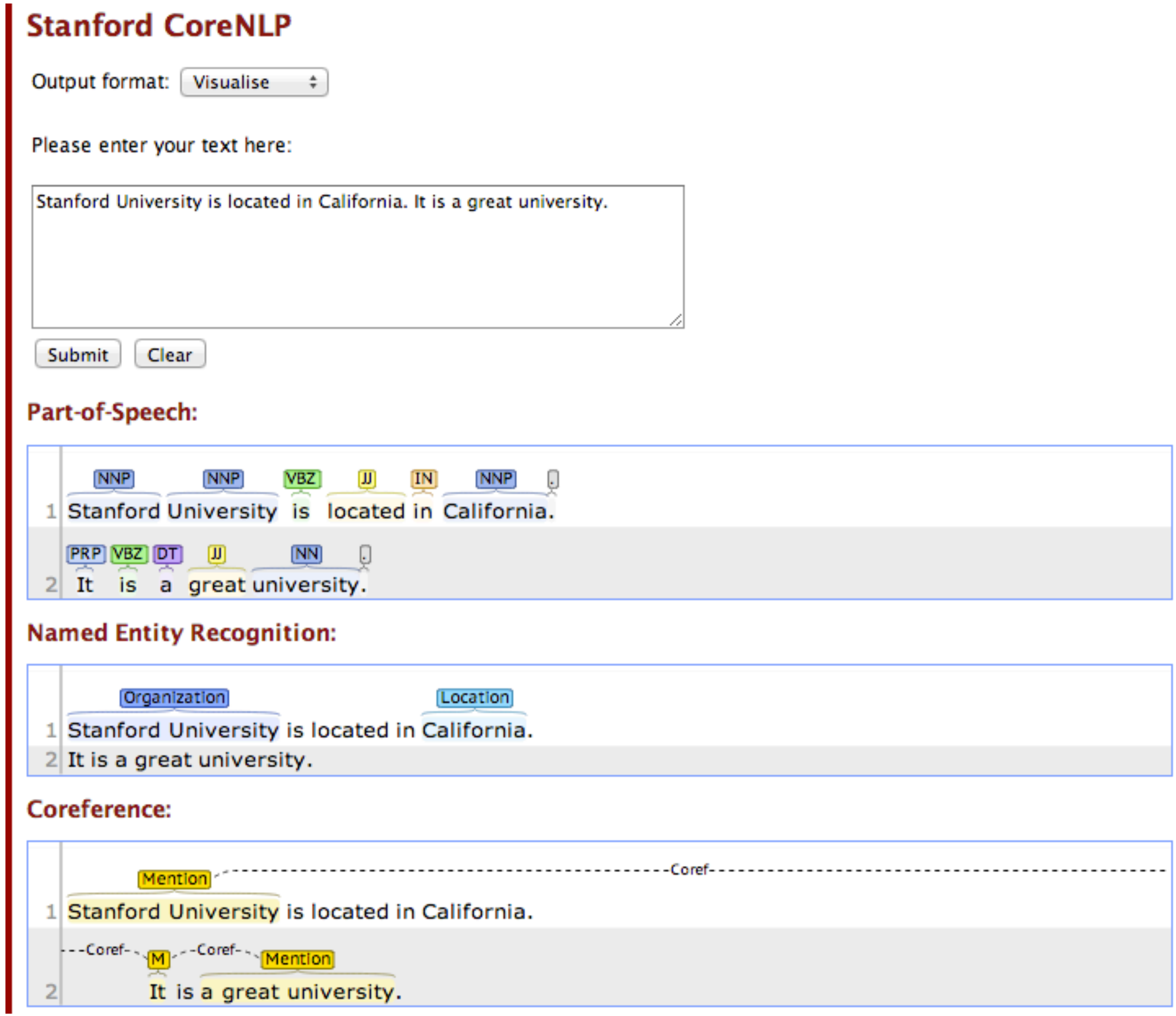
All these software distributions are open source, **licensed under the GNU General Public License** (v2 or later). Note that this is the *full* GPL, which allows many free uses, but *does not allow* its incorporation into any type of distributed proprietary software, even in part or in translation. **Commercial 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 CoreNLP http://nlp.stanford.edu:8080/corenlp/process

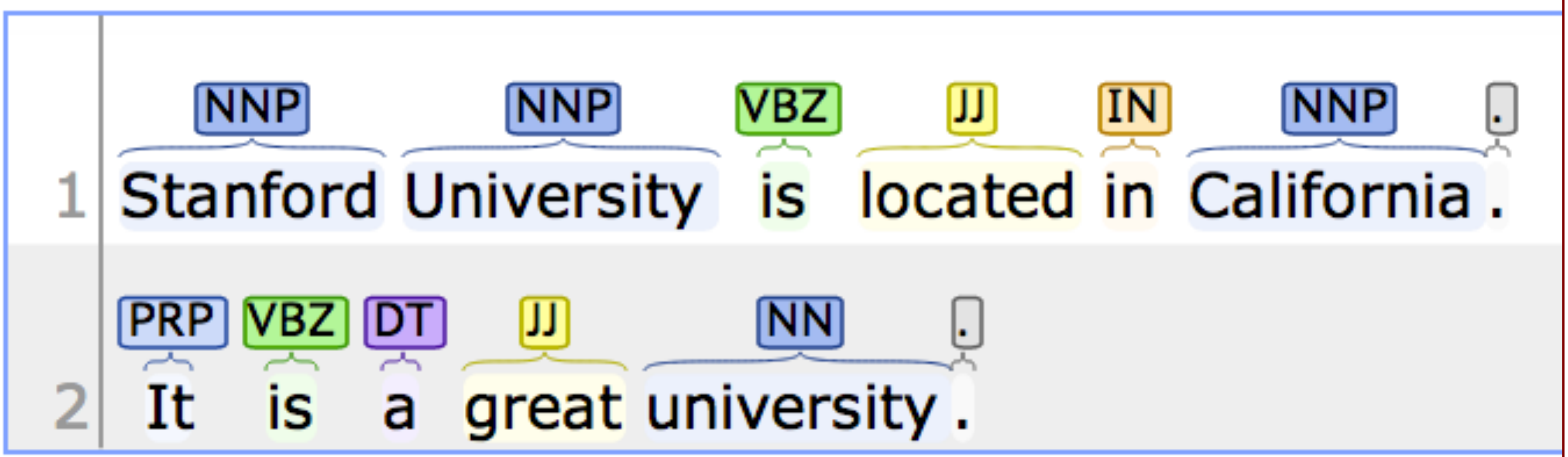


Stanford CoreNLP

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

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

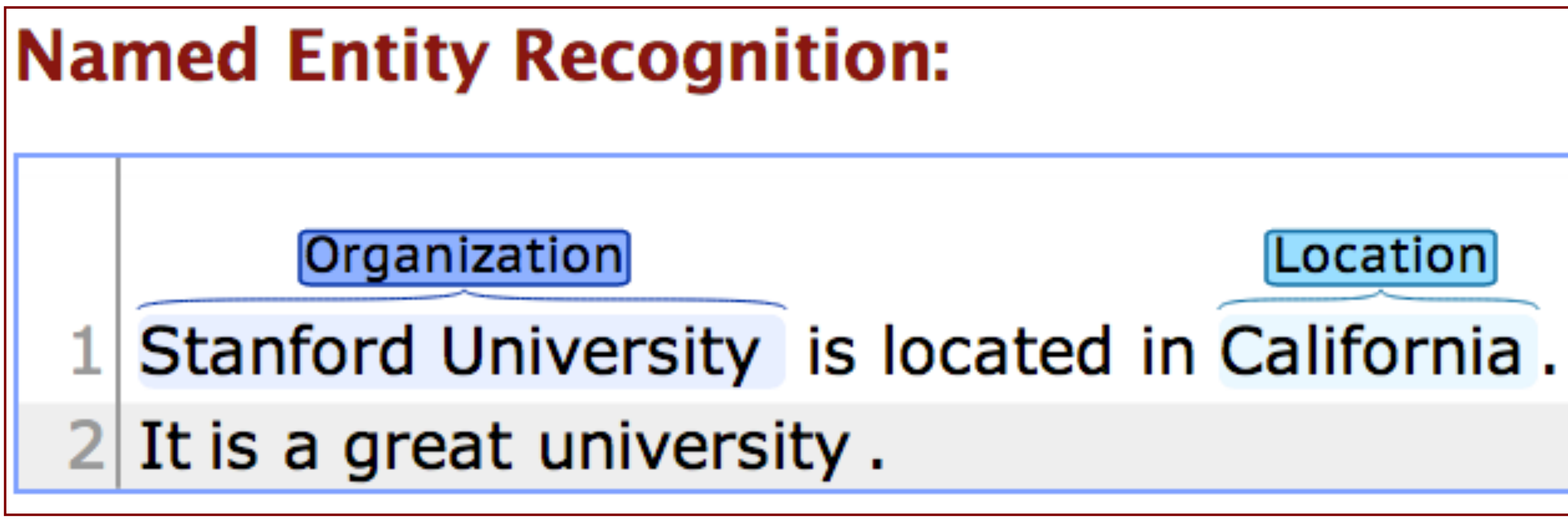
Part-of-Speech:



Stanford CoreNLP

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

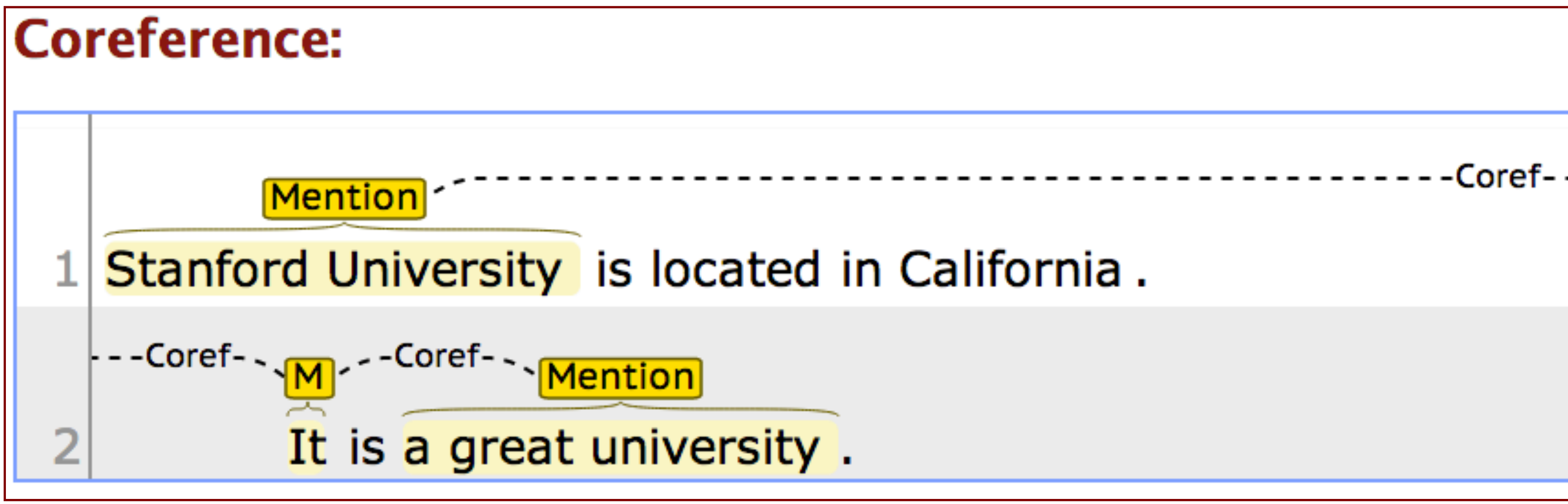
Stanford University is located in California.
It is a great university.



Stanford CoreNLP

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

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

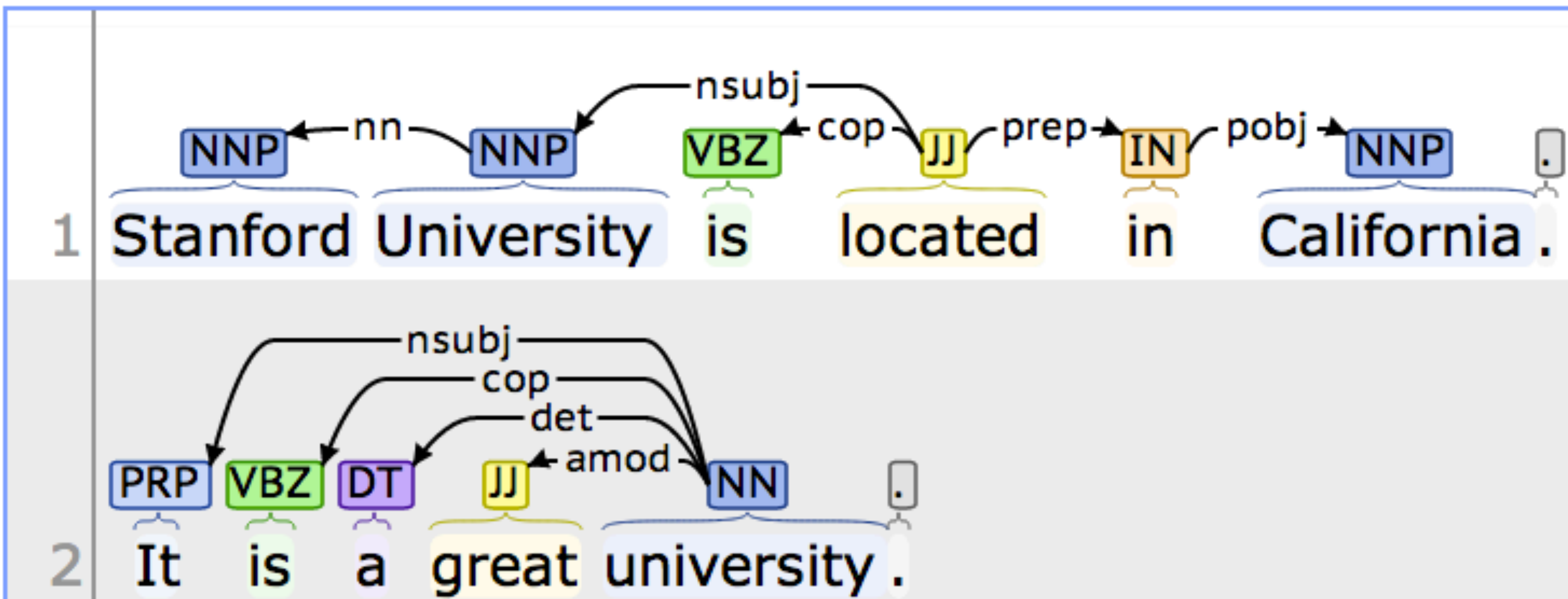


Stanford CoreNLP

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

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

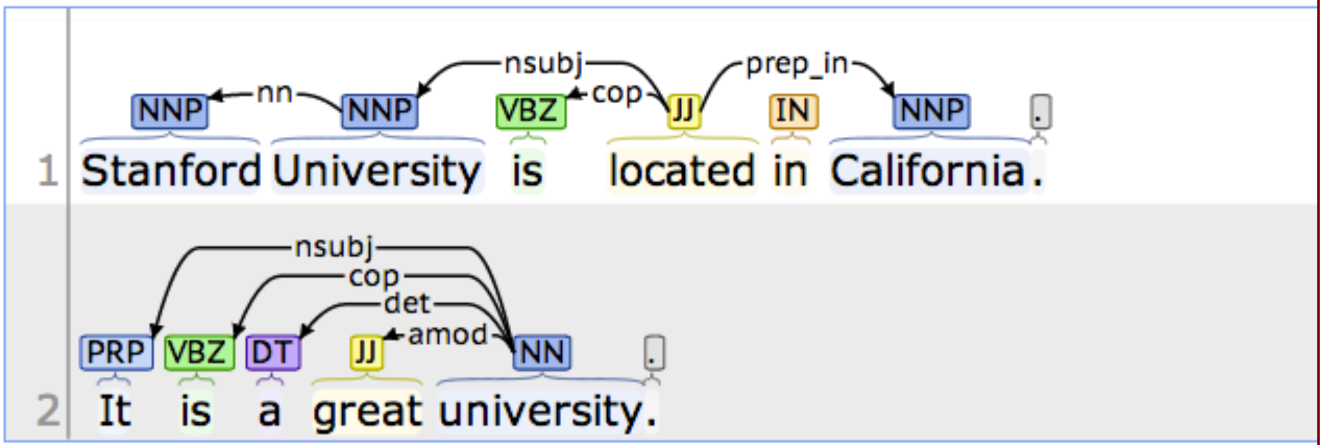
Basic dependencies:



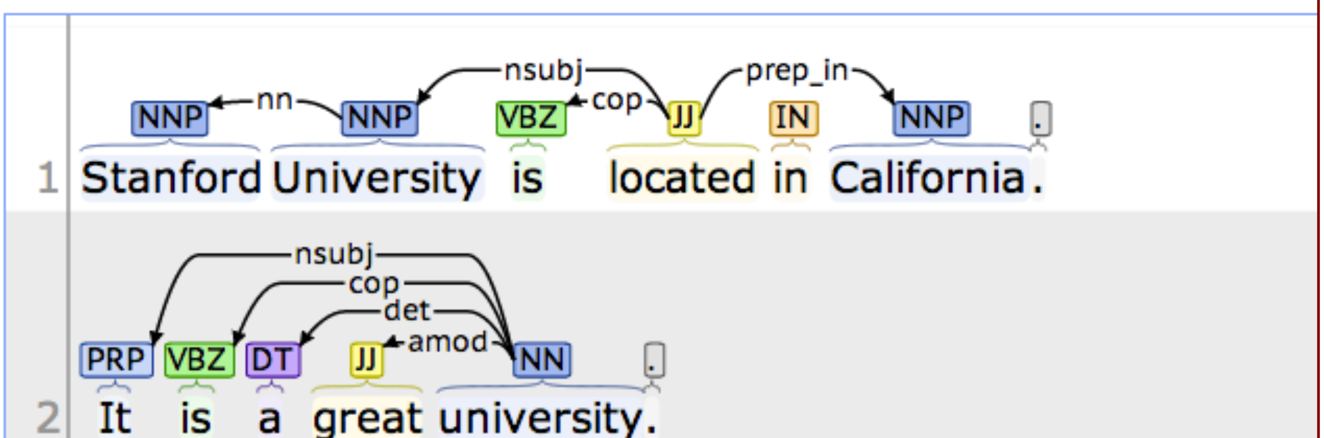
Stanford CoreNLP

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

Collapsed dependencies:



Collapsed CC-processed dependencies:



Visualisation provided using the brat visualisation/annotation software.

Stanford CoreNLP

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

Output format: Pretty print

Please enter your text here:

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

Submit Clear

Stanford CoreNLP XML Output

Document

Document Info

Sentences

Sentence #1

Tokens

Id	Word	Lemma	Char begin	Char end	POS	NER	Normalized NER	Speaker
1	Stanford	Stanford	0	8	NNP	ORGANIZATION		PER0
2	University	University	9	19	NNP	ORGANIZATION		PER0
3	is	be	20	22	VBZ	O		PER0
4	located	located	23	30	JJ	O		PER0
5	in	in	31	33	IN	O		PER0
6	California	California	34	44	NNP	LOCATION		PER0
7	.	.	44	45	.	O		PER0

Parse tree

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

Stanford CoreNLP

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

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

Sentence #1

Tokens

Id	Word	Lemma	Char begin	Char end	POS	NER	Normalized NER	Speaker
1	Stanford	Stanford	0	8	NNP	ORGANIZATION		PER0
2	University	University	9	19	NNP	ORGANIZATION		PER0
3	is	be	20	22	VBZ	O		PER0
4	located	located	23	30	JJ	O		PER0
5	in	in	31	33	IN	O		PER0
6	California	California	34	44	NNP	LOCATION		PER0
7	.	.	44	45	.	O		PER0

Parse tree

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

Stanford CoreNLP

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

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

Sentence #2

Tokens

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

Parse tree

(ROOT (S (NP (PRP It)) (VP (VBZ is) (NP (DT a) (JJ great) (NN university))) (. .)))

Stanford CoreNLP

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

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

Coreference resolution graph

1.

Sentence	Head	Text	Context
1	2 (gov)	Stanford University	
2	1	It	
2	5	a great university	

Tokens

Id	Word	Lemma	Char begin	Char end	POS	NER	Normalized NER	Speaker
1	Stanford	Stanford	0	8	NNP	ORGANIZATION		PER0
2	University	University	9	19	NNP	ORGANIZATION		PER0
3	is	be	20	22	VBZ	O	PER0	
4	located	located	23	30	JJ	O	PER0	
5	in	in	31	33	IN	O	PER0	
6	California	California	34	44	NNP	LOCATION	PER0	
7	.	.	44	45	.	O	PER0	

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 CoreNLP

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

Output format: XML

Please enter your text here:

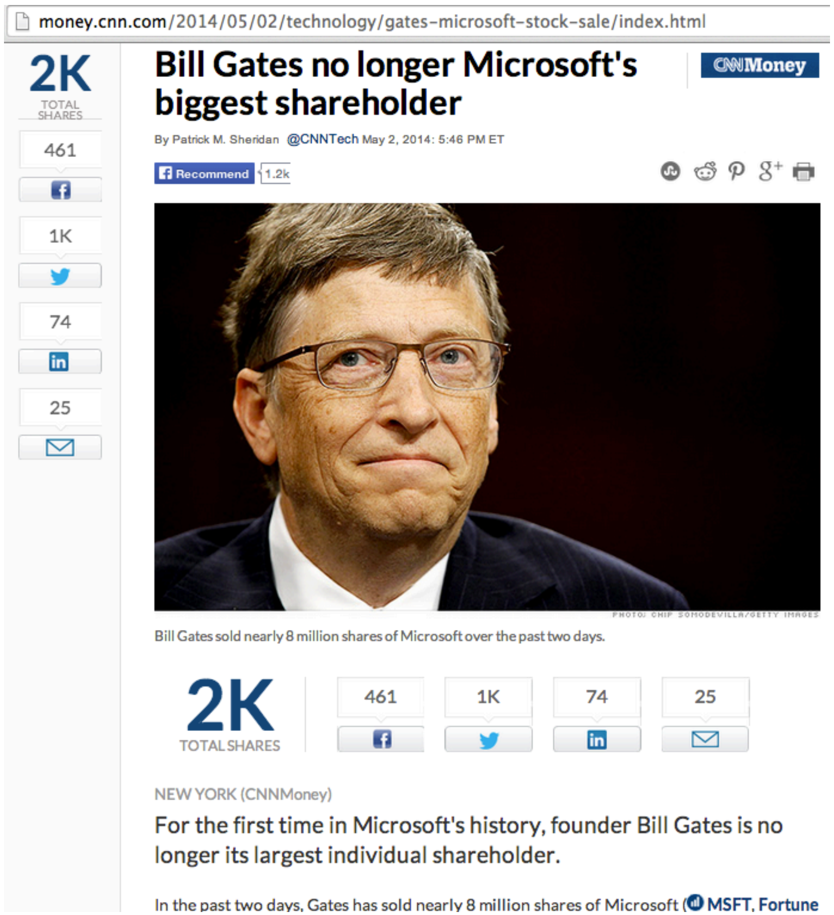
Stanford University is located in California. It is a great university.

Submit Clear


```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="CoreNLP-to-HTML.xsl" type="text/xsl"?>
<root>
  <document>
    <sentences>
      <sentence id="1">
        <tokens>
          <token id="1">
            <word>Stanford</word>
            <lemma>Stanford</lemma>
            <CharacterOffsetBegin>0</CharacterOffsetBegin>
            <CharacterOffsetEnd>8</CharacterOffsetEnd>
            <POS>NNP</POS>
            <NER>ORGANIZATION</NER>
            <Speaker>PER0</Speaker>
          </token>
          <token id="2">
            <word>University</word>
            <lemma>University</lemma>
            <CharacterOffsetBegin>9</CharacterOffsetBegin>
            <CharacterOffsetEnd>19</CharacterOffsetEnd>
            <POS>NNP</POS>
            <NER>ORGANIZATION</NER>
            <Speaker>PER0</Speaker>
          </token>
```

NER for News Article

http://money.cnn.com/2014/05/02/technology/gates-microsoft-stock-sale/index.html



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.

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: 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.

Submit Clear

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

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: inlineXML

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

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>.

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: xml

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)

Submit Clear

<wi num="0" entity="O">Bill</wi> <wi num="1" entity="O">Gates</wi> <wi num="2" entity="O">no</wi> <wi num="3" entity="O">longer</wi> <wi num="4" entity="ORGANIZATION">Microsoft</wi><wi num="5" entity="O">'s</wi> <wi num="6" entity="O">biggest</wi> <wi num="7" entity="O">shareholder</wi> <wi num="8" entity="O">By</wi> <wi num="9" entity="PERSON">Patrick</wi> <wi num="10" entity="PERSON">M.</wi> <wi num="11" entity="PERSON">Sheridan</wi> <wi num="12" entity="O">@CNNTech</wi> <wi num="13" entity="DATE">May</wi> <wi num="14" entity="DATE">2</wi><wi num="15" entity="DATE">,</wi> <wi num="16" entity="DATE">2014</wi><wi num="17" entity="O">:</wi> <wi num="18" entity="O">5:46</wi> <wi num="19" entity="O">PM</wi> <wi num="20" entity="O">ET</wi> <wi num="21" entity="O">Bill</wi> <wi num="22" entity="O">Gates</wi> <wi num="23" entity="O">sold</wi> <wi num="24" entity="O">nearly</wi> <wi num="25" entity="O">8</wi> <wi num="26" entity="O">million</wi> <wi num="27" entity="O">shares</wi> <wi num="28" entity="O">of</wi> <wi num="29" entity="ORGANIZATION">Microsoft</wi> <wi num="30" entity="O">over</wi> <wi num="31" entity="O">the</wi> <wi num="32" entity="O">past</wi> <wi num="33" entity="O">two</wi> <wi num="34" entity="O">days</wi><wi num="35" entity="O">.</wi> <wi num="0" entity="LOCATION">NEW</wi> <wi num="1" entity="LOCATION">YORK</wi> <wi num="2" entity="O">-LRB-</wi><wi num="3" entity="O">CNNMoney</wi><wi num="4" entity="O">-RRB-</wi> <wi num="5" entity="O">For</wi> <wi num="6" entity="O">the</wi> <wi num="7" entity="O">first</wi> <wi num="8" entity="O">time</wi> <wi num="9" entity="O">in</wi> <wi num="10" entity="ORGANIZATION">Microsoft</wi><wi num="11" entity="O">'s</wi> <wi num="12" entity="O">history</wi><wi num="13" entity="O">,</wi> <wi num="14" entity="O">founder</wi> <wi num="15" entity="PERSON">Bill</wi> <wi num="16" entity="PERSON">Gates</wi> <wi num="17" entity="O">is</wi> <wi num="18" entity="O">no</wi> <wi num="19" entity="O">longer</wi> <wi num="20" entity="O">its</wi> <wi num="21" entity="O">largest</wi> <wi num="22" entity="O">individual</wi> <wi num="23" entity="O">shareholder</wi><wi num="24" entity="O">.</wi> <wi num="0" entity="O">In</wi> <wi num="1" entity="O">the</wi> <wi num="2" entity="DATE">past</wi> <wi num="3" entity="DATE">two</wi> <wi num="4" entity="DATE">days</wi><wi num="5" entity="O"> </wi> <wi num="6" entity="O">Gates</wi> <wi num="7" entity="O">has</wi> <wi num="8" entity="O">sold</wi>

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

Bill/O Gates/O no/O longer/O Microsoft/ORGANIZATION's/O biggest/O shareholder/O By/O Patrick/PERSON M./PERSON Sheridan/PERSON @CNNTech/O May/DATE 2/DATE,/DATE 2014/DATE:/O 5:46/O PM/O ET/O Bill/O Gates/O sold/O nearly/O 8/O million/O shares/O of/O Microsoft/ORGANIZATION over/O the/O past/O two/O days/O./O NEW/LOCATION YORK/LOCATION -LRB-/OCNNMoney/O-RRB-/O For/O the/O first/O time/O in/O Microsoft/ORGANIZATION's/O history/O,/O founder/O Bill/PERSON Gates/PERSON is/O no/O longer/O its/O largest/O individual/O shareholder/O./O In/O the/O past/DATE two/DATE days/DATE,/O Gates/O has/O sold/O nearly/O 8/O million/O shares/O of/O Microsoft/ORGANIZATION -LRB-/OMSFT/ORGANIZATION,/O Fortune/O 500/O-RRB-/O,/O bringing/O down/O his/O total/O to/O roughly/O 330/O million/O./O That/O puts/O him/O behind/O Microsoft/ORGANIZATION's/O former/O CEO/O Steve/PERSON Ballmer/PERSON who/O owns/O 333/O million/O shares/O./O Related/O:/O Gates/O reclaims/O title/O of/O world/O's/O richest/O billionaire/O Ballmer/PERSON,/O who/O was/O Microsoft/ORGANIZATION's/O CEO/O until/O earlier/DATE this/DATE year/DATE,/O was/O one/O of/O Gates/O'/O first/O hires/O./O It/O's/O a/O passing/O of/O the/O torch/O for/O Gates/O who/O has/O always/O been/O the/O largest/O single/O owner/O of/O his/O company/O's/O stock/O./O Gates/O now/O spends/O his/O time/O and/O personal/O fortune/O helping/O run/O the/O Bill/ORGANIZATION &/ORGANIZATION Melinda/ORGANIZATION Gates/ORGANIZATION foundation/O./O The/O foundation/O has/O spent/O $/MONEY28.3/MONEY billion/MONEY fighting/O hunger/O and/O poverty/O since/O its/O inception/O back/O in/O 1997/DATE./O

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.

Submit Clear

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.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.

Submit Clear

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

Classifier: english.muc.**7class**.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 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.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 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

Stanford Named Entity Tagger (NER)

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

Stanford NER Output Format: inlineXML

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>.

Stanford Named Entity Tagger (NER)

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

Stanford NER Output Format: slashTags

Bill/O Gates/O no/O longer/O Microsoft/ORGANIZATION's/O biggest/O shareholder/O By/O Patrick/PERSON M./PERSON Sheridan/PERSON @CNNTech/O May/DATE 2/DATE,/DATE 2014/DATE:/O 5:46/O PM/O ET/O Bill/O Gates/O sold/O nearly/O 8/O million/O shares/O of/O Microsoft/ORGANIZATION over/O the/O past/O two/O days/O./O NEW/LOCATION YORK/LOCATION -LRB-/OCNNMoney/O-RRB-/O For/O the/O first/O time/O in/O Microsoft/ORGANIZATION's/O history/O,/O founder/O Bill/PERSON Gates/PERSON is/O no/O longer/O its/O largest/O individual/O shareholder/O./O In/O the/O past/DATE two/DATE days/DATE,/O Gates/O has/O sold/O nearly/O 8/O million/O shares/O of/O Microsoft/ORGANIZATION -LRB-/OMSFT/ORGANIZATION,/O Fortune/O 500/O-RRB-/O,/O bringing/O down/O his/O total/O to/O roughly/O 330/O million/O./O That/O puts/O him/O behind/O Microsoft/ORGANIZATION's/O former/O CEO/O Steve/PERSON Ballmer/PERSON who/O owns/O 333/O million/O shares/O./O Related/O:/O Gates/O reclaims/O title/O of/O world/O's/O richest/O billionaire/O Ballmer/PERSON,/O who/O was/O Microsoft/ORGANIZATION's/O CEO/O until/O earlier/DATE this/DATE year/DATE,/O was/O one/O of/O Gates/O'/O first/O hires/O./O It/O's/O a/O passing/O of/O the/O torch/O for/O Gates/O who/O has/O always/O been/O the/O largest/O single/O owner/O of/O his/O company/O's/O stock/O./O Gates/O now/O spends/O his/O time/O and/O personal/O fortune/O helping/O run/O the/O Bill/ORGANIZATION &/ORGANIZATION Melinda/ORGANIZATION Gates/ORGANIZATION foundation/O./O The/O foundation/O has/O spent/O $/MONEY28.3/MONEY billion/MONEY fighting/O hunger/O and/O poverty/O since/O its/O inception/O back/O in/O 1997/DATE./O

CKIP 中研院中文斷詞系統

http://ckipsvr.iis.sinica.edu.tw/

歐巴馬(Nb)　是(SHI)　美國(Nc)　的(DE)　一(Neu)　位(Nf)　總統(Na)

中文文字處理：中文斷詞

即時 要聞 娛樂 運動 全球 社會 產經 股市 健康 生活 文教 評論 地方 兩岸 數

莎士比亞在淡江 遇見賽萬提斯

分享 G+分享 留言 列印 存新聞 A- A+

2016-04-26 02:27 聯合報 記者徐葳倫／淡水報導 讚 分享 20 傳送 G+1 0

淡江大學舉辦「當莎士比亞遇見賽萬提斯」系列活動，讓師生幫莎士比亞、賽萬提斯著色，畫出五彩繽紛的「文學大師」。 記者徐葳倫／攝影

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

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。

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

CKIP 中研院中文斷詞系統

http://ckipsvr.iis.sinica.edu.tw/

中文斷詞系統

相關系統： 斷詞系統 | 剖析系統 | 詞首詞尾 | 平衡語料庫 | 廣義知網 | 句結構樹庫 | 錯字偵測

- 簡介
- 未知詞擷取做法
- 詞類標記列表
- 線上展示
- 線上服務申請
- 線上資源
- 公告
- 聯絡我們

隱私權聲明 | 版權聲明

自 2014/01/06 起，本斷詞系統已經處理過 28270134 篇文章

送出 清除

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。

CKIP 中研院中文斷詞系統

http://ckipsvr.iis.sinica.edu.tw/

中文斷詞系統

相關系統： 斷詞系統 | 剖析系統 | 詞首詞尾 | 平衡語料庫 | 廣義知網 | 句結構樹庫 | 錯字偵測

- 簡介
- 未知詞擷取做法
- 詞類標記列表
- 線上展示
- 線上服務申請
- 線上資源
- 公告
- 聯絡我們

隱私權聲明 | 版權聲明

數位典藏國家型科技計畫 National Digital Archives Program

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)　：(COLONCATEGOR

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)

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)

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)

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(VH)　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(Neqa)　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)　還(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)　五彩繽紛(VH)　的(DE)　莎士比亞(Nb)　和(Caa)　賽萬提(Nb)　斯人(Na)　像(VG)　。(PERIODCATEGORY)

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

讀(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(Neqa)　作品(Na)　都(D)　放(VC)　在(P)　圖書館(Nc)　８樓(Nc)　，(COMMACATEGORY)

CKIP 中研院中文斷詞系統

http://ckipsvr.iis.sinica.edu.tw/

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。

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)

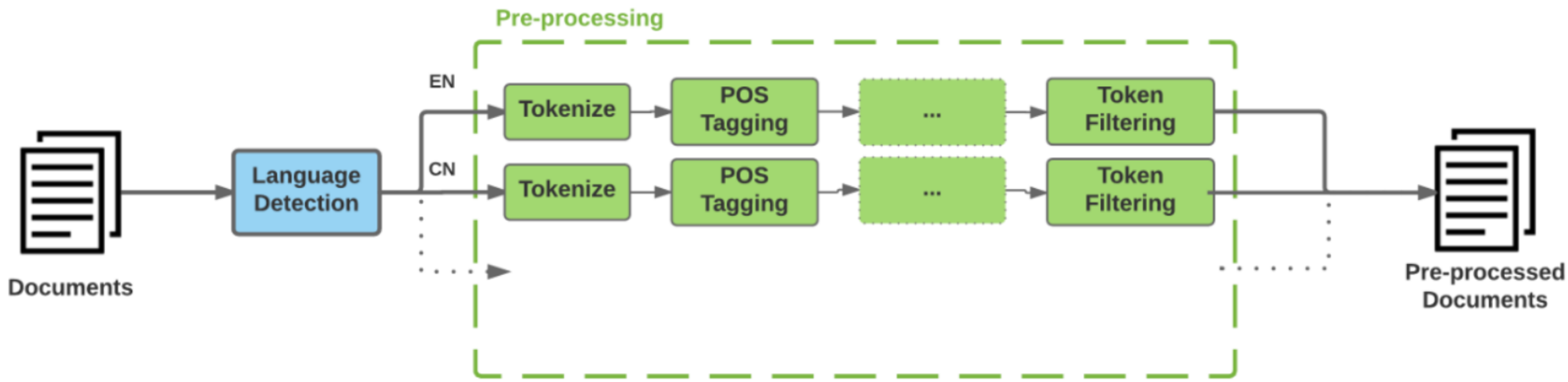
Vector Representations of Words

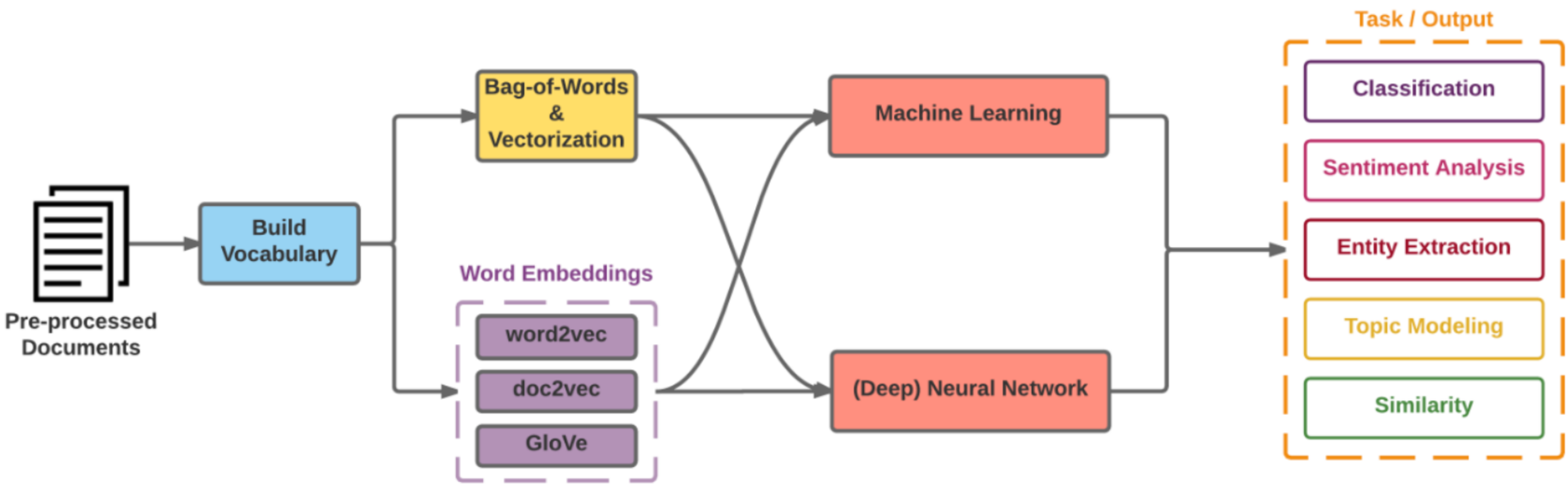
Word Embeddings

Word2Vec

GloVe

Modern NLP Pipeline





Source: https://github.com/fortiema/talks/blob/master/opendata2016sh/pragmatic-nlp-opendata2016sh.pdf

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

Source: Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching word vectors with subword information." *arXiv preprint arXiv:1607.04606* (2016).

Facebook Research FastText Word2Vec: wiki.zh.vec

(861MB) (332647 word 300 vec)

wiki.zh.vec

```
yg -0.3978 0.49084 -0.54621 0.078991 0.8584 -0.26163 -0.45787 0.060828 0.36513 -0.03771 0.80791 0.16613 1.4828 -0.89862 0.085965
迴圈 -0.034834 0.71651 -0.4377 0.48344 0.31117 -0.51783 -0.40156 -0.057097 0.31535 -0.088301 0.23436 0.30884 1.2932 -0.6704 0.21
ぶっ -0.23267 0.39349 -0.90806 -0.53805 0.59308 -0.31819 -0.64229 0.16871 0.10086 0.09342 1.0914 -0.16019 1.6954 -0.70604 -0.218
三公 0.54129 0.55641 -0.4348 0.25094 0.1631 -0.10326 -0.54099 0.064742 0.13175 0.10217 0.84938 -0.10287 1.312 -0.74969 0.24025 -
水貨 -0.14451 0.80455 -0.6145 0.55905 0.58307 -0.02559 -0.41088 -0.19056 -0.09178 0.33935 1.1927
刚才 0.19347 0.553 -0.64736 0.26358 0.83816 -0.24098 -0.83997 -0.16232 -0.024786 -0.2483 0.69732
無知 -0.0089777 0.90866 -0.25306 0.72983 0.67791 -0.3285 -0.63835 0.075295 0.4774 -0.04134 0.721
好轉 -0.026068 0.92676 -0.47469 0.50129 0.67343 -0.32509 -0.32917 0.066499 0.3875 0.0011722 0.66
紀事 0.40541 0.67654 -0.5351 0.30329 0.43042 -0.24675 -0.19287 0.34207 0.35516 -0.076331 0.85916
變回 -0.089933 0.88136 -0.43524 0.59963 0.6403 -0.70981 -0.56788 -0.074018 0.16905 -0.086594 0.6
牟尼 -0.26578 0.6434 0.028982 -0.044001 0.88297 -0.17646 -0.64672 0.040483 0.43653 0.084908 0.74
埋藏 -0.0985 0.85082 -0.33363 0.24784 0.71518 -0.59054 -0.73731 0.050949 0.36726 -0.076886 0.817
正大 0.21069 0.27605 -0.83862 -0.099698 0.47894 -0.32196 -0.38288 -0.01892 0.40548 -0.029619 0.7
kis -0.30595 0.18482 -0.71287 -0.314 0.44776 -0.44245 -0.36447 -0.23723 0.00098801 -0.2528 0.608
合奏 0.1841 0.60874 -0.51376 -0.48002 0.21506 -0.55515 -0.71746 0.030735 0.39508 -0.40856 0.6226
精兵 0.25619 0.77186 -0.48847 0.23118 0.27254 0.21305 -0.3517 0.47305 0.24882 -0.34756 1.025 0.18
疲勞 -0.072521 1.0381 -0.51933 0.19421 0.67573 -0.45204 -0.20126 0.22704 0.44196 0.018401 0.3473
襯 -0.11771 1.4272 -1.0849 0.77532 0.87026 -0.6892 -0.3521 0.036517 0.42727 -0.1871 0.82789 -0.0
小貓 -0.21554 0.73988 -0.39628 0.044656 1.0602 -0.67047 -0.54102 0.11888 0.1693 0.19343 1.0841 0.
lai -0.25451 0.31596 -0.29228 -0.19144 0.99059 -0.24459 -0.66342 0.063093 -0.061142 -0.22749 0.6
偏東 -0.50835 1.0943 0.043918 0.29173 1.0161 -0.32493 -0.27305 0.026946 0.46811 -0.3874 1.4049 0.
大约是 -0.35726 -0.03476 -0.28672 0.075447 0.18175 -0.39421 -0.32088 0.025225 0.34808 0.074744 0.
franch -0.6046 -0.3235 0.024041 -0.2756 0.74761 -0.14654 0.0082566 -0.10071 0.53593 -0.17374 0.2
brazilian -0.54029 -0.63905 -0.094006 -0.68768 0.33263 -0.1583 -0.060424 0.20644 0.46234 -0.0764
夹竹桃 -0.4361 0.011429 -0.078896 -0.078186 0.37747 -0.052101 -0.096683 0.10769 0.62661 -0.37252
continent -0.37761 -0.72151 -0.42248 -0.81768 0.5016 -0.48569 0.13464 0.12644 0.32292 0.18099 0.
我还是 0.097443 0.28929 -0.14202 0.034027 0.50621 -0.1647 -0.45849 -0.16198 0.13965 -0.33451 0.61
vienna -0.25827 -0.050966 0.050502 -0.63466 0.4949 -0.17448 -0.59978 0.20269 0.37532 0.059419 0.
固态 -0.12678 0.4556 -0.27108 0.12506 0.52106 -0.058477 -0.69296 0.12162 0.26508 -0.089028 0.752
吉普 -0.33693 0.48335 -0.58455 0.13722 0.74856 -0.24529 -0.41125 -0.13832 0.33871 -0.12051 0.864
實物 0.030096 0.65756 -0.67982 0.2203 0.38492 -0.19001 -0.53136 -0.10322 0.24523 0.15287 0.92591
教职 0.11559 0.67087 -0.5111 0.14955 0.61417 -0.51571 -0.47901 0.29445 0.37629 -0.24232 0.4608 -
愓 0.50469 1.5357 -0.64393 0.48668 0.69479 -0.23443 -0.47863 0.16288 0.3347 -0.51673 0.86777 0.0
岸上 0.088323 0.85815 -0.485 0.30383 0.75965 -0.25031 -0.76678 0.12805 0.37641 -0.088752 0.65012
议和 0.26835 0.94854 -0.27972 0.097623 0.43305 -0.031361 -0.57406 0.21608 0.3324 -0.36823 0.6987
aka -0.21332 0.11216 -0.48872 -0.18531 0.79093 -0.34221 -0.51122 0.10067 0.29963 -0.075253 0.642
滑鐵盧 -0.28726 0.88014 -0.39751 -0.056992 0.37408 -0.16967 -0.20673 -0.048533 -0.1978 -0.13107 0
```

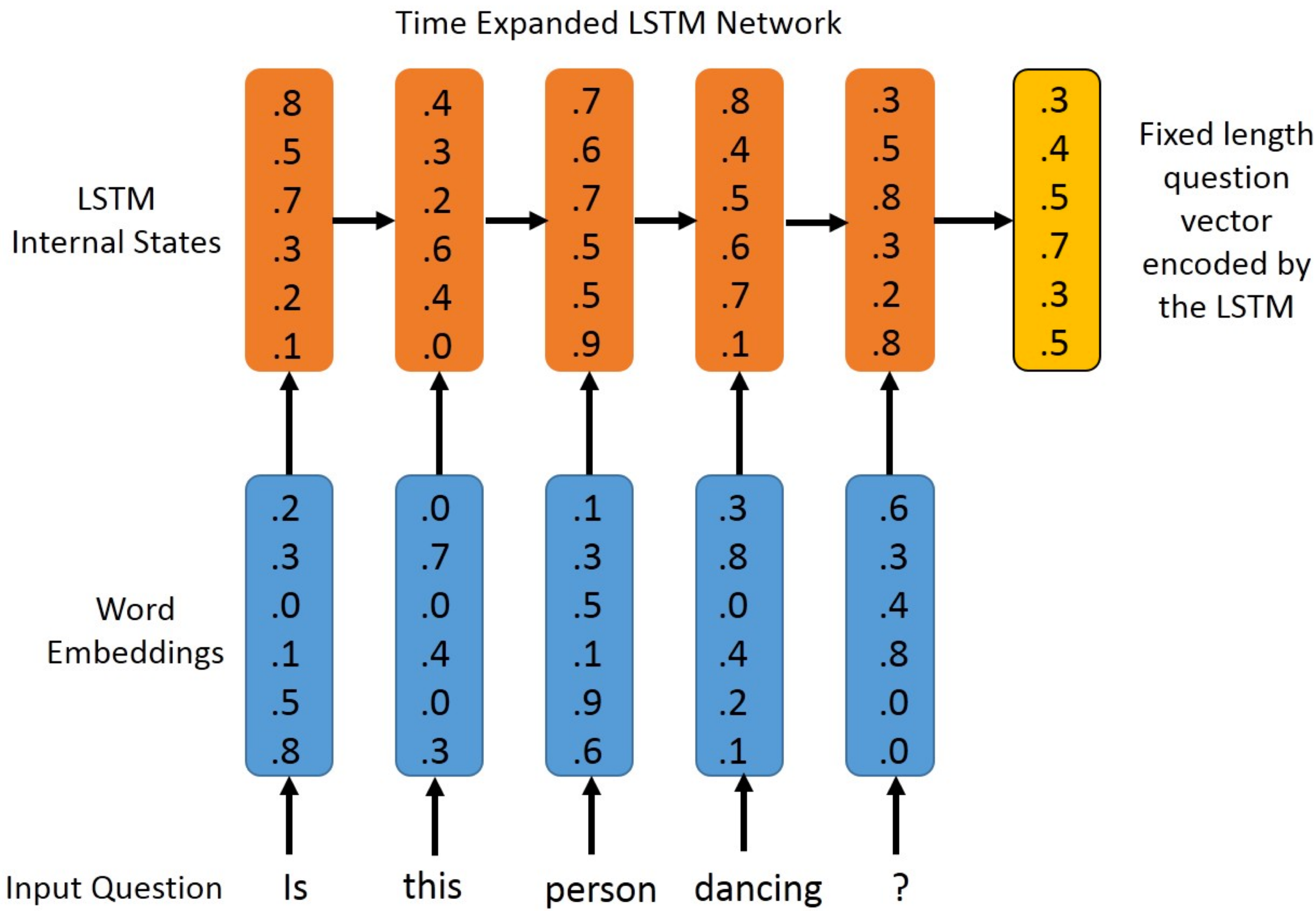
Models

The models can be downloaded from:

- Afrikaans: *bin+text*, *text*
- Albanian: *bin+text*, *text*
- Arabic: *bin+text*, *text*
- Armenian: *bin+text*, *text*
- Asturian: *bin+text*, *text*
- Azerbaijani: *bin+text*, *text*
- Bashkir: *bin+text*, *text*
- Basque: *bin+text*, *text*
- Belarusian: *bin+text*, *text*
- Bengali: *bin+text*, *text*
- Bosnian: *bin+text*, *text*
- Breton: *bin+text*, *text*
- Bulgarian: *bin+text*, *text*
- Burmese: *bin+text*, *text*
- Catalan: *bin+text*, *text*
- Cebuano: *bin+text*, *text*
- Chechen: *bin+text*, *text*
- Chinese: *bin+text*, *text*
- Chuvash: *bin+text*, *text*
- Croatian: *bin+text*, *text*
- Czech: *bin+text*, *text*

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

Word Embeddings in LSTM RNN



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

	SPACY	SYNTAXNET	NLTK	CORENLP
Easy installation	+	−	+	+
Python API	+	−	+	−
Multi-language support	○	+	+	+
Tokenization	+	+	+	+
Part-of-speech tagging	+	+	+	+
Sentence segmentation	+	+	+	+
Dependency parsing	+	+	−	+
Entity Recognition	+	−	+	+
Integrated word vectors	+	−	−	−
Sentiment analysis	+	−	+	+
Coreference resolution	−	−	−	+

Source: 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

SYSTEM	LANGUAGE	ACCURACY	SPEED (WPS)
spaCy	**Cython**	**91.8**	**13,963**
ClearNLP	Java	91.7	10,271
CoreNLP	Java	89.6	8,602
MATE	Java	**92.5**	550
Turbo	C++	92.4	349

Source: https://spacy.io/docs/api/

Processing Speed of NLP libraries

	ABSOLUTE (MS PER DOC)			RELATIVE (TO SPACY)		
SYSTEM	TOKENIZE	TAG	PARSE	TOKENIZE	TAG	PARSE
spaCy	**0.2ms**	**1ms**	**19ms**	**1x**	**1x**	**1x**
CoreNLP	2ms	10ms	49ms	10x	10x	2.6x
ZPar	1ms	8ms	850ms	5x	8x	44.7x
NLTK	4ms	443ms	n/a	20x	443x	n/a

Source: https://spacy.io/docs/api/

Google SyntaxNet (2016): Best Syntactic Dependency Parsing Accuracy

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

Source: https://spacy.io/docs/api/

Named Entity Recognition (NER)

SYSTEM	PRECISION	RECALL	F-MEASURE
spaCy	0.7240	0.6514	0.6858
CoreNLP	**0.7914**	**0.7327**	**0.7609**
NLTK	0.5136	0.6532	0.5750
LingPipe	0.5412	0.5357	0.5384

Natural Language Processing with Python

Source: https://www.python.org/community/logos/

Python in Google Colab

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

python101.ipynb

File Edit View Insert Runtime Tools Help

COMMENT SHARE

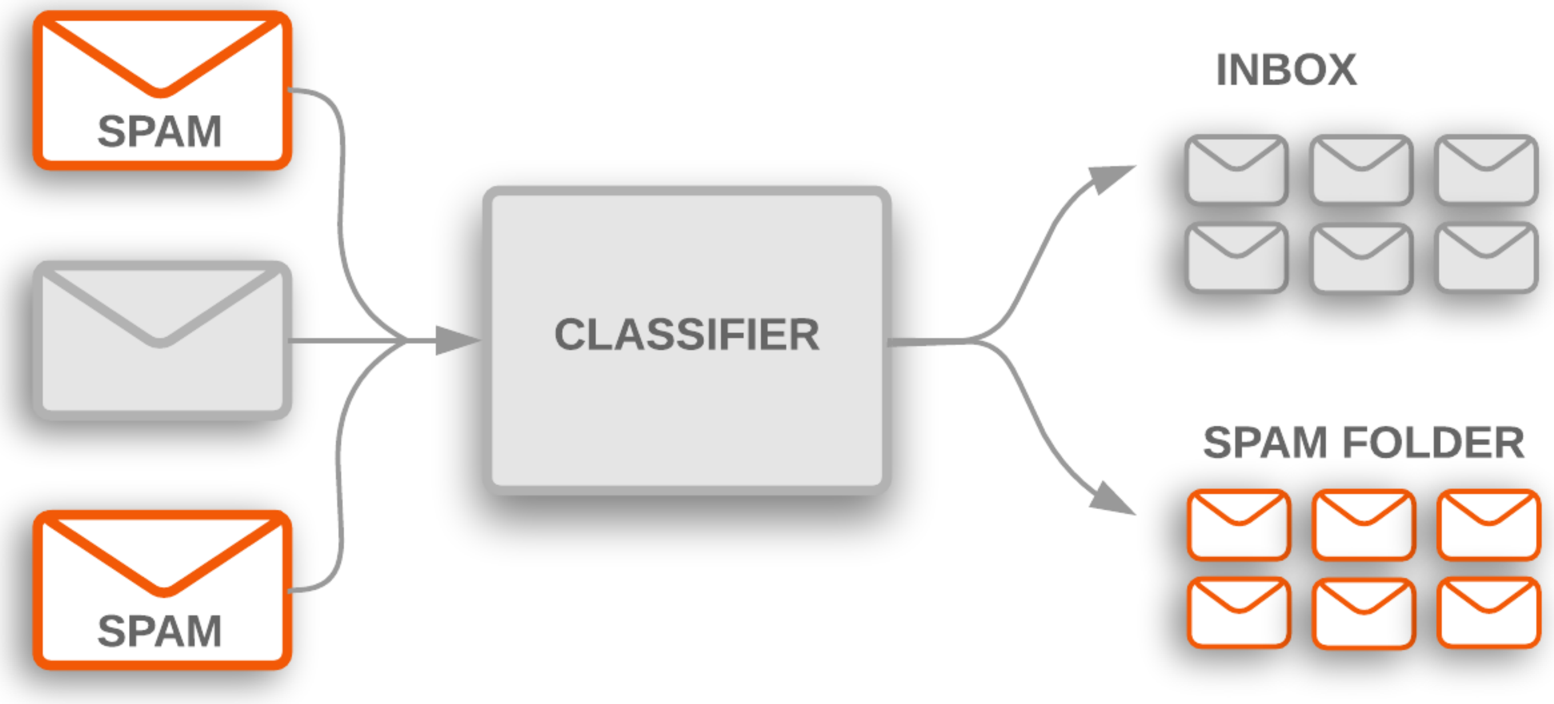
CODE TEXT CELL CELL CONNECT EDITING

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)
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)
```

```
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. 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.]]
```

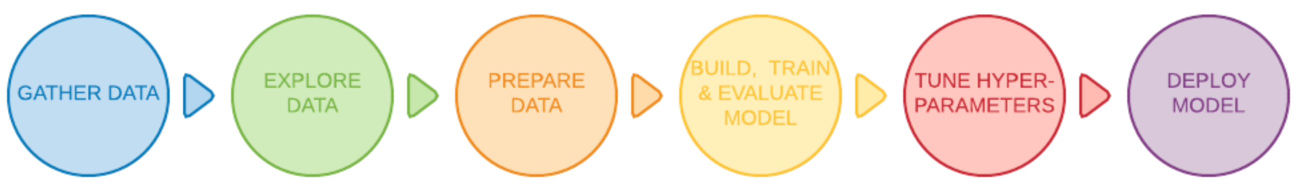
Text Classification



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

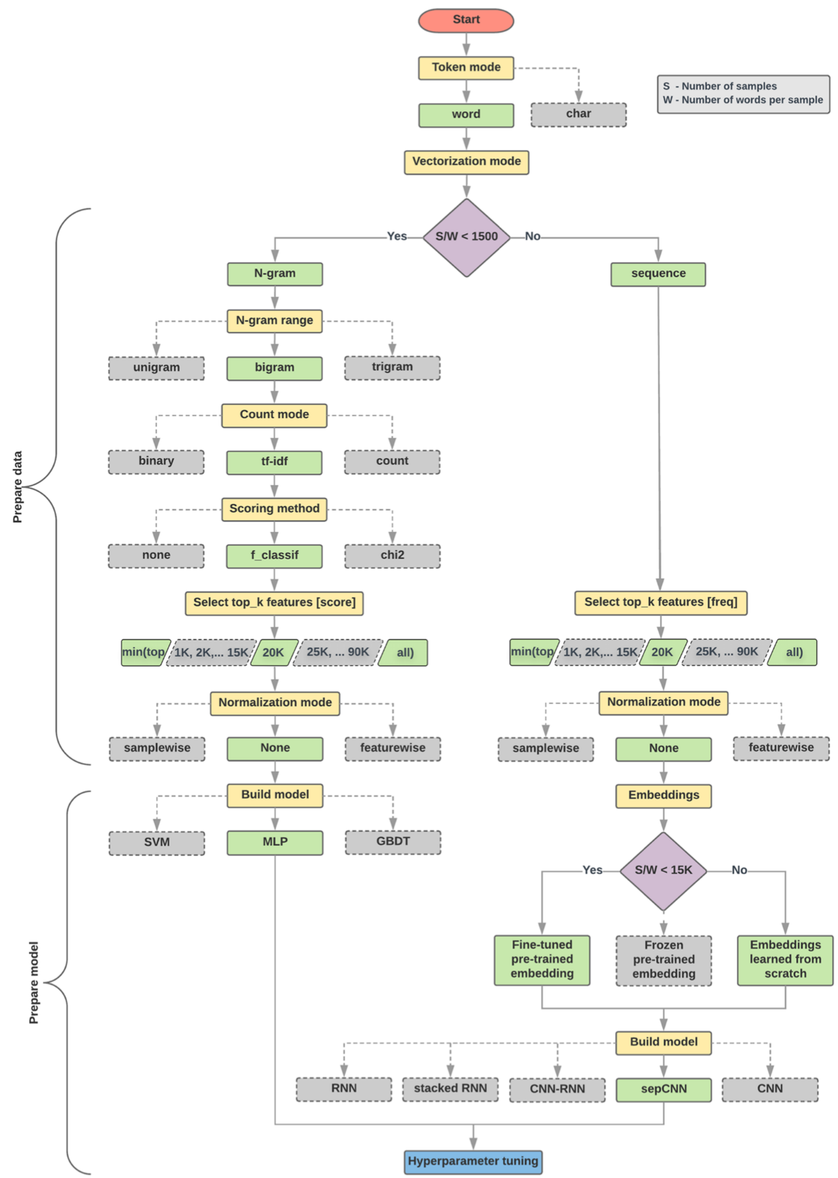
Text Classification Workflow

- 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



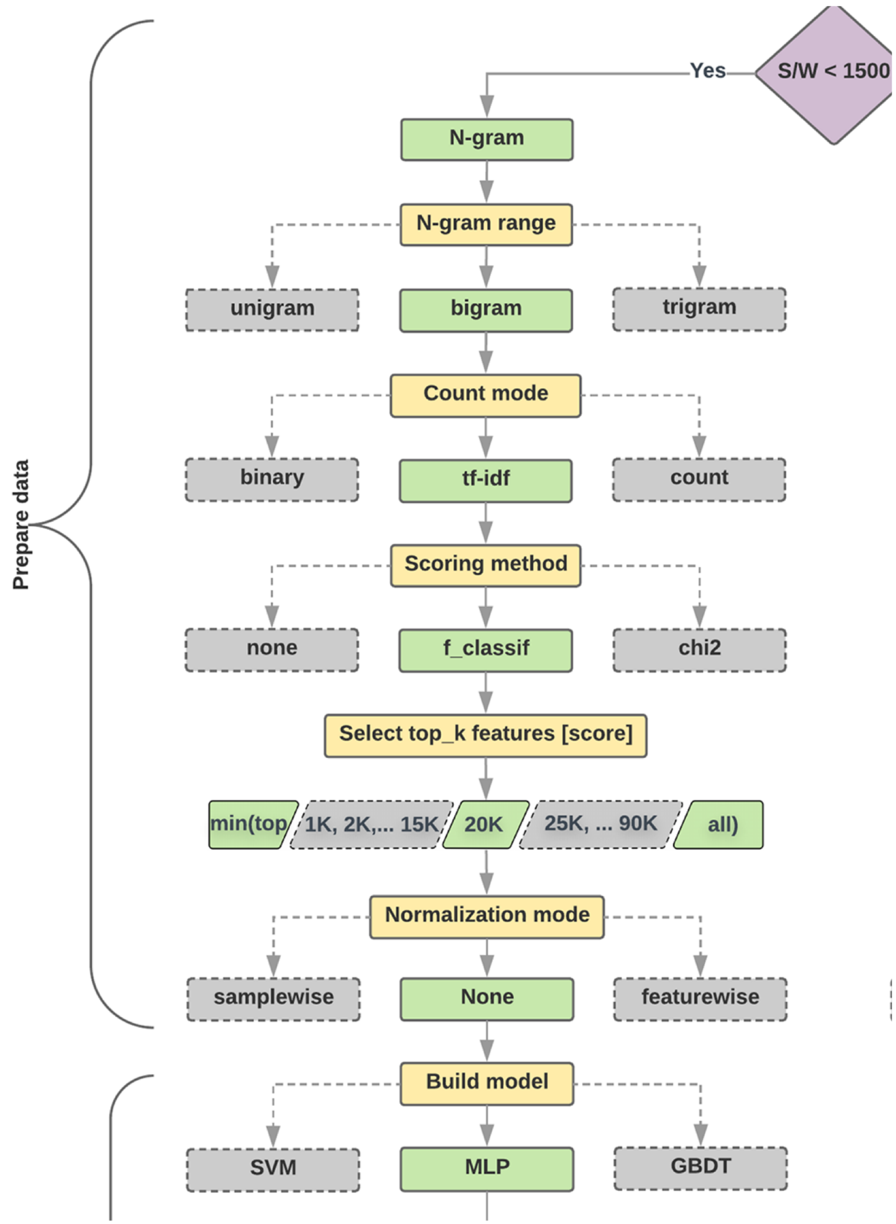
Source: https://developers.google.com/machine-learning/guides/text-classification/

Text Classification Flowchart



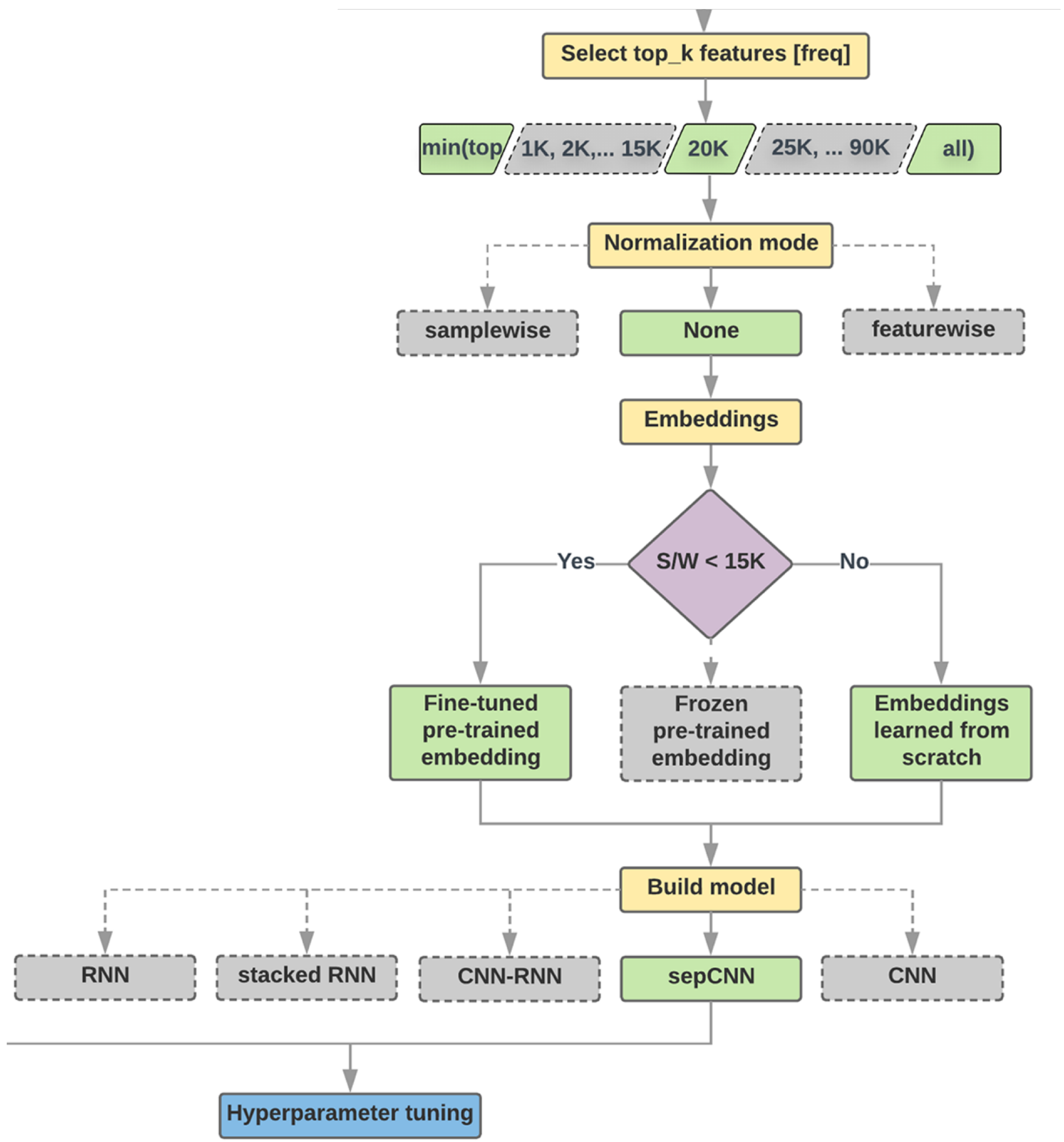
Source: https://developers.google.com/machine-learning/guides/text-classification/step-2-5

Text Classification S/W<1500: N-gram



Source: https://developers.google.com/machine-learning/guides/text-classification/step-2-5

Text Classification S/W>=1500: Sequence

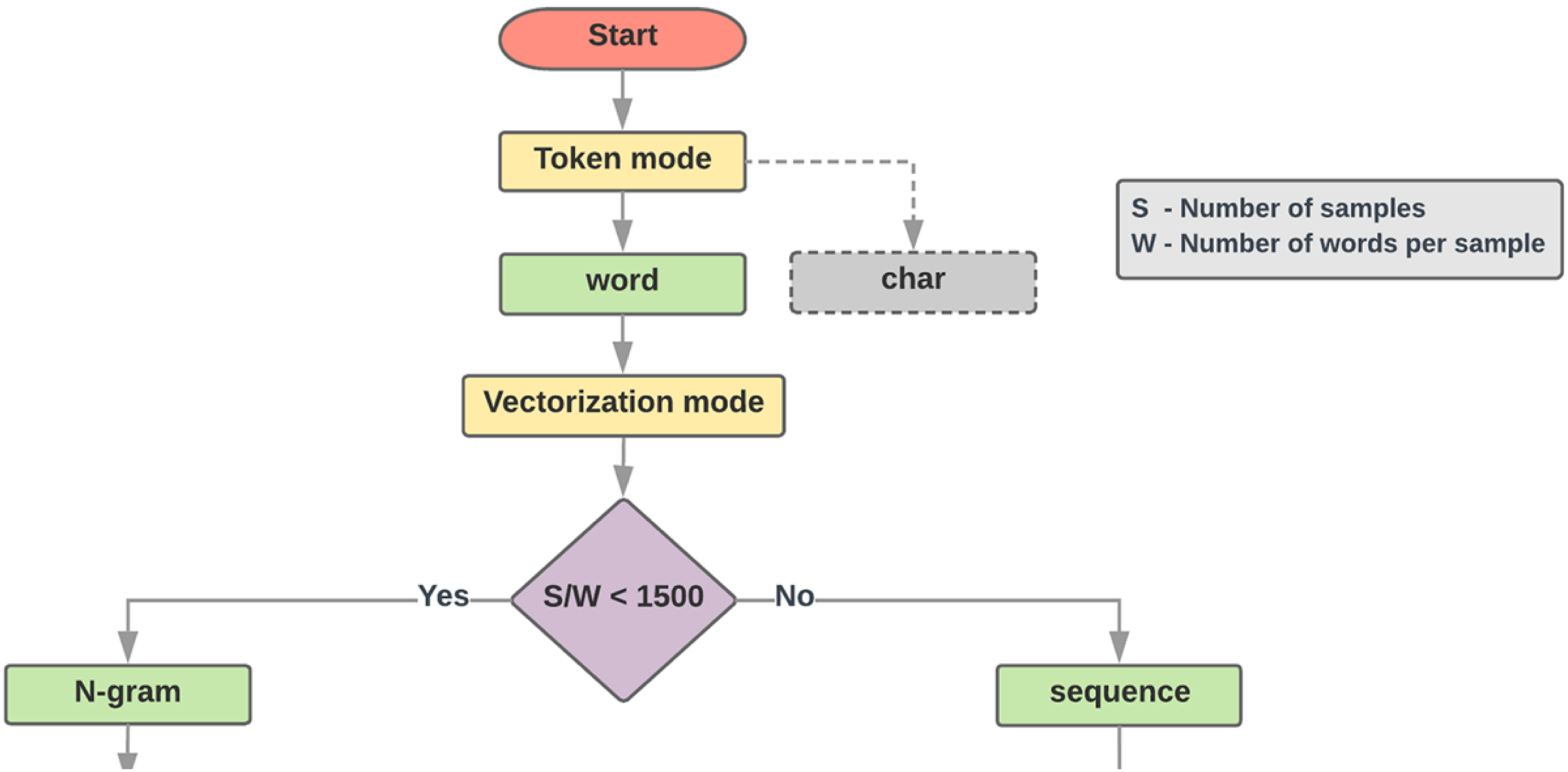


Source: https://developers.google.com/machine-learning/guides/text-classification/step-2-5

Step 2.5: Choose a Model

Samples/Words < 1500

150,000/100 = 1500



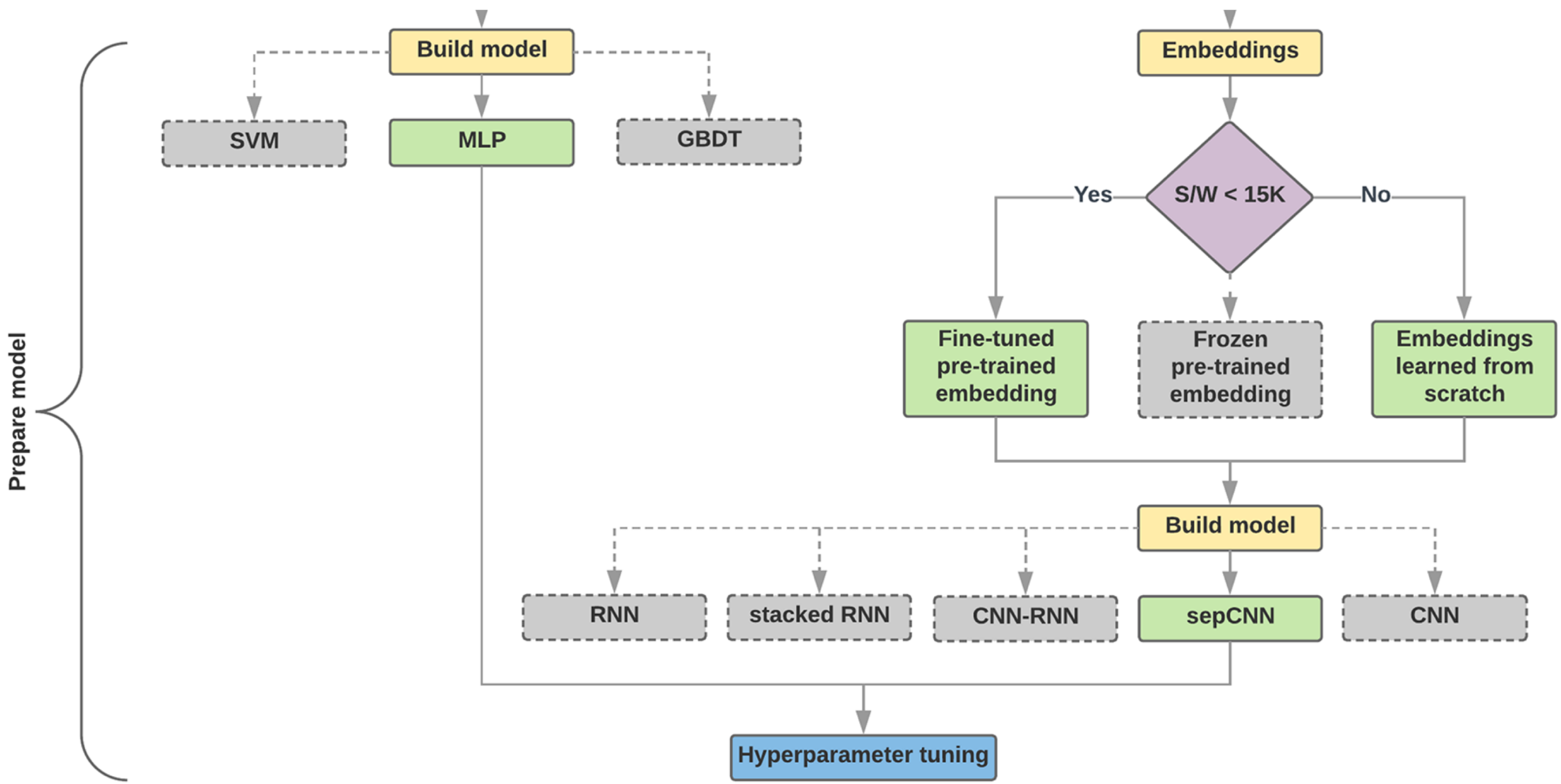
IMDb review dataset,
the samples/words-per-sample ratio is ~ 144

Source: https://developers.google.com/machine-learning/guides/text-classification/step-2-5

Step 2.5: Choose a Model

Samples/Words < 15,000

1,500,000/100 = 15,000



Source: https://developers.google.com/machine-learning/guides/text-classification/step-2-5

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]
```

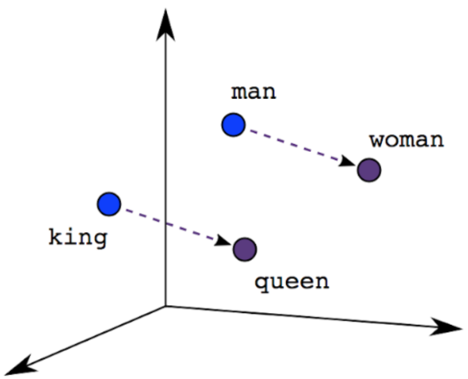
Source: https://developers.google.com/machine-learning/guides/text-classification/step-3

One-hot encoding

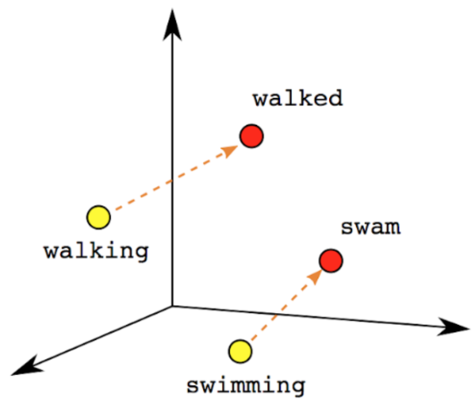
```
                  'The mouse ran up the clock' =

The       1       [ [0, 1, 0, 0, 0, 0, 0],
mouse     2         [0, 0, 1, 0, 0, 0, 0],
ran       3         [0, 0, 0, 1, 0, 0, 0],
up        4         [0, 0, 0, 0, 1, 0, 0],
the       1         [0, 1, 0, 0, 0, 0, 0],
clock     5         [0, 0, 0, 0, 0, 1, 0] ]

                    [0, 1, 2, 3, 4, 5, 6]
```

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

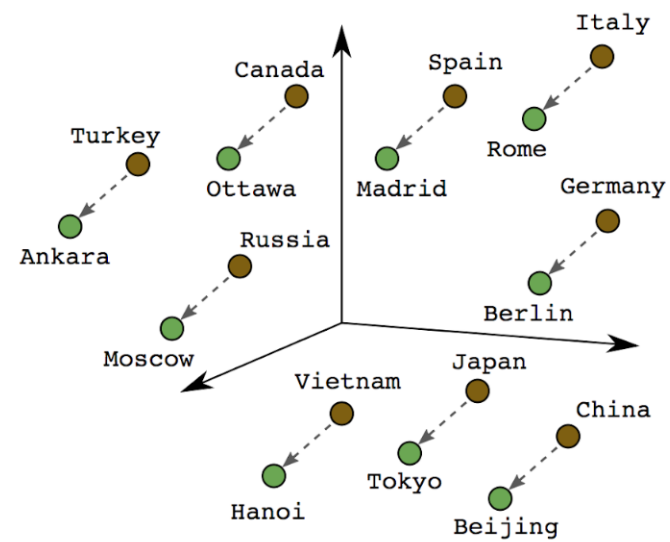
Word embeddings



Male-Female



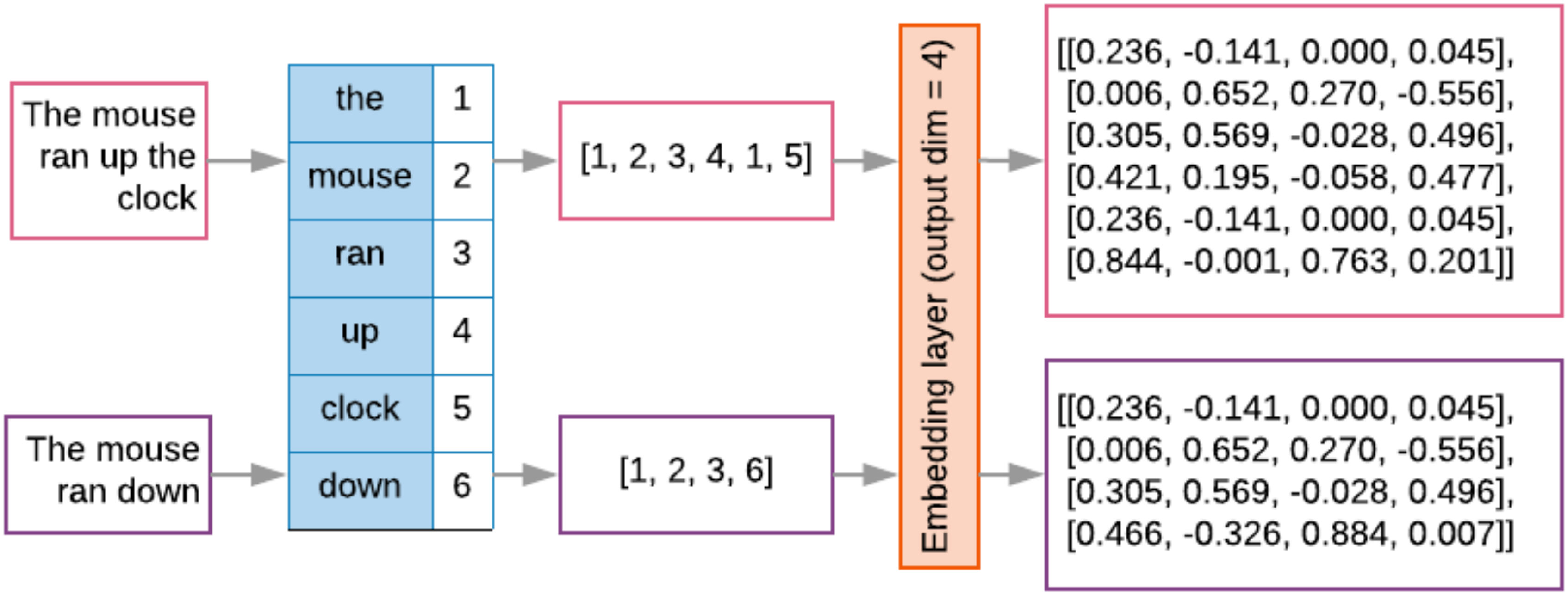
Verb Tense



Country-Capital

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

Word embeddings



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
sortedset = sorted(set(terms))
print('terms =', terms)
print('sortedset =', sortedset)
```


```
terms = ['the', 'mouse', 'ran', 'up', 'the', 'clock', 'the', 'mouse', 'ran', 'down']
sortedset = ['clock', 'down', 'mouse', 'ran', 'the', 'up']
```

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

```python
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']
```

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

```
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/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

```
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)
```

```
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/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

from keras.preprocessing.text import Tokenizer

```python
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)
```

```
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), ('ni
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

```
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/

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/

t.texts_to_matrix(docs, mode='tfidf')

```
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)
```

```
texts_to_matrix:
[[0. 0. 1.25276297 1.25276297 0. 0. 0. 0. 0. ]
[0. 0.98082925 0. 0. 1.25276297 0. 0. 0. 0. ]
[0. 0. 0. 0. 0. 1.25276297 1.25276297 0. 0. ]
[0. 0.98082925 0. 0. 0. 0. 0. 1.25276297 0. ]
[0. 0. 0. 0. 0. 0. 0. 0. 1.25276297]]
```

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

NLTK (Natural Language Toolkit)

NLTK 3.0 documentation

NEXT | MODULES | INDEX

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:

```
>>> import nltk
```

TABLE OF CONTENTS

SEARCH

Go

Enter search terms or a module, class or function name.

http://www.nltk.org/

conda list

jupyter

nltk 3.2.2 py36_0

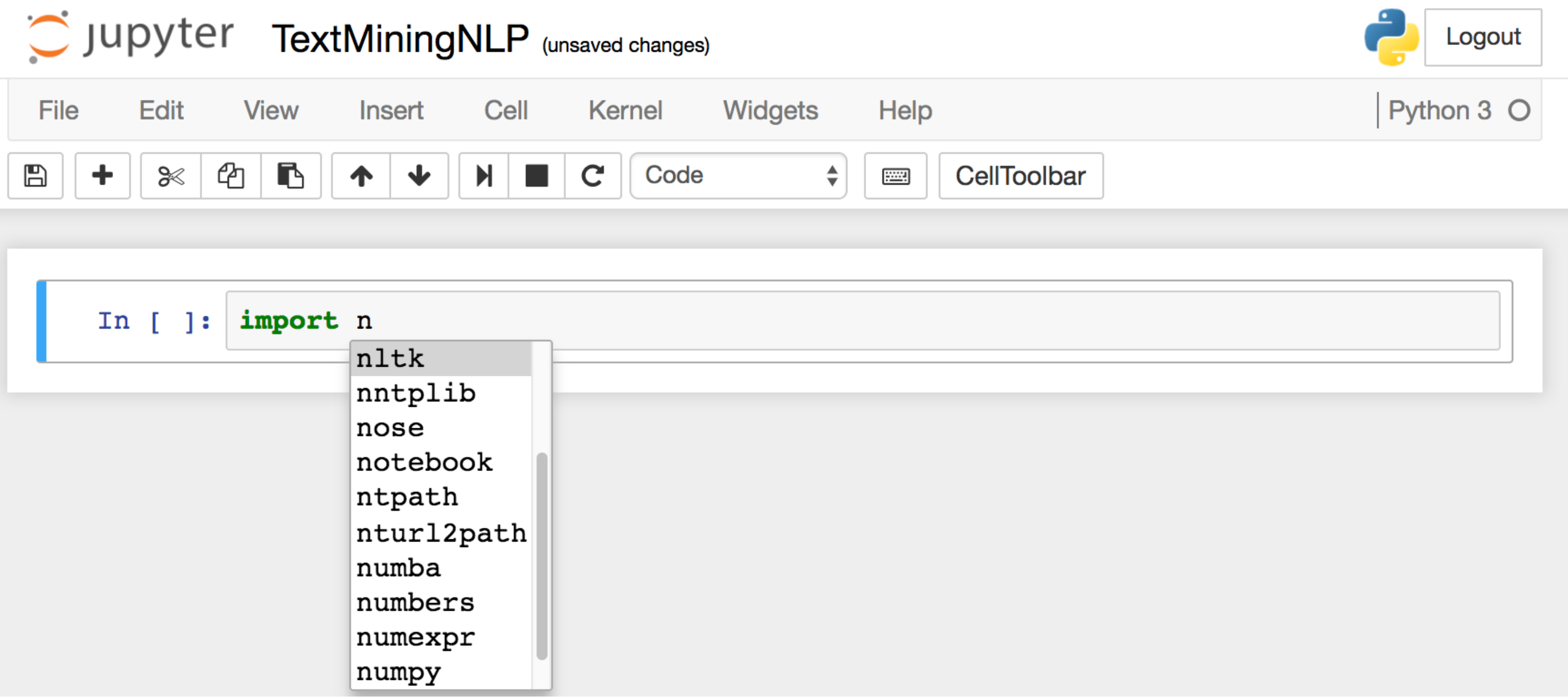
```
matplotlib            2.0.0          np111py36_0
mistune               0.7.3               py36_1
mkl                   2017.0.1                 0
mkl-service           1.1.2               py36_3
mpmath                0.19                py36_1
multipledispatch      0.4.9               py36_0
nbconvert             4.2.0               py36_0
nbformat              4.2.0               py36_0
networkx              1.11                py36_0
nltk                  3.2.2               py36_0
nose                  1.3.7               py36_1
notebook              4.3.1               py36_0
numba                 0.30.1         np111py36_0
numexpr               2.6.1          np111py36_2
numpy                 1.11.3              py36_0
numpydoc              0.6.0               py36_0
odo                   0.5.0               py36_1
openpyxl              2.4.1               py36_0
openssl               1.0.2k                   1
pandas                0.19.2         np111py36_1
pandas-datareader     0.2.1               py36_0
partd                 0.3.7               py36_0
path.py               10.0                py36_0
pathlib2              2.2.0               py36_0
patsy                 0.4.1               py36_0
pep8                  1.7.0               py36_0
pexpect               4.2.1               py36_0
pickleshare           0.7.4               py36_0
pillow                4.0.0               py36_0
pip                   9.0.1               py36_1
plotly                1.12.9              py36_0
ply                   3.9                 py36_0
prompt_toolkit        1.0.9               py36_0
psutil                5.0.1               py36_0
```

help('modules')

```
In [2]: help('modules')
```

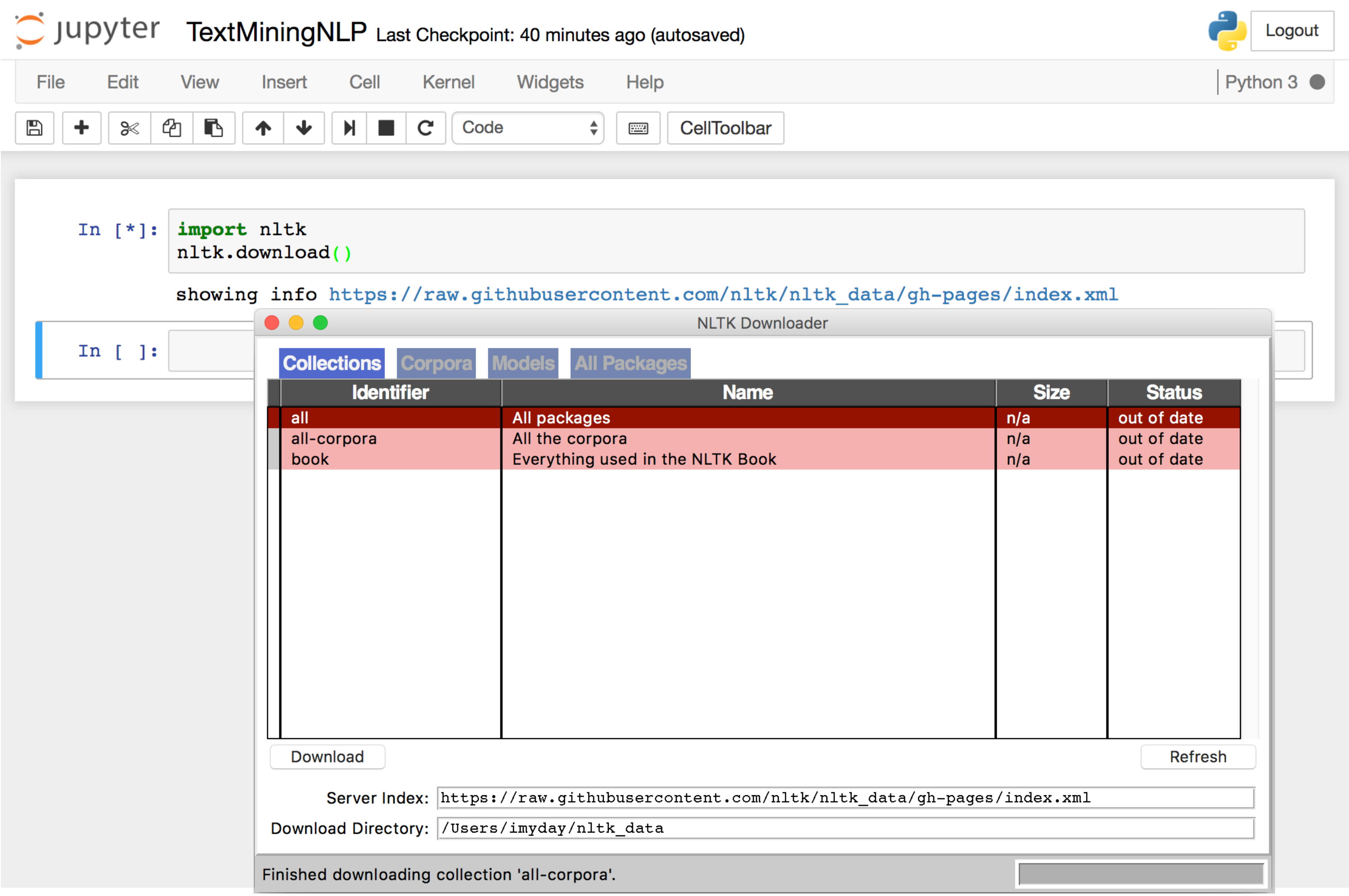
```
_Qt                 copy                nltk                tarfile
_Res                copy_reg            nntplib             telnetlib
_Scrap              copyreg             nose                tempfile
_Snd                crypt               notebook            terminado
_TE                 cryptography        ntpath              terminalcommand
_Win                csv                 nturl2path          termios
__builtin__         ctypes              numba               test_path
__future__          curl                numbers             test_pycosat
_abcoll             curses              numexpr             tests
_ast                cycler              numpy               textwrap
_bisect             cython              odo                 this
_builtinSuites      cythonmagic         opcode              thread
_cffi_backend       cytoolz             openpyxl            threading
_codecs             datashape           operator            time
_codecs_cn          datetime            optparse            timeit
_codecs_hk          dateutil            os                  tkColorChooser
_codecs_iso2022     dbhash              os2emxpath          tkCommonDialog
_codecs_jp          dbm                 osax                tkFileDialog
_codecs_kr          decimal             pandas              tkFont
_codecs_tw          decorator           parser              tkMessageBox
```

import nltk



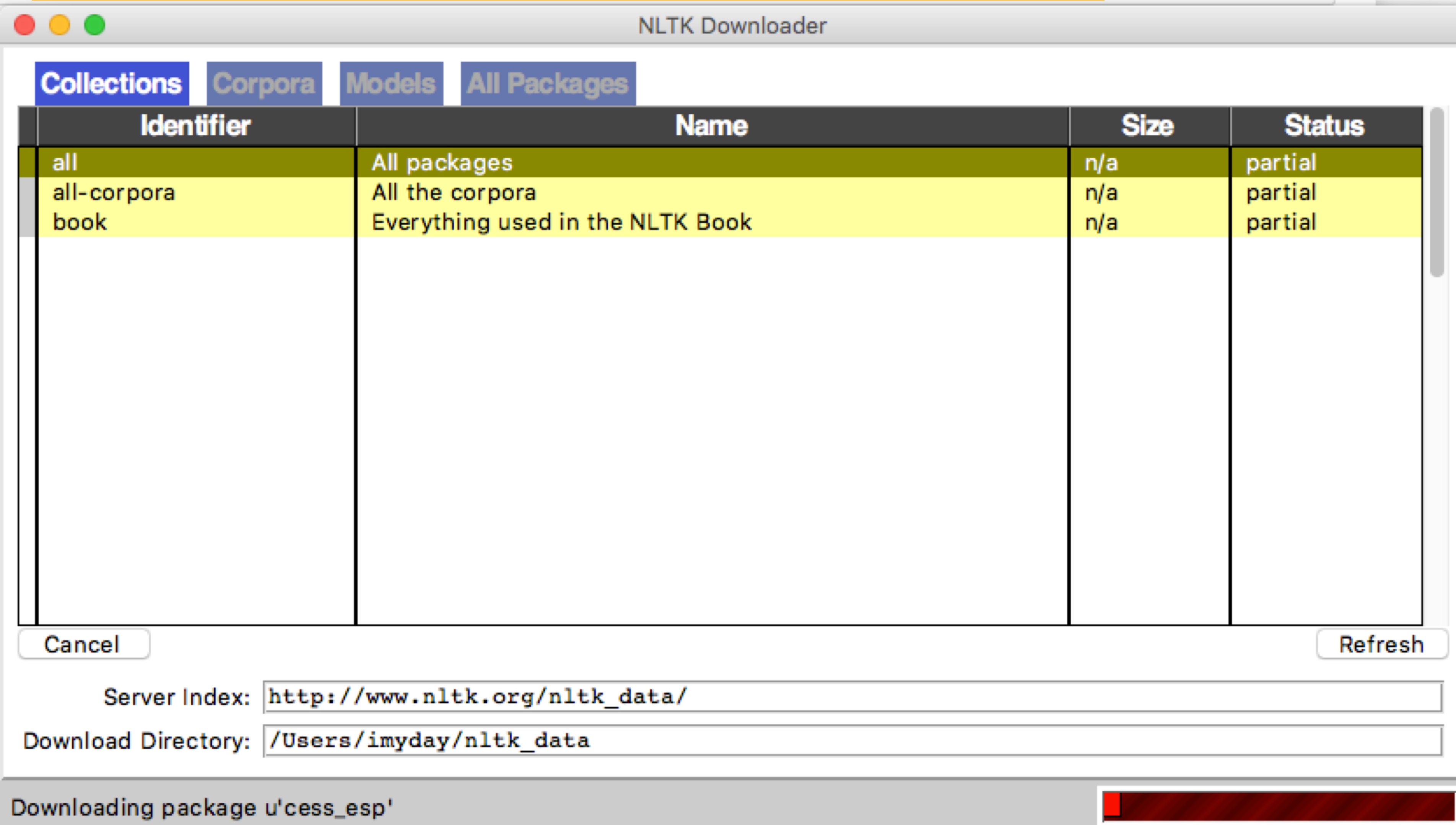
Source: http://www.nltk.org/

```
import nltk
nltk.download()
```



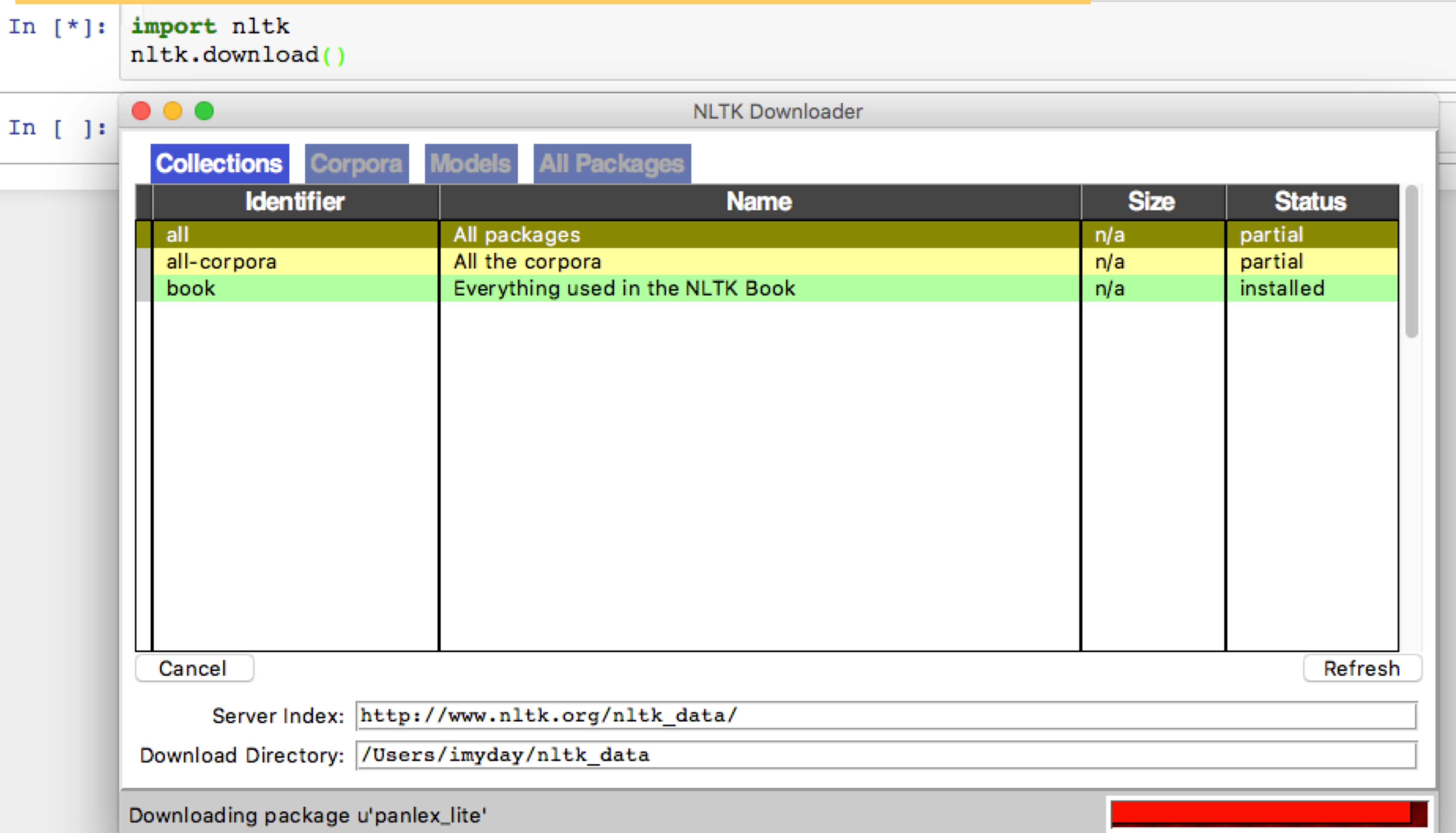
Source: http://www.nltk.org/

```
import nltk
nltk.download()
```

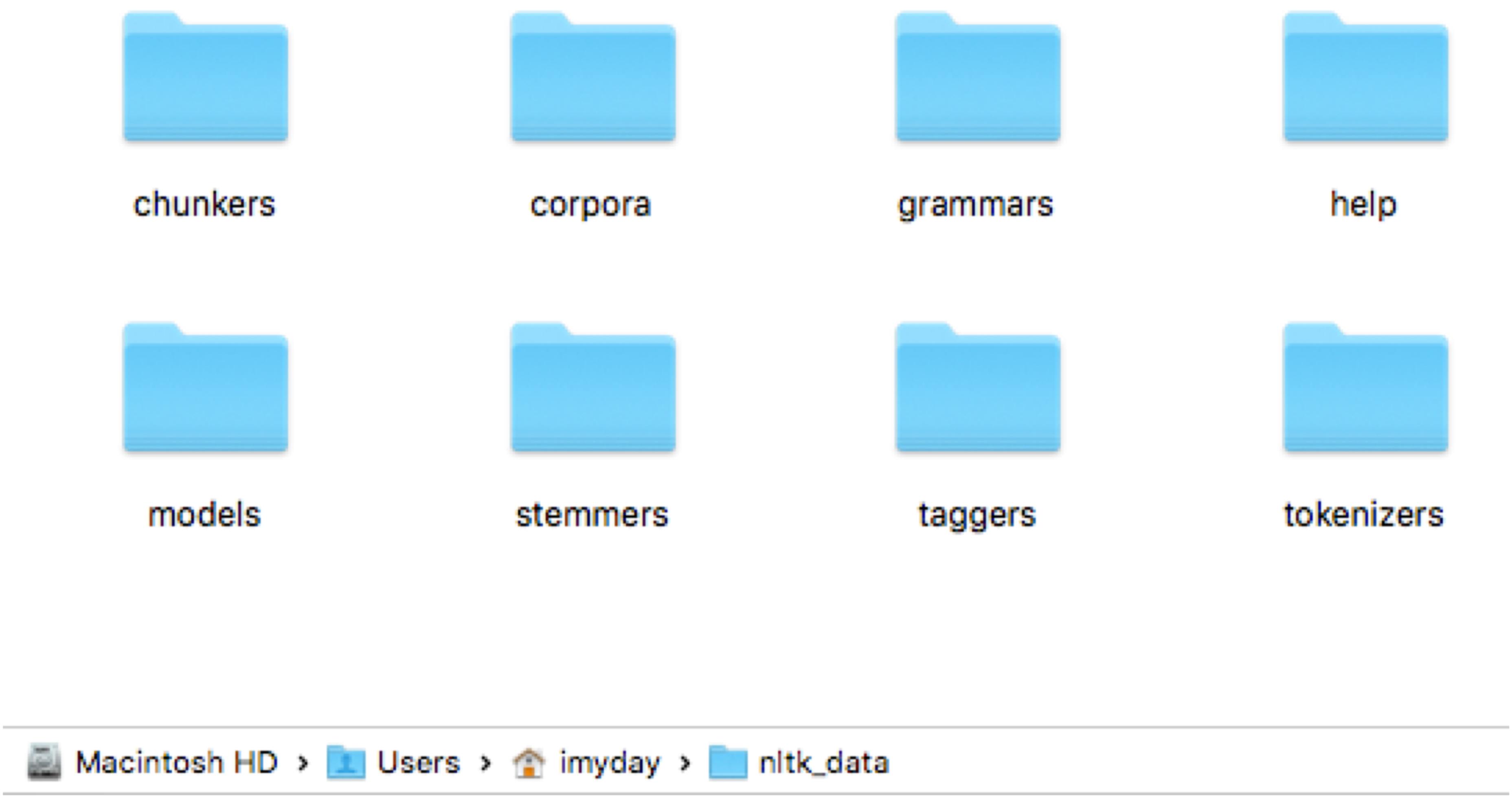


Source: http://www.nltk.org/

```python
import nltk
nltk.download()
```



nltk_data


chunkers
corpora
grammars
help
models
stemmers
taggers
tokenizers
Macintosh HD › Users › imyday › nltk_data

At eight o'clock on Thursday morning Arthur didn't feel very good.

Source: http://www.nltk.org/

```
[('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'),
('.', '.')]
```

Source: http://www.nltk.org/

```
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', 'ver
        y', 'good', '.']
```

Source: 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')]
```

Source: http://www.nltk.org/

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'),
         ('.', '.')]
```

Source: http://www.nltk.org/

print(tagged)

```
In [5]: print(tagged)

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

```
[('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.

Source: http://www.nltk.org/

```
entities = nltk.chunk.ne_chunk(tagged)
entities
```


```
entities = nltk.chunk.ne_chunk(tagged)
entities
```

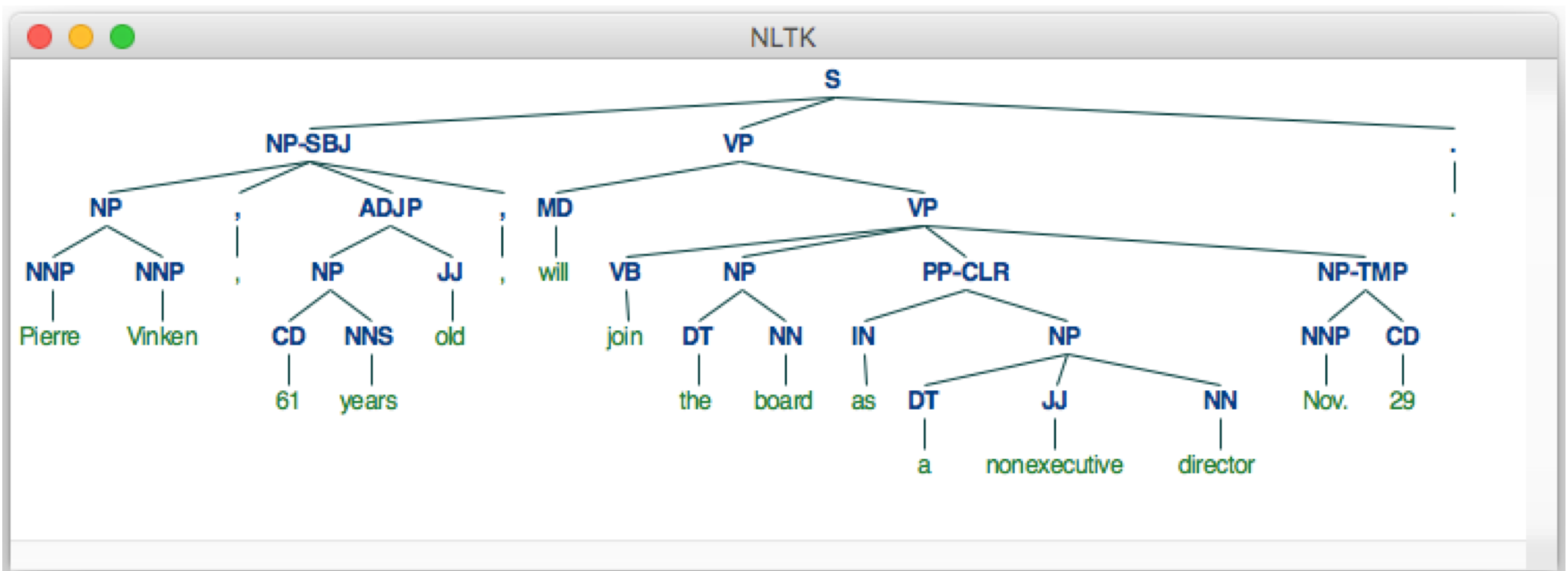

```
Tree('S', [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'NN'), ('on', 'IN'), ('Thursday', 'NN
P'), ('morning', 'NN'), Tree('PERSON', [('Arthur', 'NNP')]), ('did', 'VBD'), ("n't", 'RB'),
('feel', 'VB'), ('very', 'RB'), ('good', 'JJ'), ('.', '.')])
```

```
Tree('S', [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'),
           ('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/

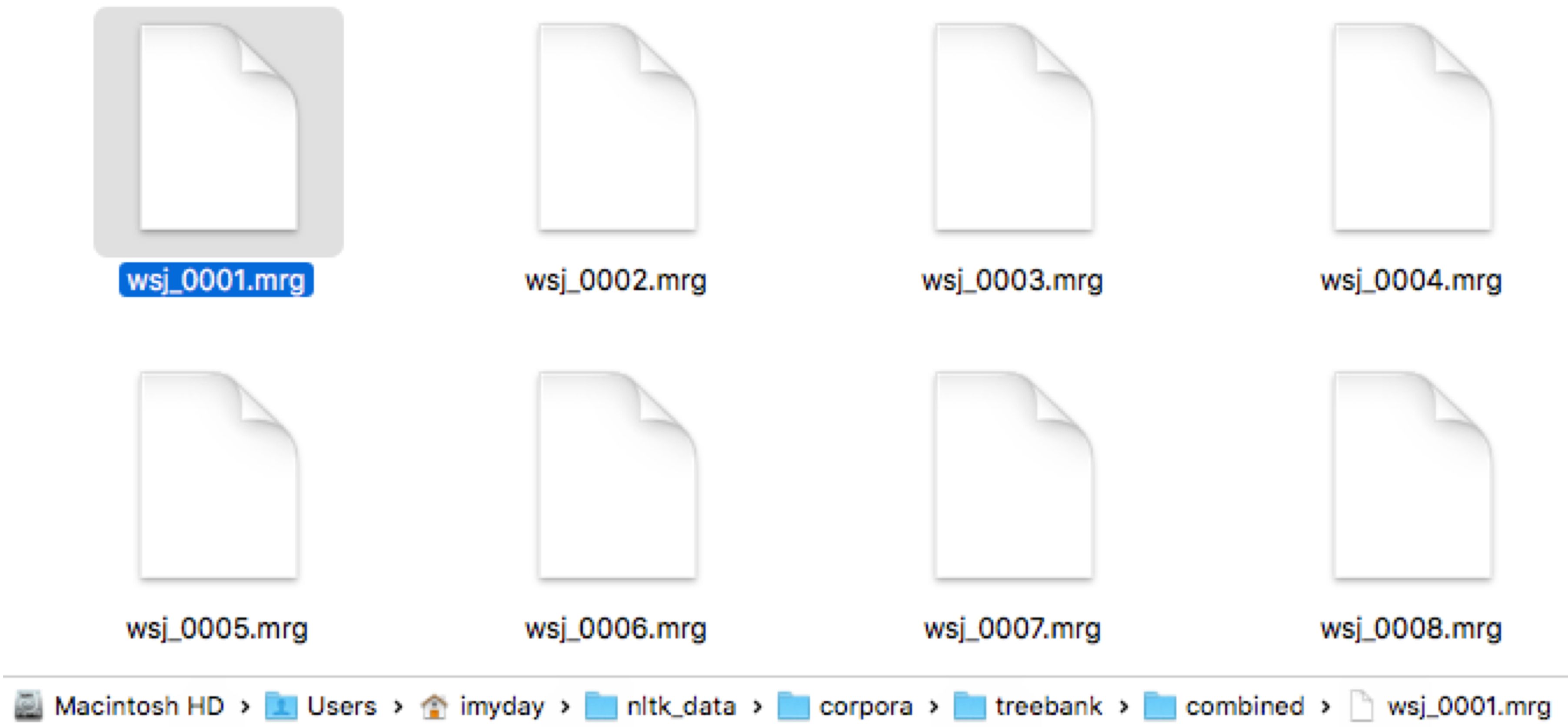
```
from nltk.corpus import treebank
t = treebank.parsed_sents('wsj_0001.mrg')[0]
t.draw()
```


```
from nltk.corpus import treebank
t = treebank.parsed_sents('wsj_0001.mrg')[0]
t.draw()
```

Source: http://www.nltk.org/

wsj_0001.mrg


wsj_0001.mrg
wsj_0002.mrg
wsj_0003.mrg
wsj_0004.mrg
wsj_0005.mrg
wsj_0006.mrg
wsj_0007.mrg
wsj_0008.mrg
Macintosh HD › Users › imyday › nltk_data › corpora › treebank › combined › wsj_0001.mrg

wsj_0001.mrg

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) (NN director) ))
        (NP-TMP (NNP Nov.) (CD 29) )))
    (. .) ))
( (S
    (NP-SBJ (NNP Mr.) (NNP Vinken) )
    (VP (VBZ is)
      (NP-PRD
        (NP (NN chairman) )
        (PP (IN of)
          (NP
            (NP (NNP Elsevier) (NNP N.V.) )
            (, ,)
            (NP (DT the) (NNP Dutch) (VBG publishing) (NN group) )))))
    (. .) ))

```

Source: http://www.nltk.org/

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 necesarry models for TextBlob's NLTK hooks to function > 'python -m textbl
ob.download_corpora'

import json
import multiprocessing
import regex as re
```

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

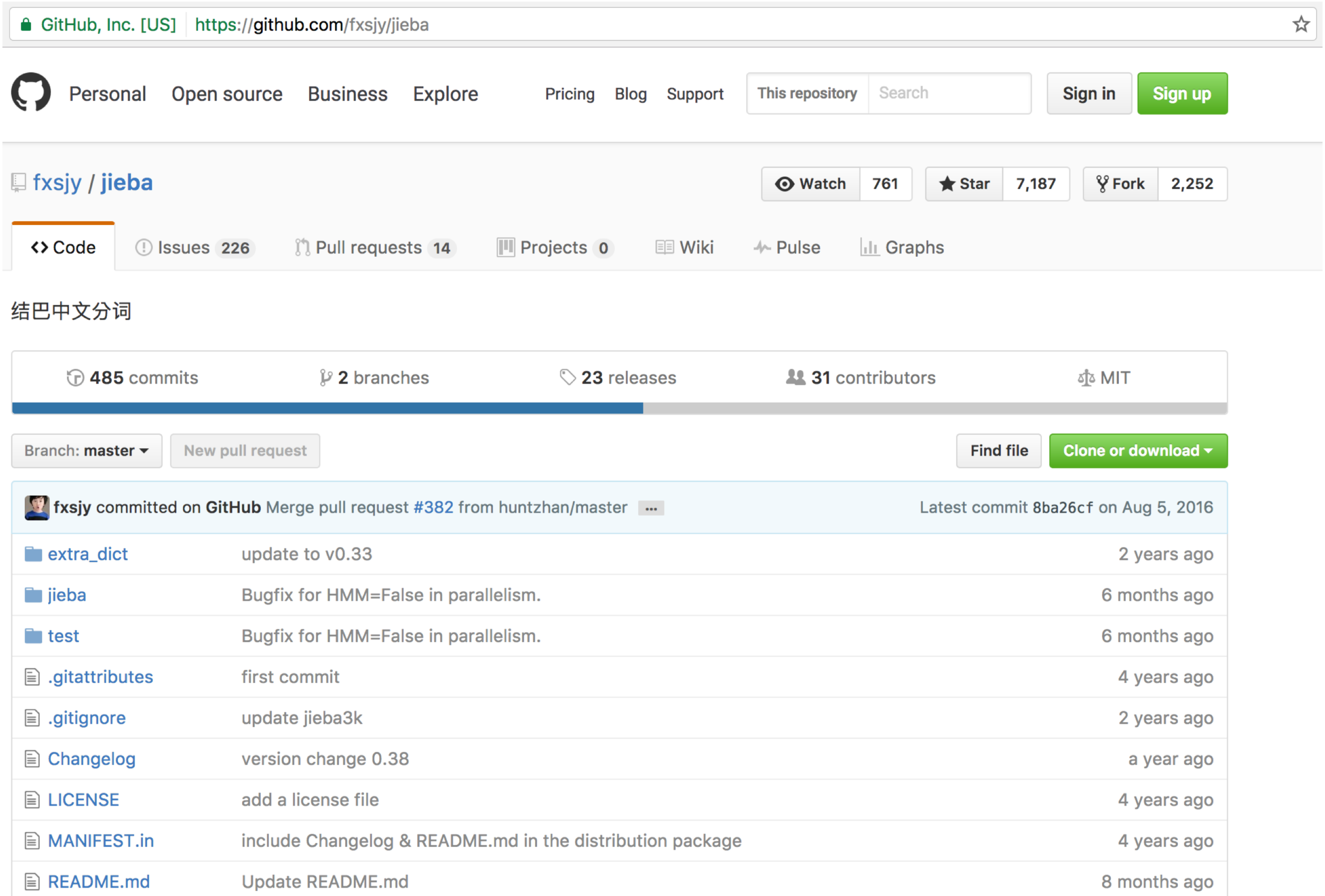
1. Ingest Data

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



https://github.com/fxsjy/jieba

Python Jieba “结巴”中文分词

```
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)

```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))   #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)
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.

Source: http://fortune.com/ai-artificial-intelligence-deep-machine-learning/

TensorFlow NLP Examples

- Basic Text Classification
  (Text Classification) (46 Seconds)
  - https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/basic_text_classification.ipynb
- NMT with Attention
  (20-30 minutes)
  - https://colab.research.google.com/github/tensorflow/tensorflow/blob/master/tensorflow/contrib/eager/python/examples/nmt_with_attention/nmt_with_attention.ipynb

Text Classification
IMDB Movie Reviews

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

CO tf02_basic-text-classification.ipynb ☆

File Edit View Insert Runtime Tools Help

COMMENT SHARE A

CODE TEXT CELL CELL CONNECT EDITING

Table of contents Code snippets Files

Copyright 2018 The TensorFlow Authors.

↳ *2 cells hidden*

Text classification with movie reviews

View on TensorFlow.org Run in Google Colab View source on GitHub

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.

```
# memory footprint support libraries/code
!ln -sf /opt/bin/nvidia-smi /usr/bin/nvidia-smi
!pip install gputil
!pip install psutil
!pip install humanize
import psutil
import humanize
import os
import GPUtil as GPU
GPUs = GPU.getGPUs()
gpu = GPUs[0]
def printm():
    process = psutil.Process(os.getpid())
```

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

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.
- Bruce Croft, Donald Metzler, and Trevor Strohman (2008), Search Engines: Information Retrieval in Practice, Addison Wesley, http://www.search-engines-book.com/
- Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson.
- Jake VanderPlas (2016), Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
- Steven Bird, Ewan Klein and Edward Loper (2009), Natural Language Processing with Python, O'Reilly Media, http://www.nltk.org/book/ , http://www.nltk.org/book_1ed/
- 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.
- Dipanjan Sarkar (2016), Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data, Apress.
- Bing Liu (2009), Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.

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 人工智慧 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/