# Artificial Intelligence for Text Analytics

國立臺北大學
National Taipei University

# Foundations of Text Analytics: Natural Language Processing (NLP)

1102AITA02
MBA, IM, NTPU (M5026) (Spring 2022)
Tue 2, 3, 4 (9:10-12:00) (B8F40)

https://meet.google.com/paj-zhhj-mya

aws educate | Cloud Ambassador
2020 Cohort

aws academy Accredited Educator

aws certified Cloud Practitioner

aws certified Solutions Architect Associate


## Min-Yuh Day, Ph.D,
## Associate Professor

Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday


2022-03-01

# Syllabus

| Week | Date | Subject/Topics |
|---|---|---|
| 1 | 2022/02/22 | Introduction to Artificial Intelligence for Text Analytics |
| 2 | 2022/03/01 | Foundations of Text Analytics: Natural Language Processing (NLP) |
| 3 | 2022/03/08 | Python for Natural Language Processing |
| 4 | 2022/03/15 | Natural Language Processing with Transformers |
| 5 | 2022/03/22 | Case Study on Artificial Intelligence for Text Analytics I |
| 6 | 2022/03/29 | Text Classification and Sentiment Analysis |

# Syllabus

| Week | Date | Subject/Topics |
|---|---|---|
| 7 | 2022/04/05 | Tomb-Sweeping Day (Holiday, No Classes) |
| 8 | 2022/04/12 | Midterm Project Report |
| 9 | 2022/04/19 | Multilingual Named Entity Recognition (NER),<br>Text Similarity and Clustering |
| 10 | 2022/04/26 | Text Summarization and Topic Models |
| 11 | 2022/05/03 | Text Generation |
| 12 | 2022/05/10 | Case Study on Artificial Intelligence for Text Analytics II |

# Syllabus

| Week | Date | Subject/Topics |
|---|---|---|
| 13 | 2022/05/17 | Question Answering and Dialogue Systems |
| 14 | 2022/05/24 | Deep Learning, Transfer Learning,<br>Zero-Shot, and Few-Shot Learning for Text Analytics |
| 15 | 2022/05/31 | Final Project Report I |
| 16 | 2022/06/07 | Final Project Report II |
| 17 | 2022/06/14 | Self-learning |
| 18 | 2022/06/21 | Self-learning |

# Foundations of Text Analytics: Natural Language Processing (NLP)

# Outline

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

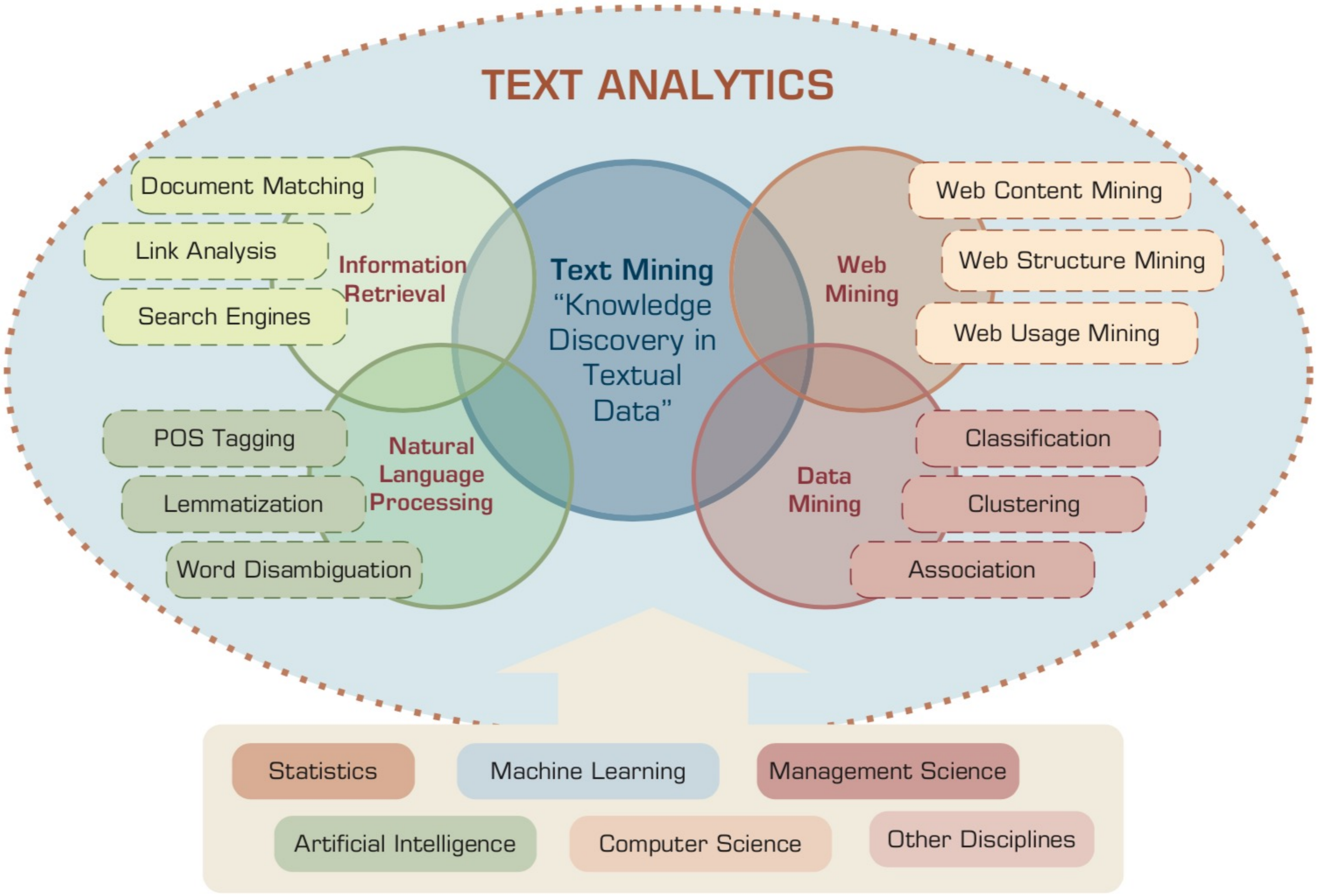
# Artificial Intelligence
# (AI)

# Text Analytics (TA)

# Text Mining (TM)

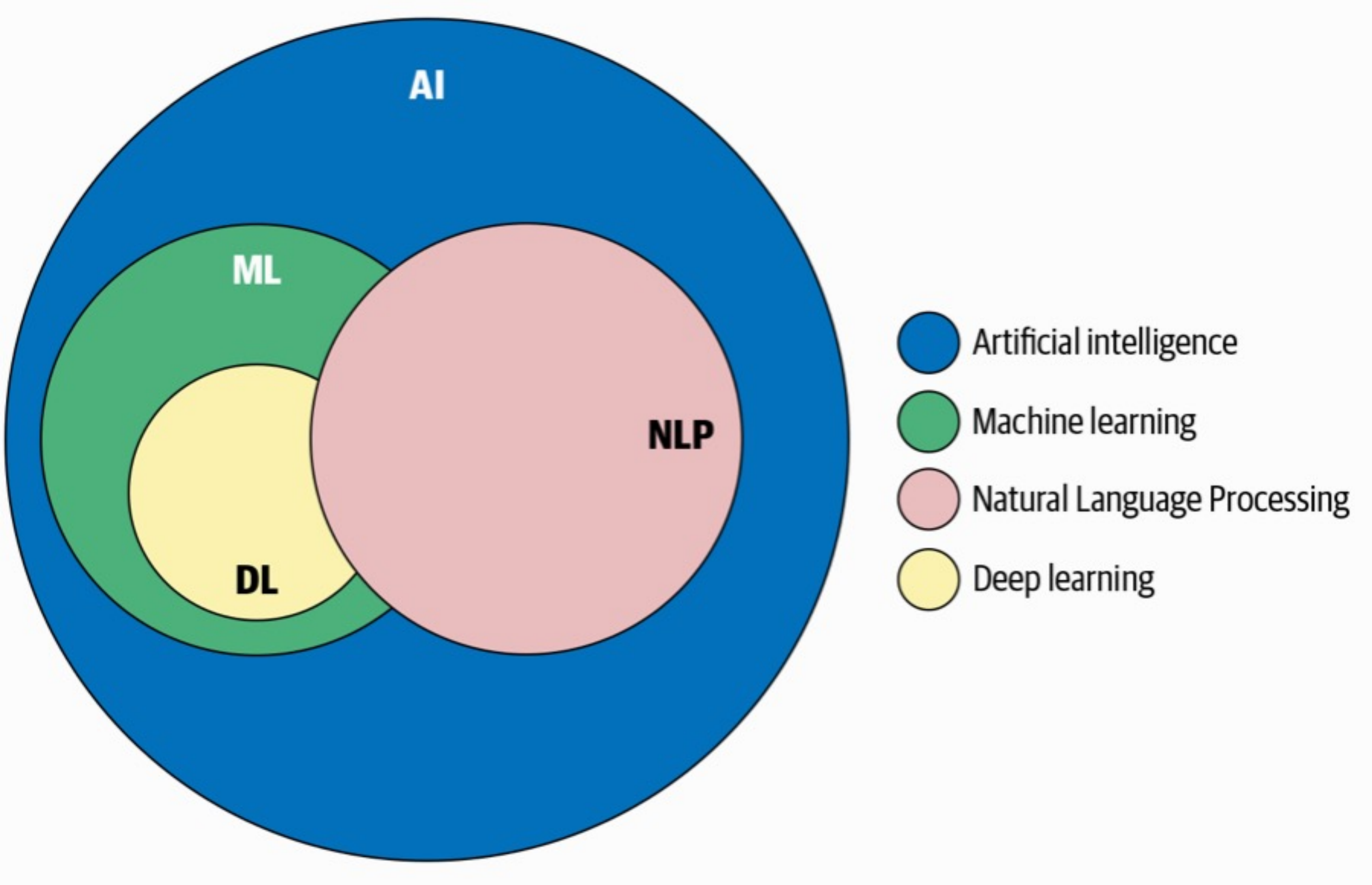
# Natural Language Processing (NLP)

# 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

# AI, NLP, ML, DL

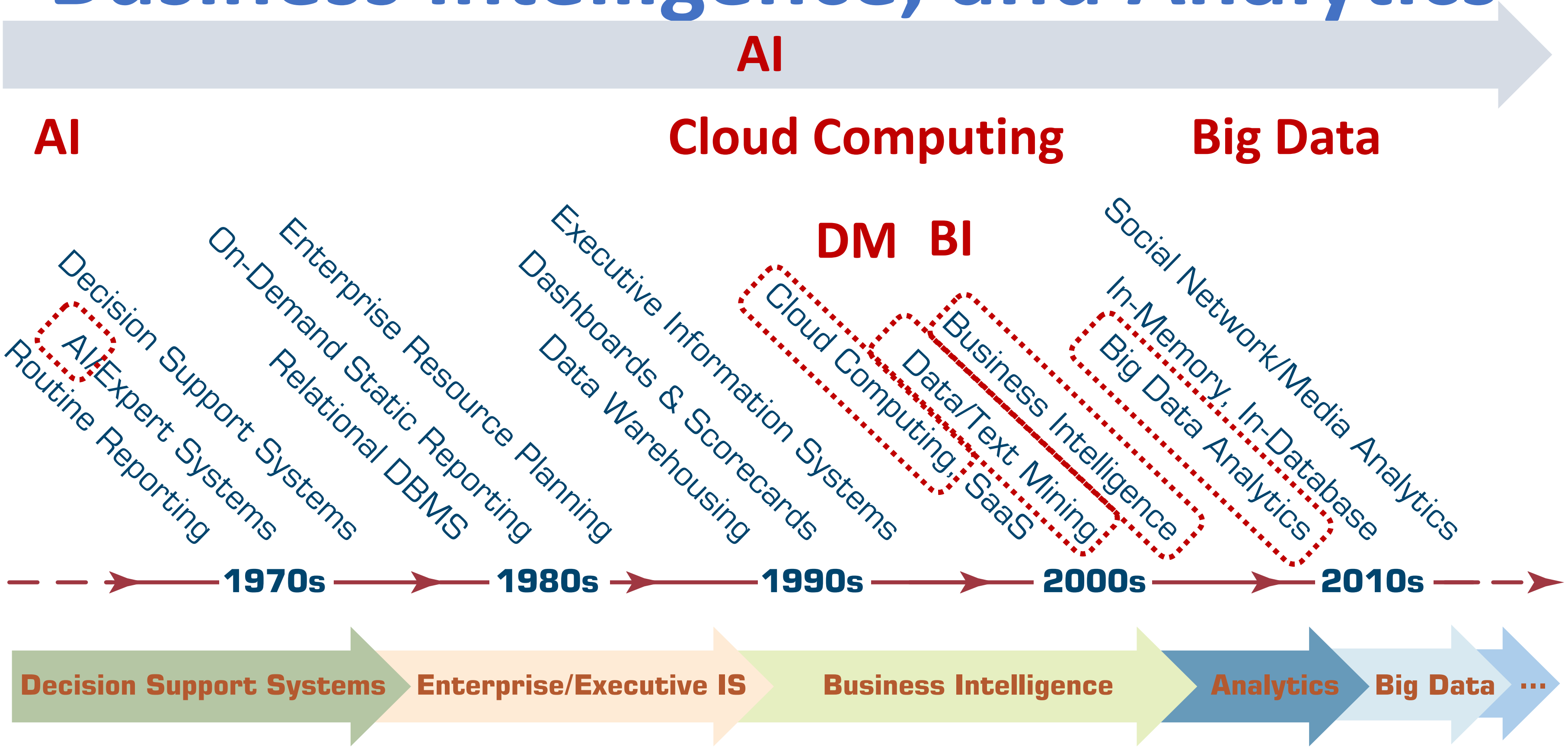



Source: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.

# Artificial Intelligence (AI)

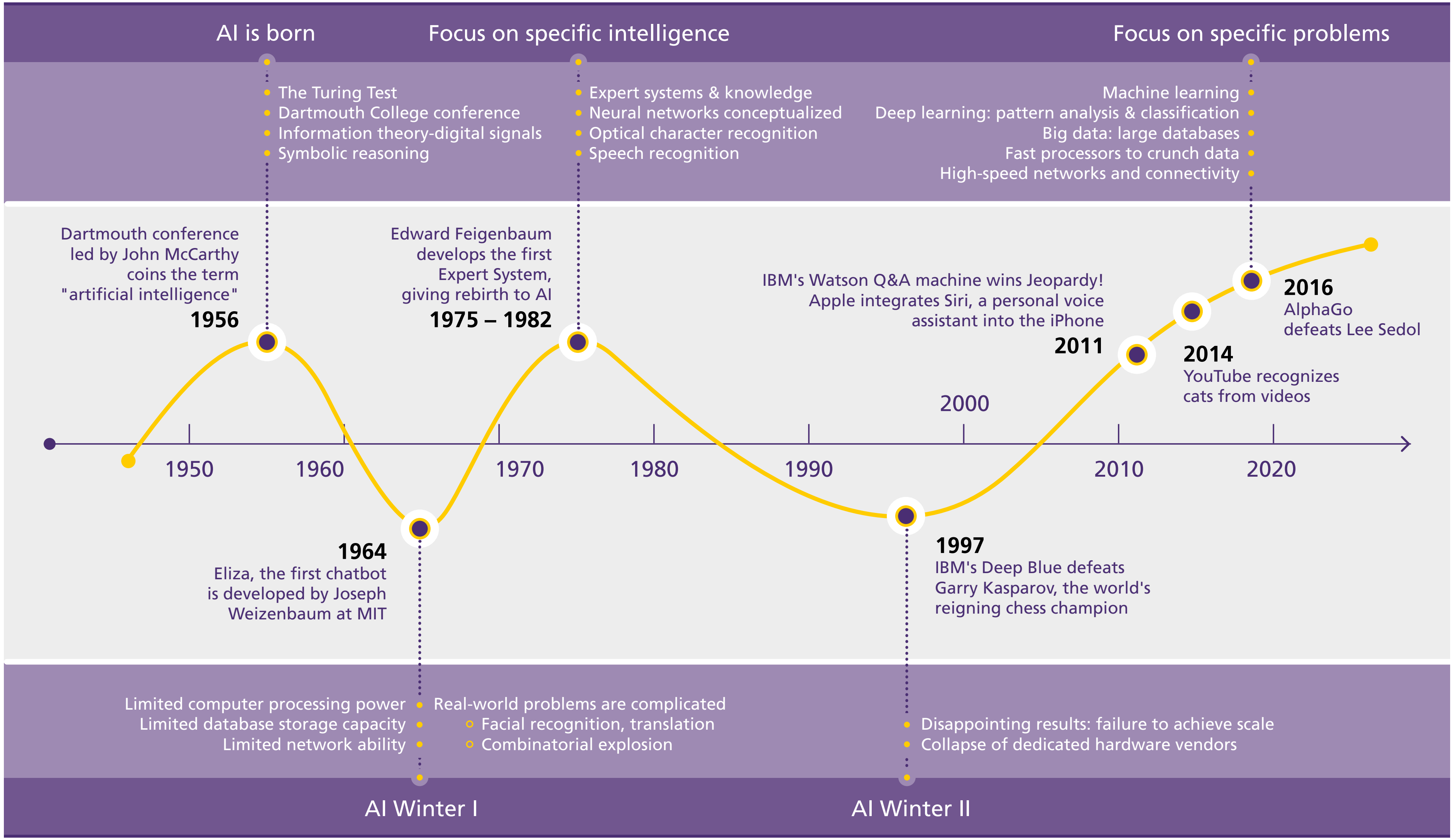
# AI, Big Data, Cloud Computing

# Evolution of Decision Support, Business Intelligence, and Analytics




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

# The Rise of AI




Source: DHL (2018), Artificial Intelligence in Logistics, http://www.globalhha.com/doclib/data/upload/doc_con/5e50c53c5bf67.pdf/

# Definition
# of
# Artificial Intelligence
# (A.I.)

# Artificial Intelligence

"… the **science** and

**engineering**

of

making

**intelligent machines**"

(John McCarthy, 1955)

Source: https://digitalintelligencetoday.com/artificial-intelligence-defined-useful-list-of-popular-definitions-from-business-and-science/

# Artificial Intelligence

**"... technology that thinks and acts like humans"**

Source: https://digitalintelligencetoday.com/artificial-intelligence-defined-useful-list-of-popular-definitions-from-business-and-science/

# Artificial Intelligence

“… intelligence exhibited by machines or software”

Source: https://digitalintelligencetoday.com/artificial-intelligence-defined-useful-list-of-popular-definitions-from-business-and-science/

# 4 Approaches of AI

| | |
|---|---|
| Thinking Humanly | Thinking Rationally |
| Acting Humanly | Acting Rationally |

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

# 4 Approaches of AI

| | |
|---|---|
| **2.**<br>**Thinking Humanly:**<br>**The Cognitive Modeling Approach** | **3.**<br>**Thinking Rationally:**<br>**The "Laws of Thought" Approach** |
| **1.**<br>**Acting Humanly:**<br>**The Turing Test Approach** (1950) | **4.**<br>**Acting Rationally:**<br>**The Rational Agent Approach** |

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

# AI Acting Humanly: The Turing Test Approach

## (Alan Turing, 1950)

- Knowledge Representation
- Automated Reasoning
- Machine Learning (ML)
  - Deep Learning (DL)
- Computer Vision (Image, Video)
- Natural Language Processing (NLP)
- Robotics

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

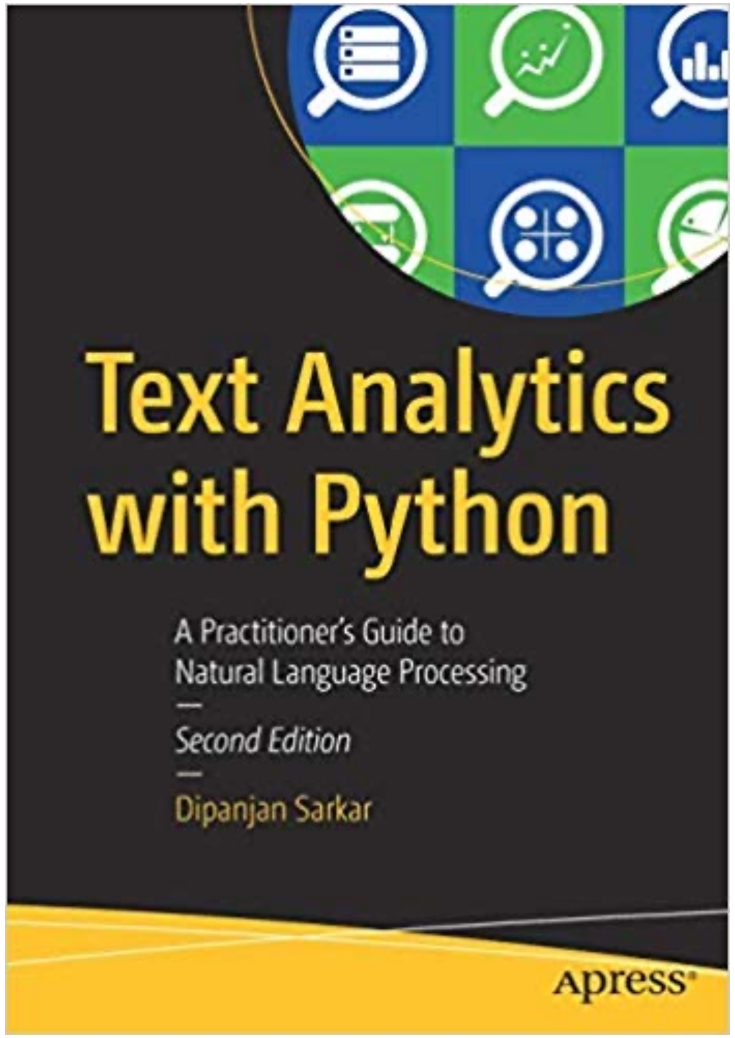
# Text Analytics
# and
# Text Mining

Dipanjan Sarkar (2019),

# Text Analytics with Python:

## A Practitioner's Guide to Natural Language Processing,

Second Edition. APress.



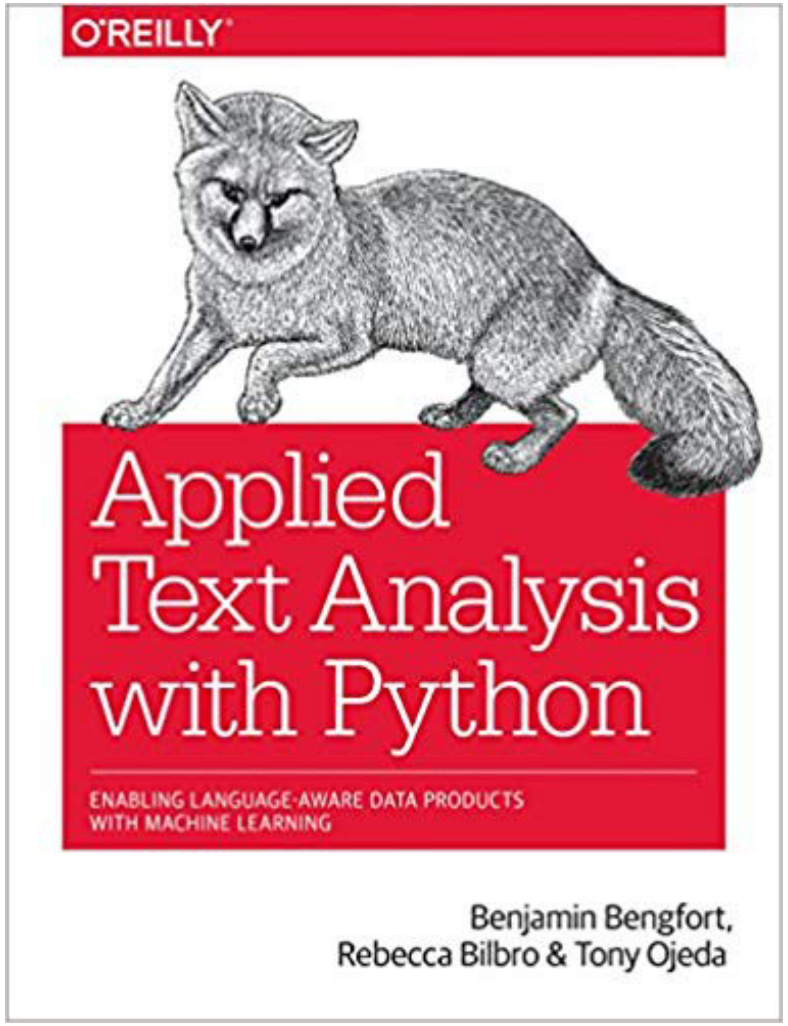

Source: https://www.amazon.com/Text-Analytics-Python-Practitioners-Processing/dp/1484243536

Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018),

# Applied Text Analysis with Python:

## Enabling Language-Aware Data Products with Machine Learning,

O'Reilly.

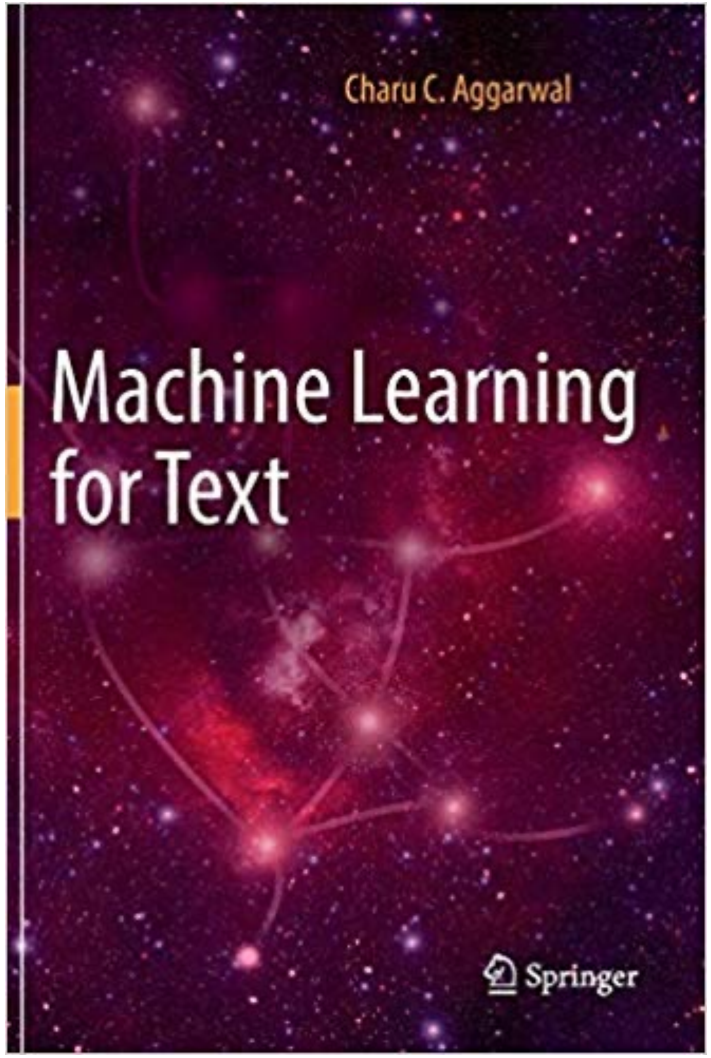



Source: https://www.amazon.com/Applied-Text-Analysis-Python-Language-Aware/dp/1491963042

## Charu C. Aggarwal (2018),

# Machine Learning for Text,

## Springer



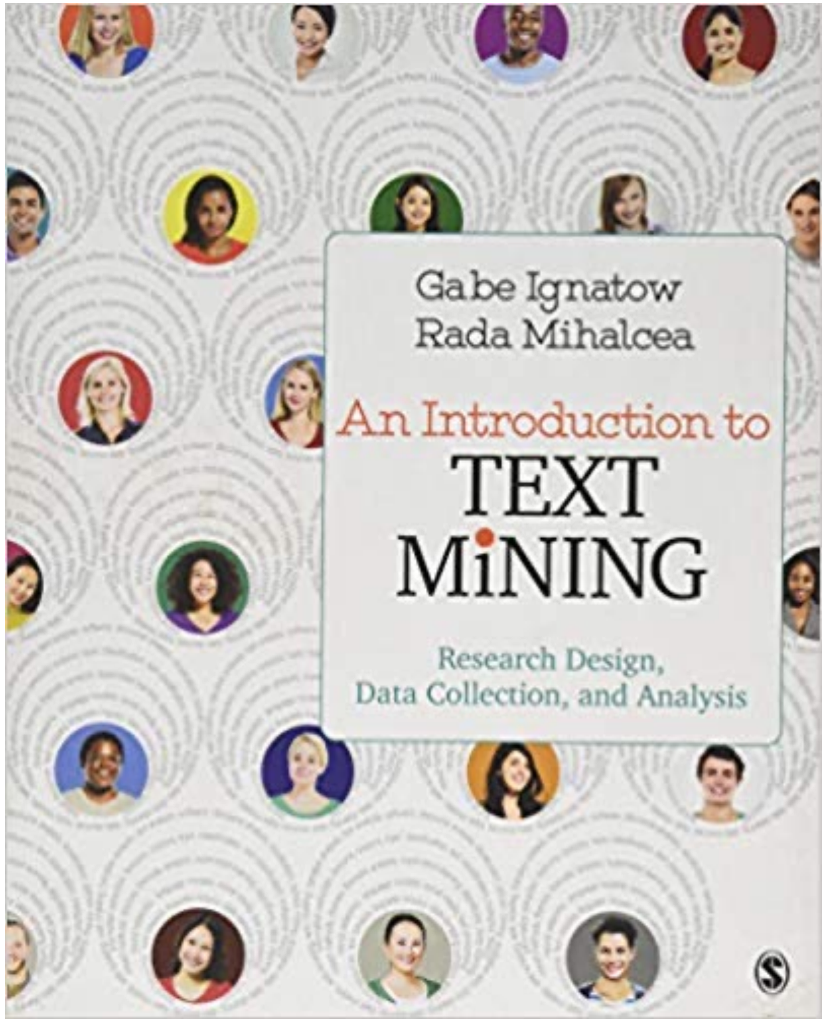

Source: https://www.amazon.com/Machine-Learning-Text-Charu-Aggarwal/dp/3319735306

**Gabe Ignatow and Rada F. Mihalcea (2017),**

# An Introduction to Text Mining:

## Research Design, Data Collection, and Analysis,

**SAGE Publications.**



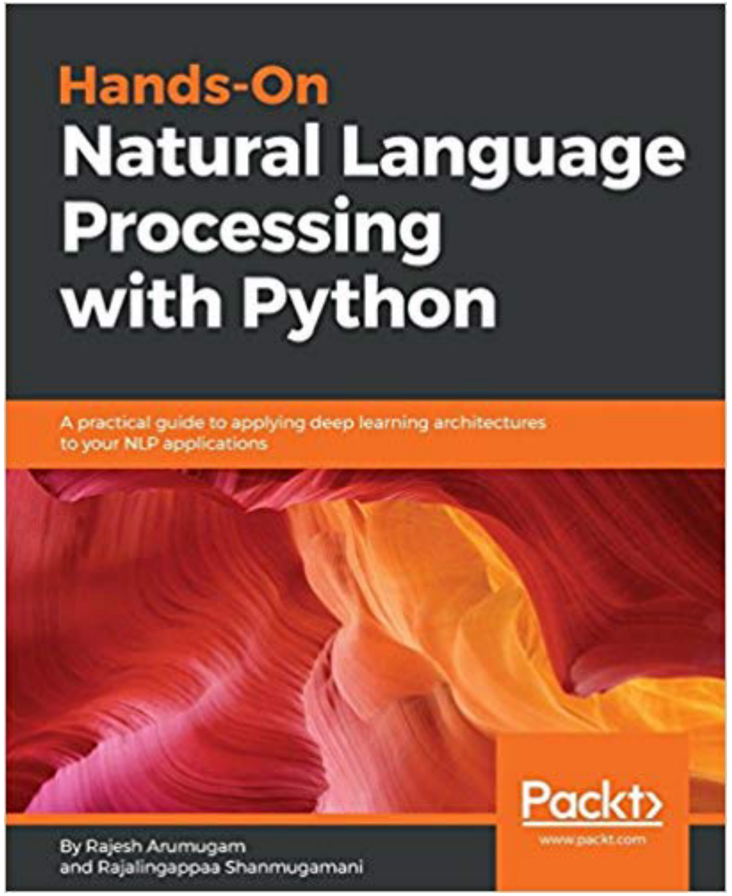

Source: https://www.amazon.com/Introduction-Text-Mining-Research-Collection/dp/1506337007

Rajesh Arumugam (2018),

# Hands-On Natural Language Processing with Python:

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

Packt



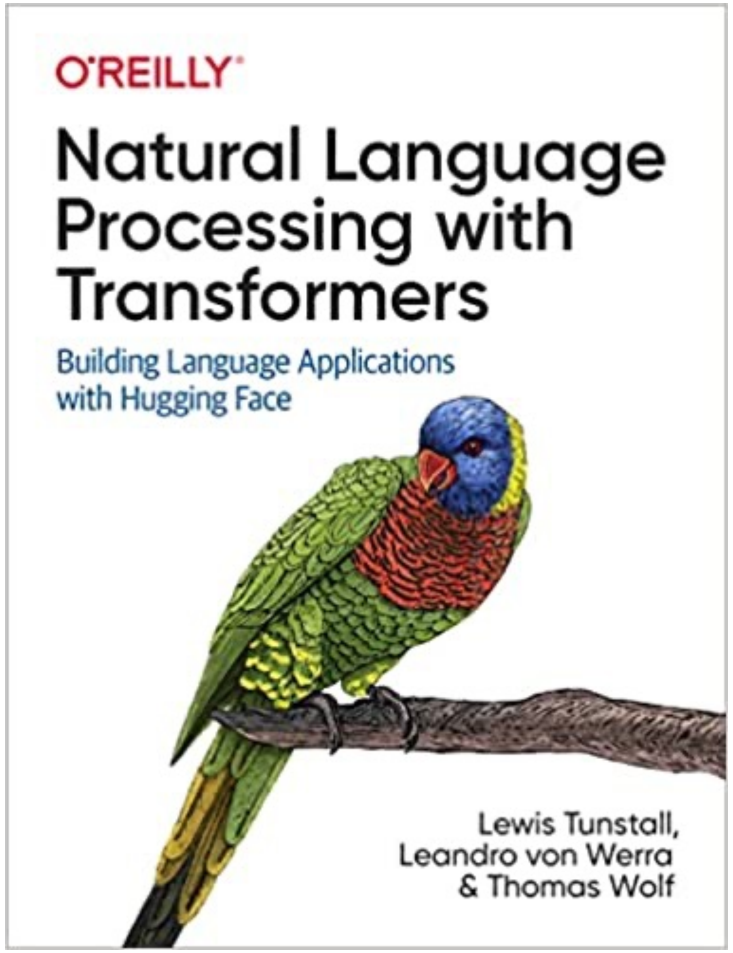

https://www.amazon.com/Hands-Natural-Language-Processing-Python/dp/178913949X

Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022),

# Natural Language Processing with Transformers:

## Building Language Applications with Hugging Face,

O'Reilly Media.



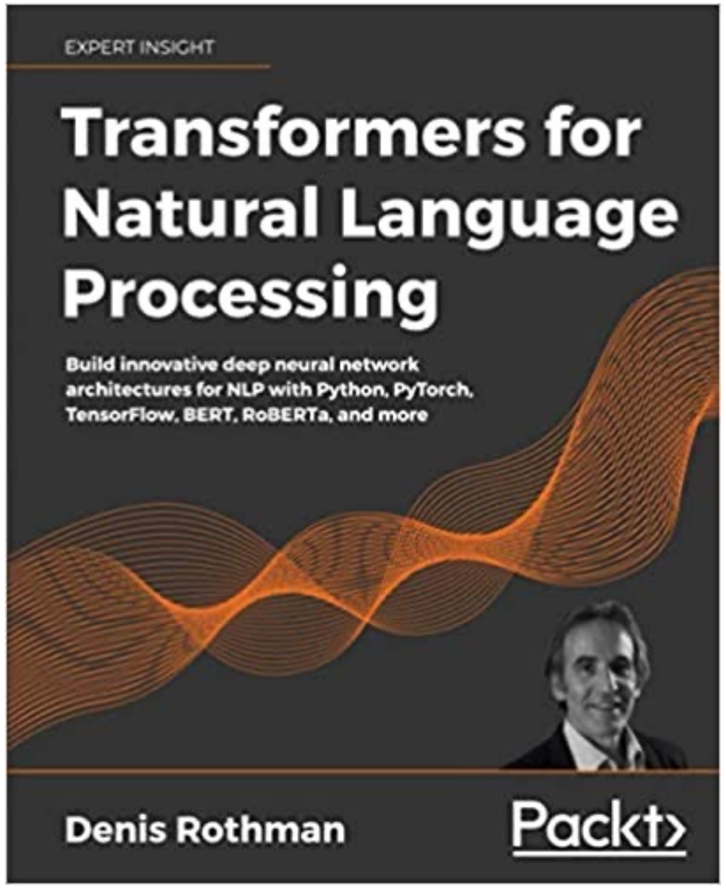

Source: https://www.amazon.com/Natural-Language-Processing-Transformers-Applications/dp/1098103246

# Denis Rothman (2021),

# Transformers for Natural Language Processing:

Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more,

Packt Publishing.



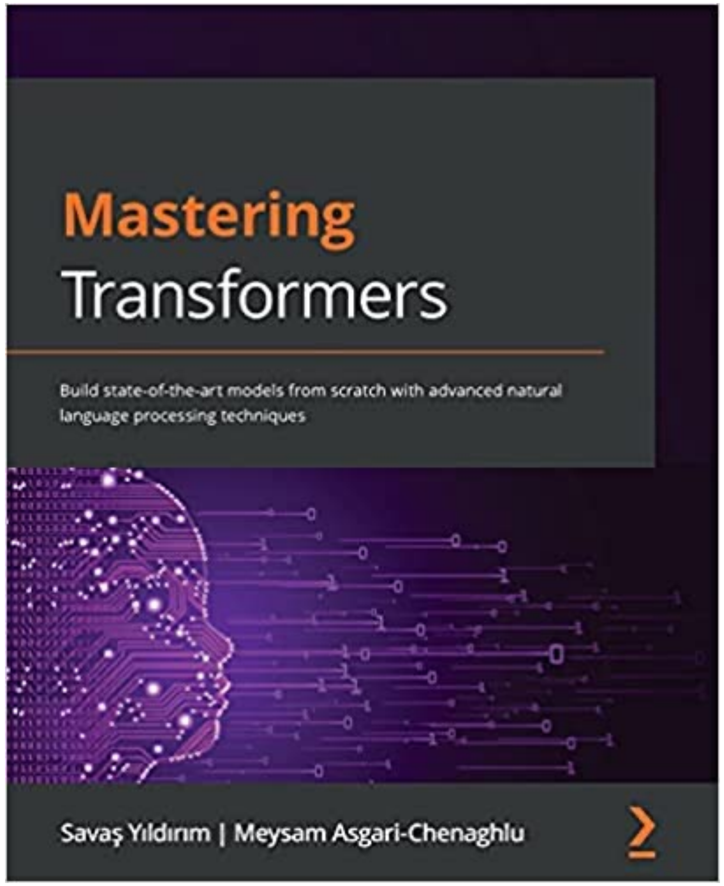

Source: https://www.amazon.com/Transformers-Natural-Language-Processing-architectures/dp/1800565798

**Savaş Yıldırım and Meysam Asgari-Chenaghlu (2021),**

# Mastering Transformers:

Build state-of-the-art models from scratch with advanced natural language processing techniques,

Packt Publishing.



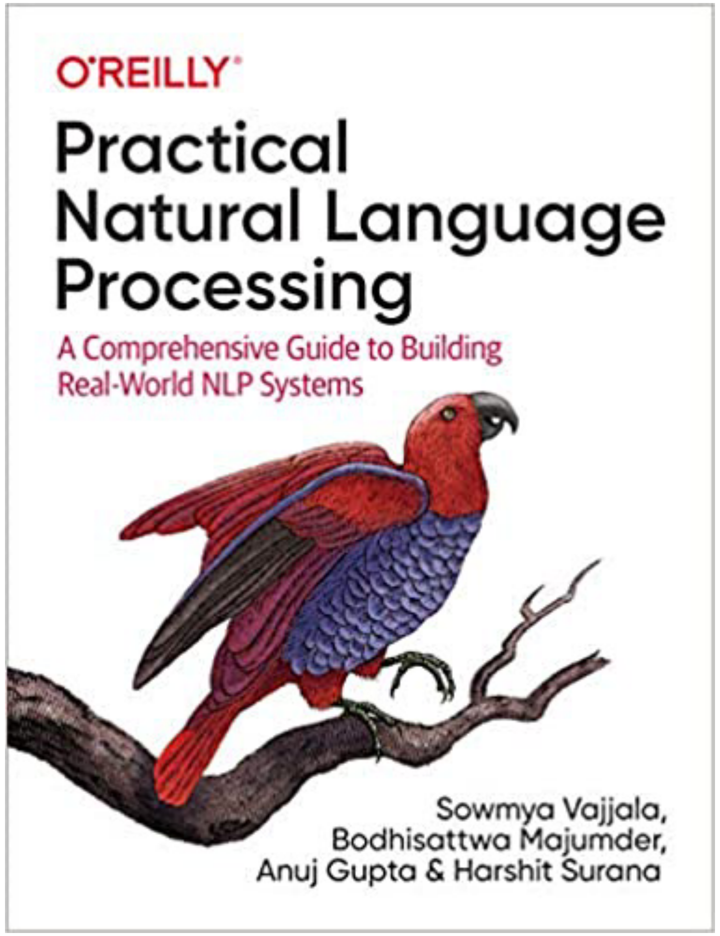

Source: https://www.amazon.com/Mastering-Transformers-state-art-processing/dp/1801077657/

# Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020),

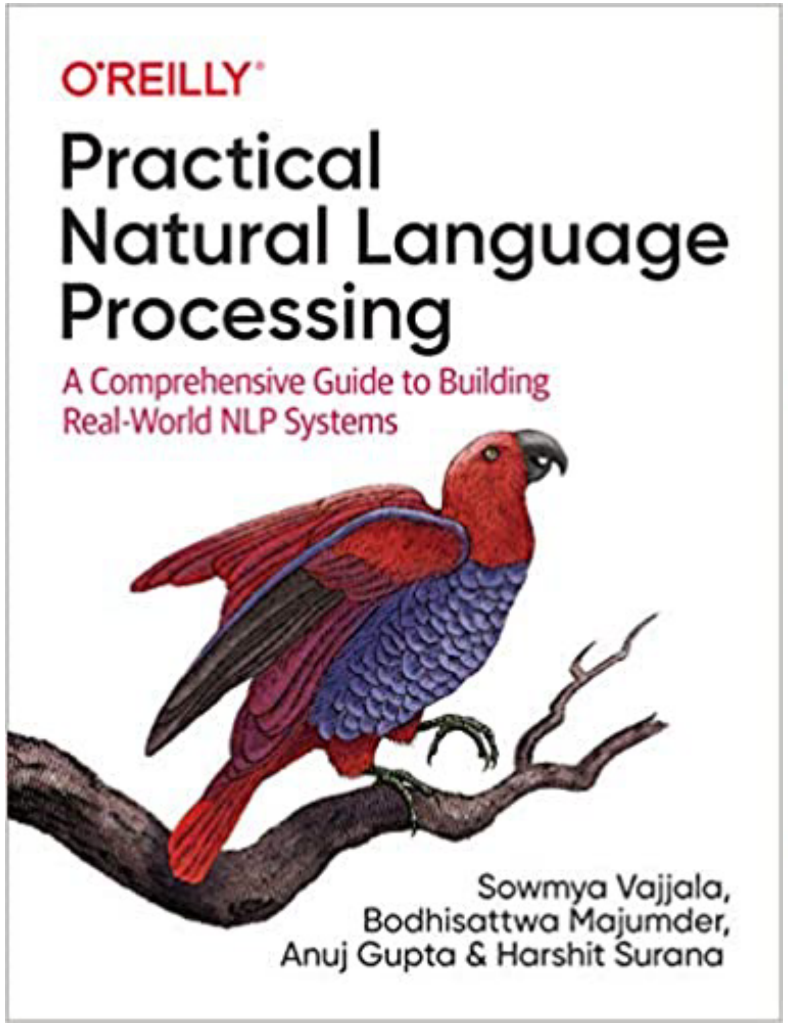
# Practical Natural Language Processing:

## A Comprehensive Guide to Building Real-World NLP Systems,

O'Reilly Media.




Source: https://www.amazon.com/Practical-Natural-Language-Processing-Pragmatic/dp/1492054054




| | | | | | |
|---|---|---|---|---|---|
| **FOUNDATIONS** *Covered in Chapters 1 to 3* | ML for NLP | NLP Pipelines | Data Gathering | Multilingual NLP | Text Representation |
| **CORE TASKS** *Covered in Chapters 3 to 7* | Text Classification | Information Extraction | Conversational Agents | Information Retrieval | Question Answering |
| **GENERAL APPLICATIONS** *Covered in Chapters 4 to 7* | Spam Classification | Calendar Event Extracton | Personal Assistants | Search Engines | JEOPARDY! Jeopardy! |
| **INDUSTRY SPECIFIC** *Covered in Chapters 8 to 10* | Social Media Analysis | Retail Data Extraction | Health Records Analysis | Financial Analysis | Legal Entity Extraction |
| **AI PROJECT PLAYBOOK** *Covered in Chapters 2 & 11* | Project Processes | Best Practices | Model Iterations | MLOps | AI Teams & Hiring |

Source: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.

Source: https://www.amazon.com/Practical-Natural-Language-Processing-Pragmatic/dp/1492054054

# Text Analytics (TA)

# 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

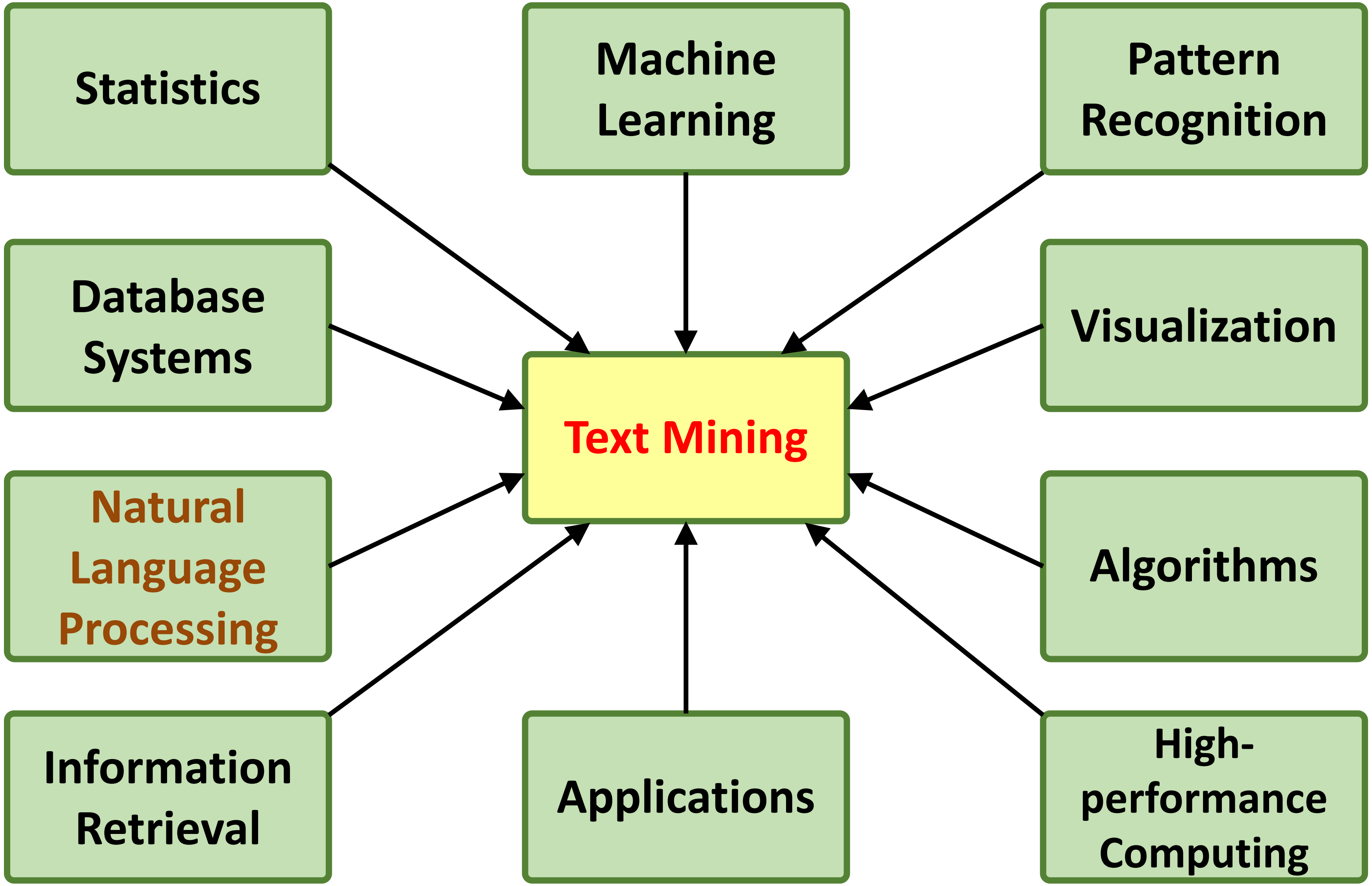
# Text Mining

- **Text Data Mining**
- **Knowledge Discovery in Textual Databases**



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

# Text Mining Technologies




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

# 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

# Emotions

| | |
|---|---|
| Love | Anger |
| Joy | Sadness |
| Surprise | 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.

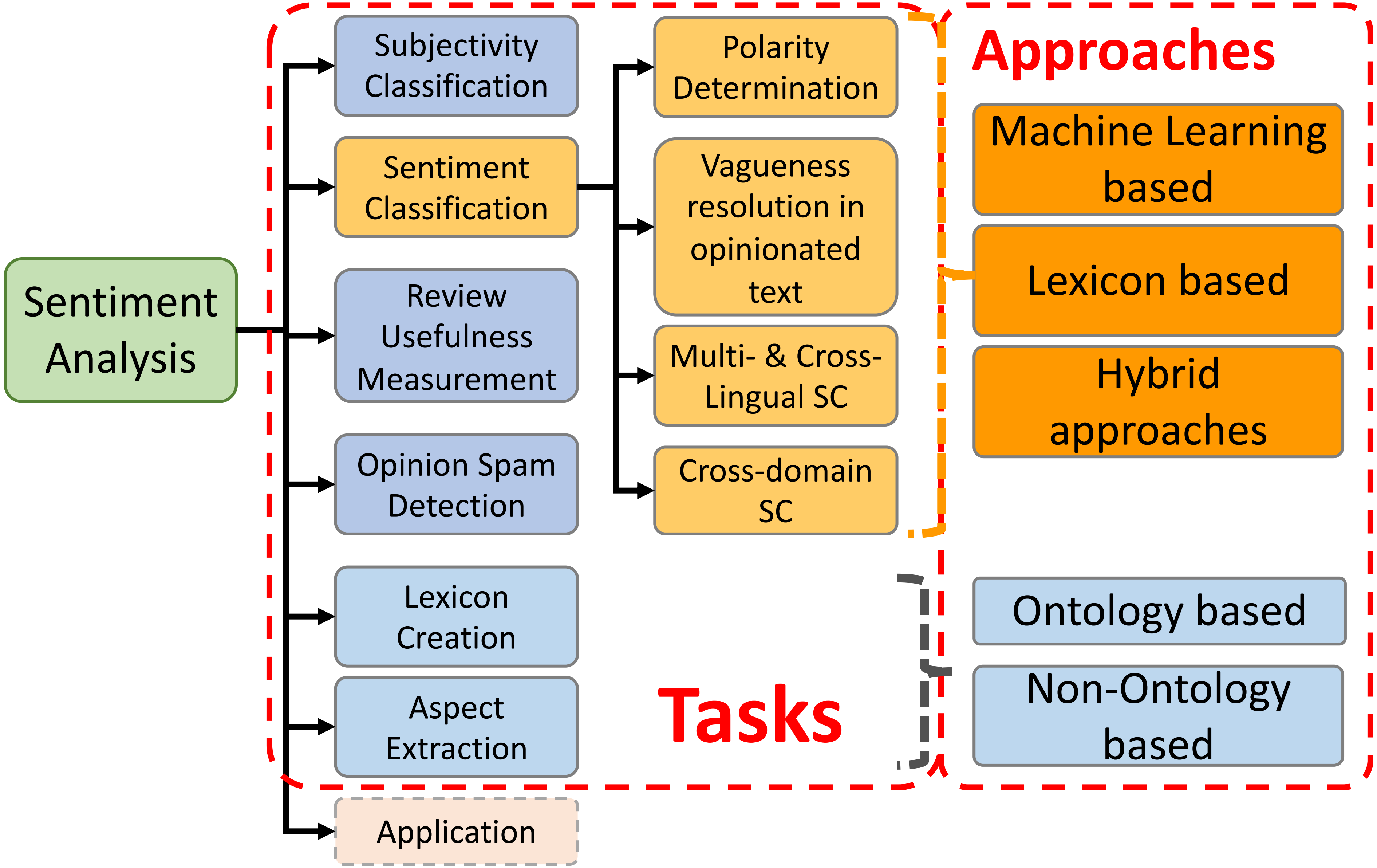
(6) She also thought the phone was too expensive, and wanted me to return it to the shop. … "

+Positive Opinion

-Negative Opinion

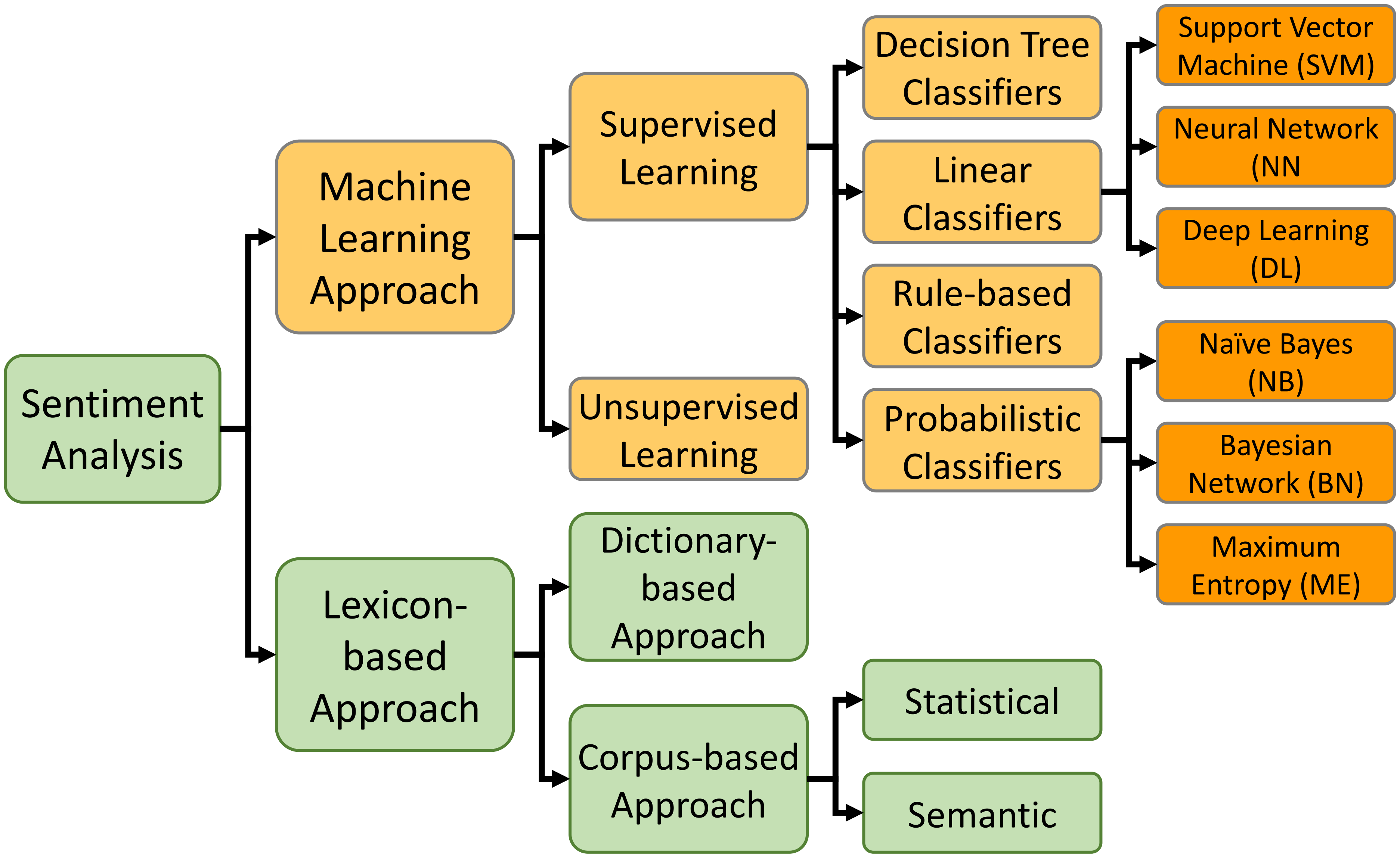
Source: Bing Liu (2011) , "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data," Springer, 2nd Edition,

# Sentiment Analysis



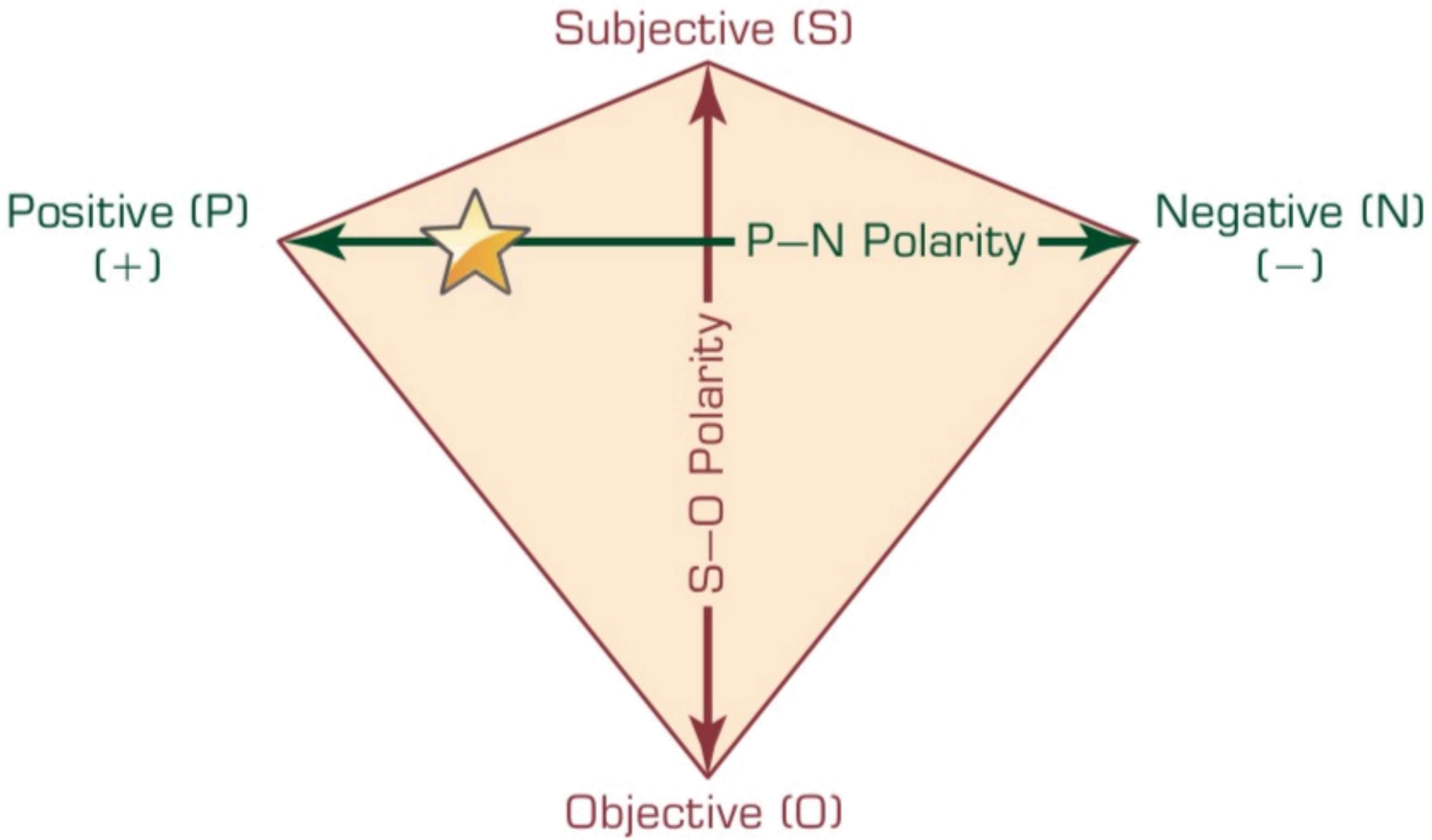

Source: Kumar Ravi and Vadlamani Ravi (2015), "A survey on opinion mining and sentiment analysis: tasks, approaches and applications." Knowledge-Based Systems, 89, pp.14-46.

# Sentiment Classification Techniques



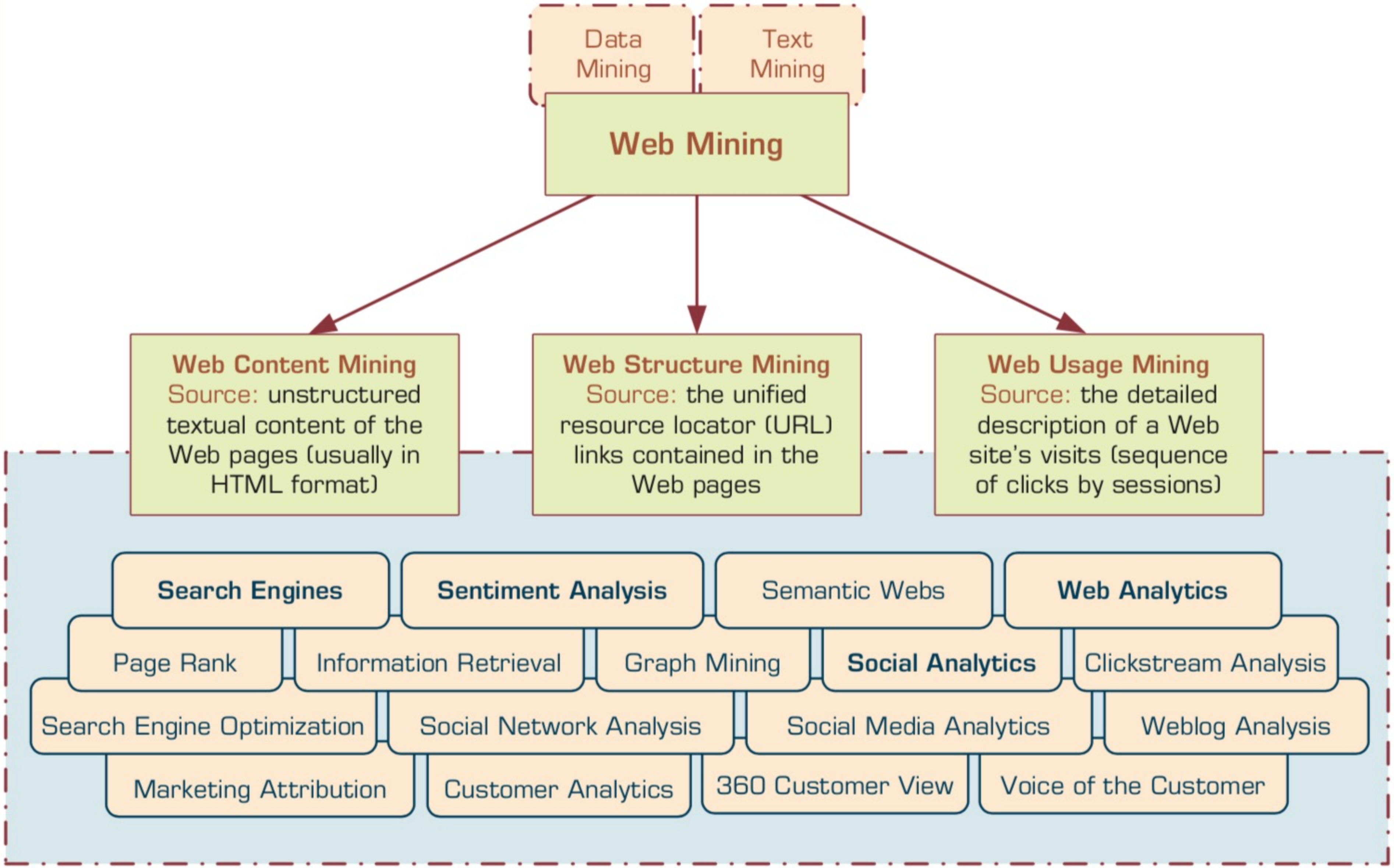

Source: Jesus Serrano-Guerrero, Jose A. Olivas, Francisco P. Romero, and Enrique Herrera-Viedma (2015), "Sentiment analysis: A review and comparative analysis of web services," Information Sciences, 311, pp. 18-38.

# P–N Polarity and S–O Polarity Relationship



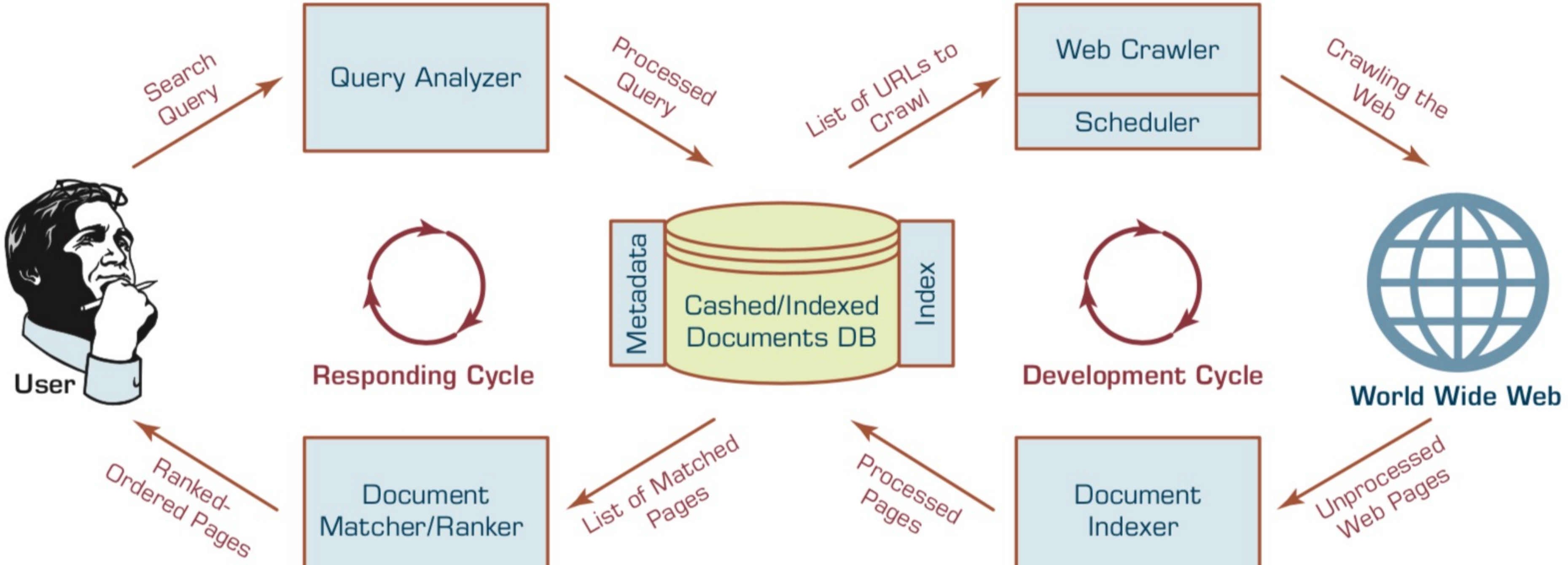

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

# 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




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

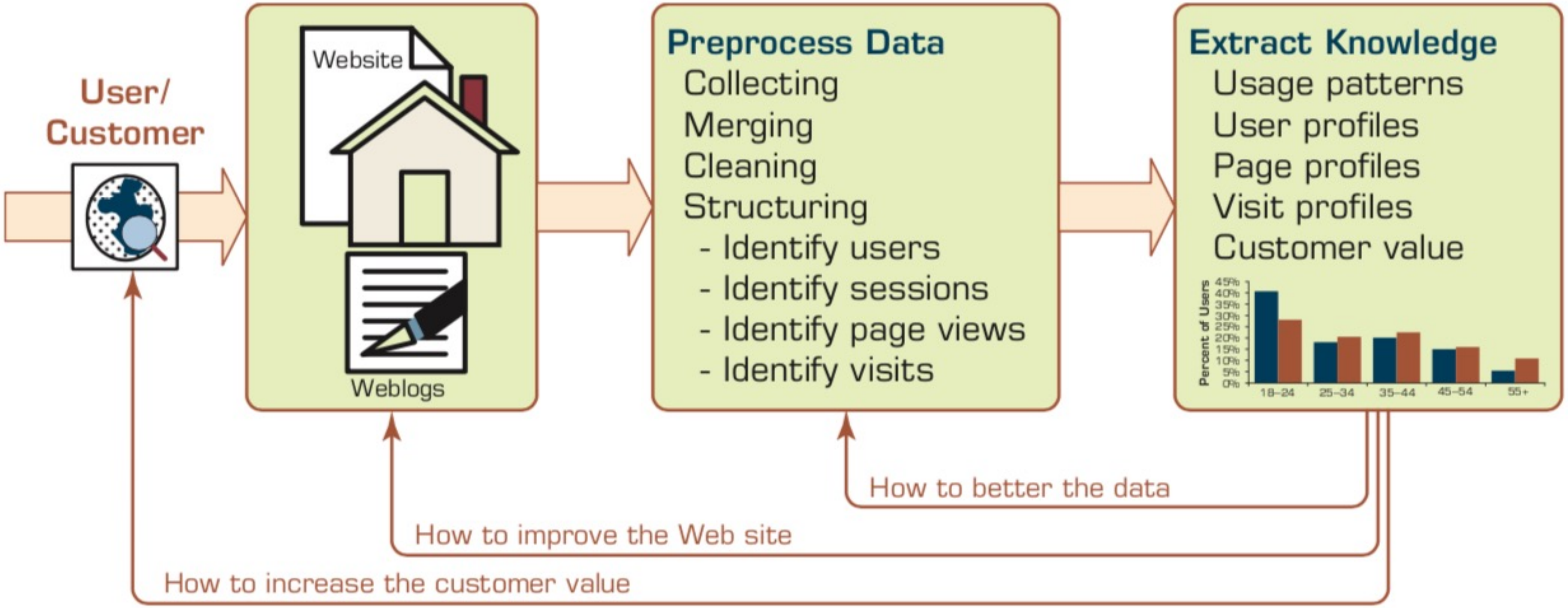
# Web Usage Mining (Web Analytics)

- 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

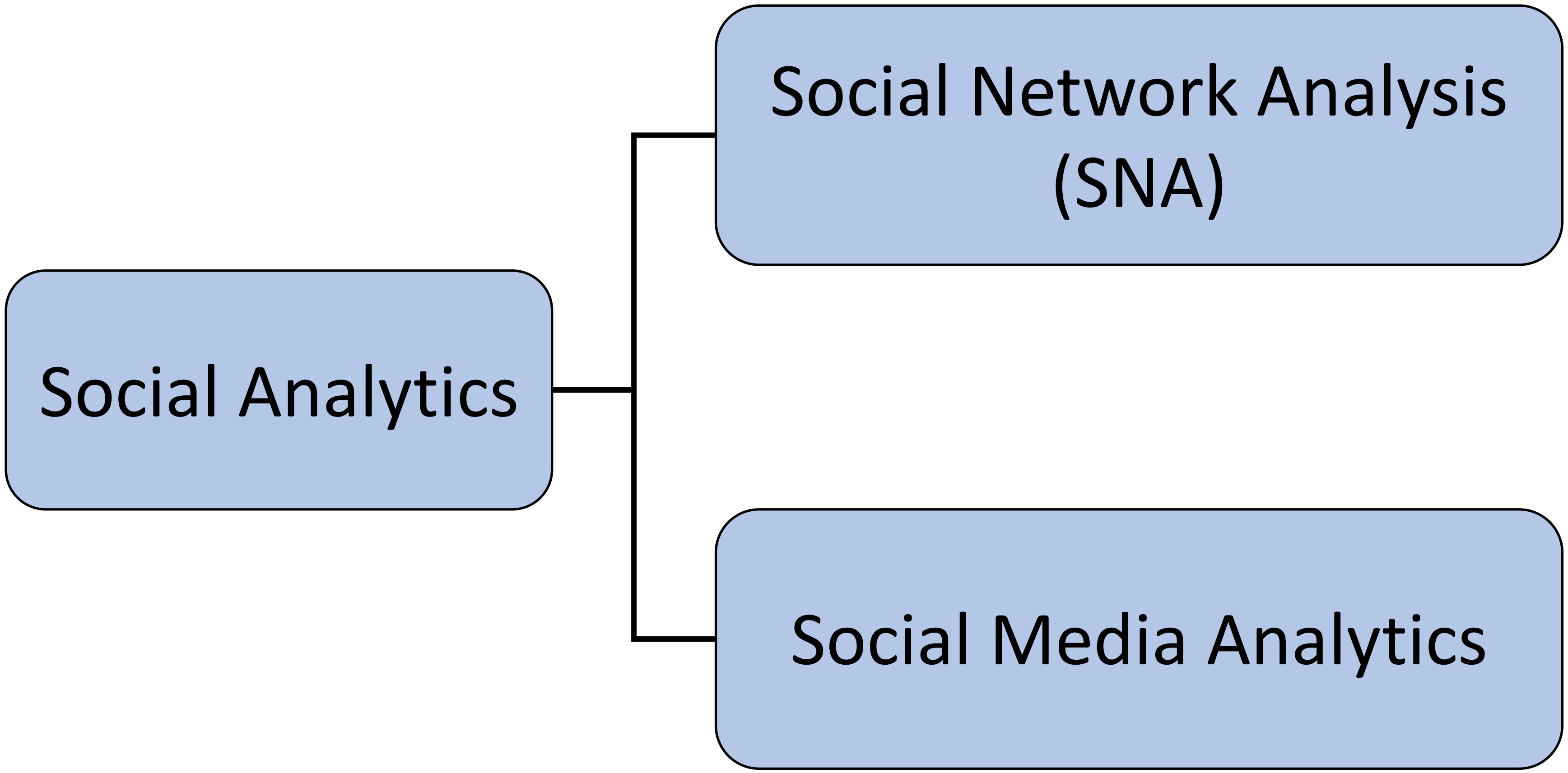
# Social Analytics

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



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

# Branches of Social Analytics




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)

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.**

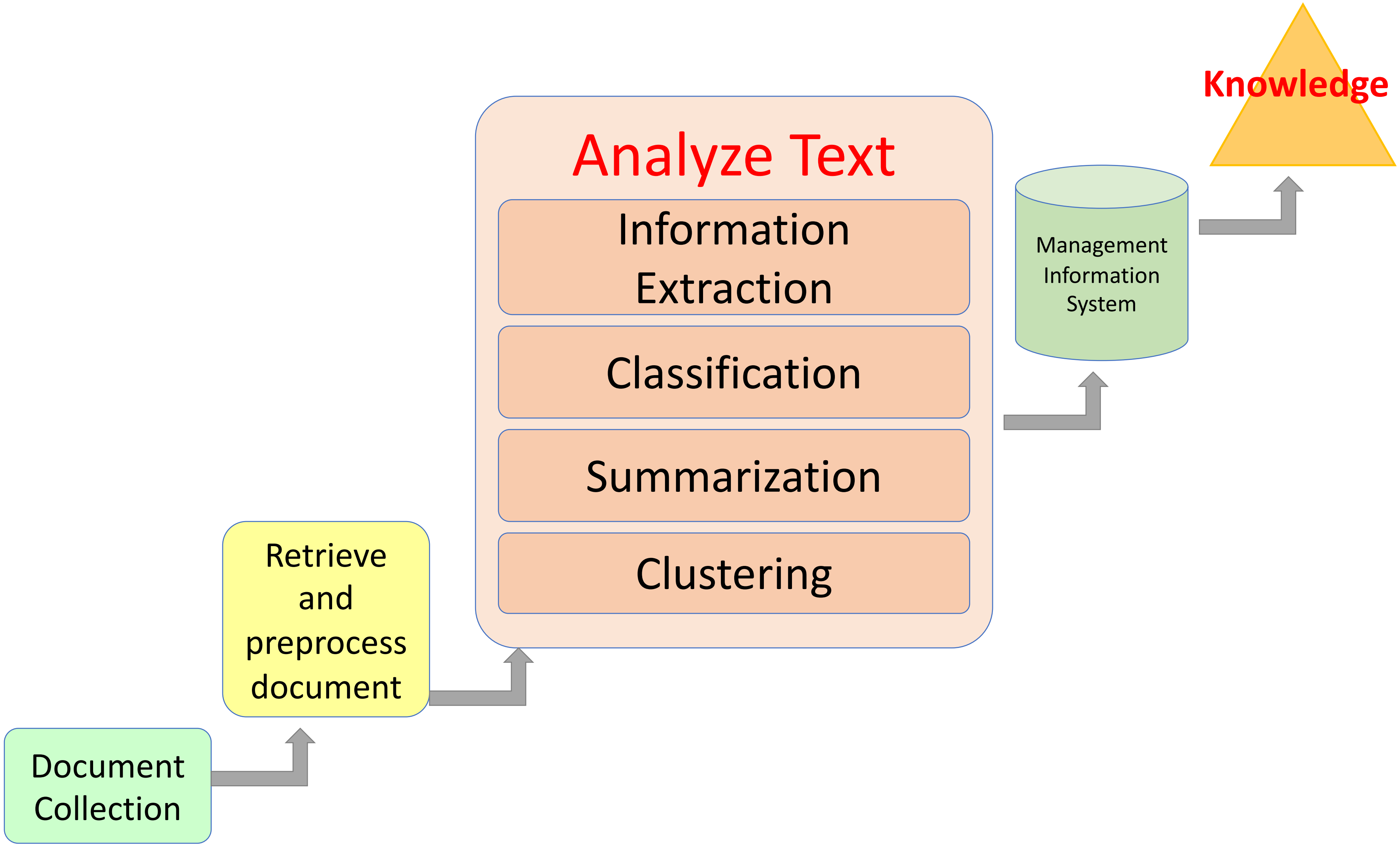
Source: Vishal Gupta and Gurpreet S. Lehal (2009), "A survey of text mining techniques and applications," Journal of emerging technologies in web intelligence, vol. 1, no. 1, pp. 60-76.

# Text Mining:

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

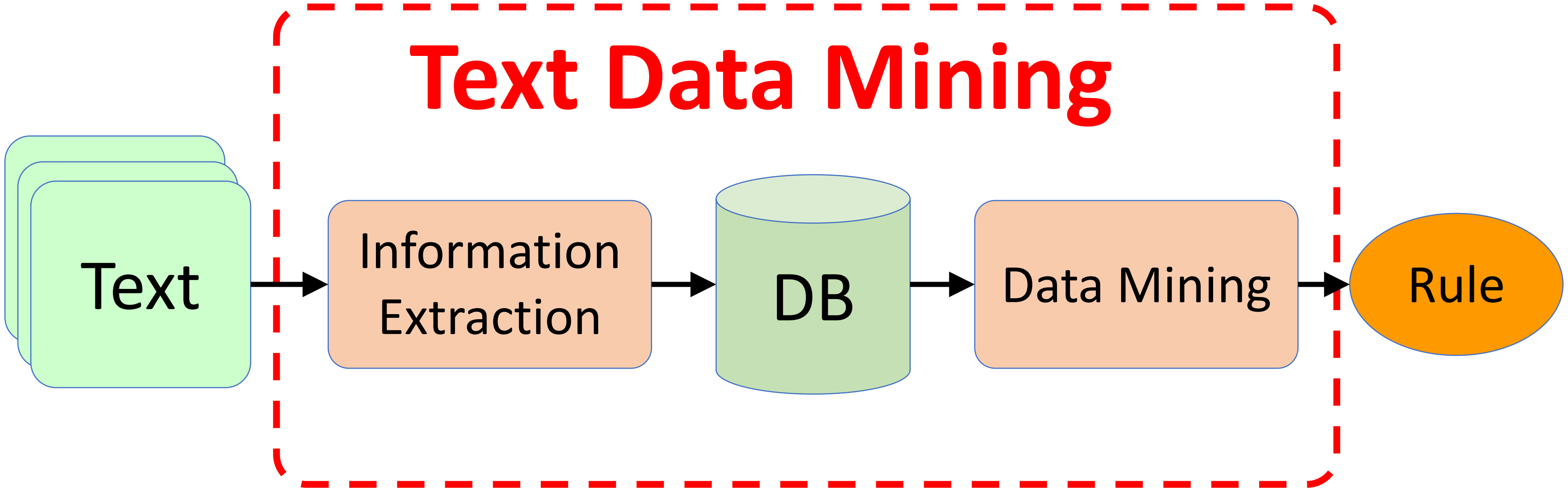
Source: Vishal Gupta and Gurpreet S. Lehal (2009), "A survey of text mining techniques and applications," Journal of emerging technologies in web intelligence, vol. 1, no. 1, pp. 60-76.

# 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.

# Natural Language Processing (NLP)

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



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

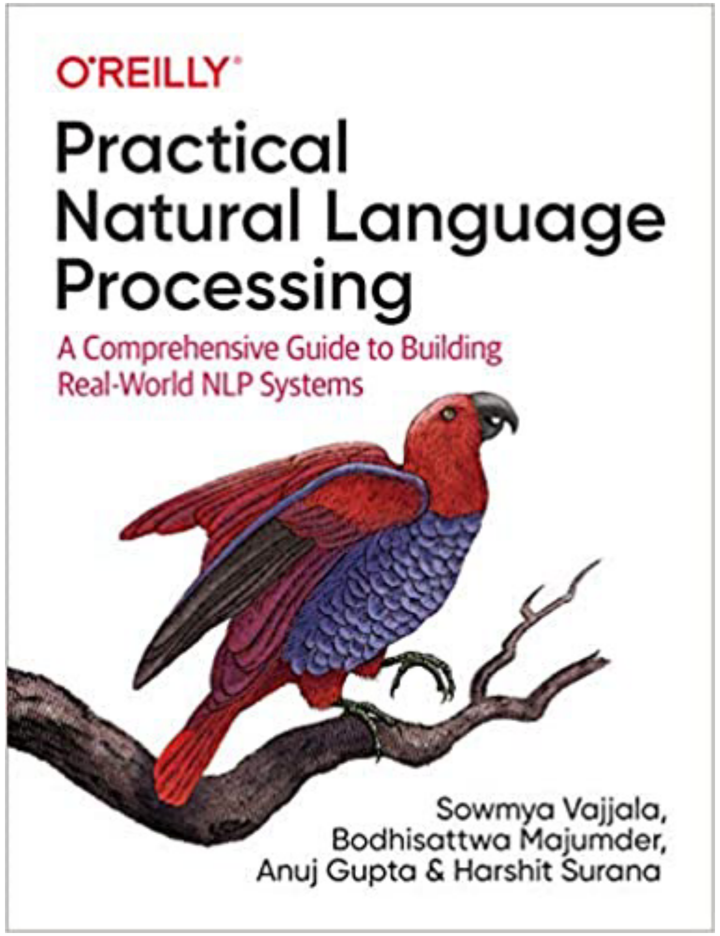
# Natural Language Processing (NLP) and Text Mining

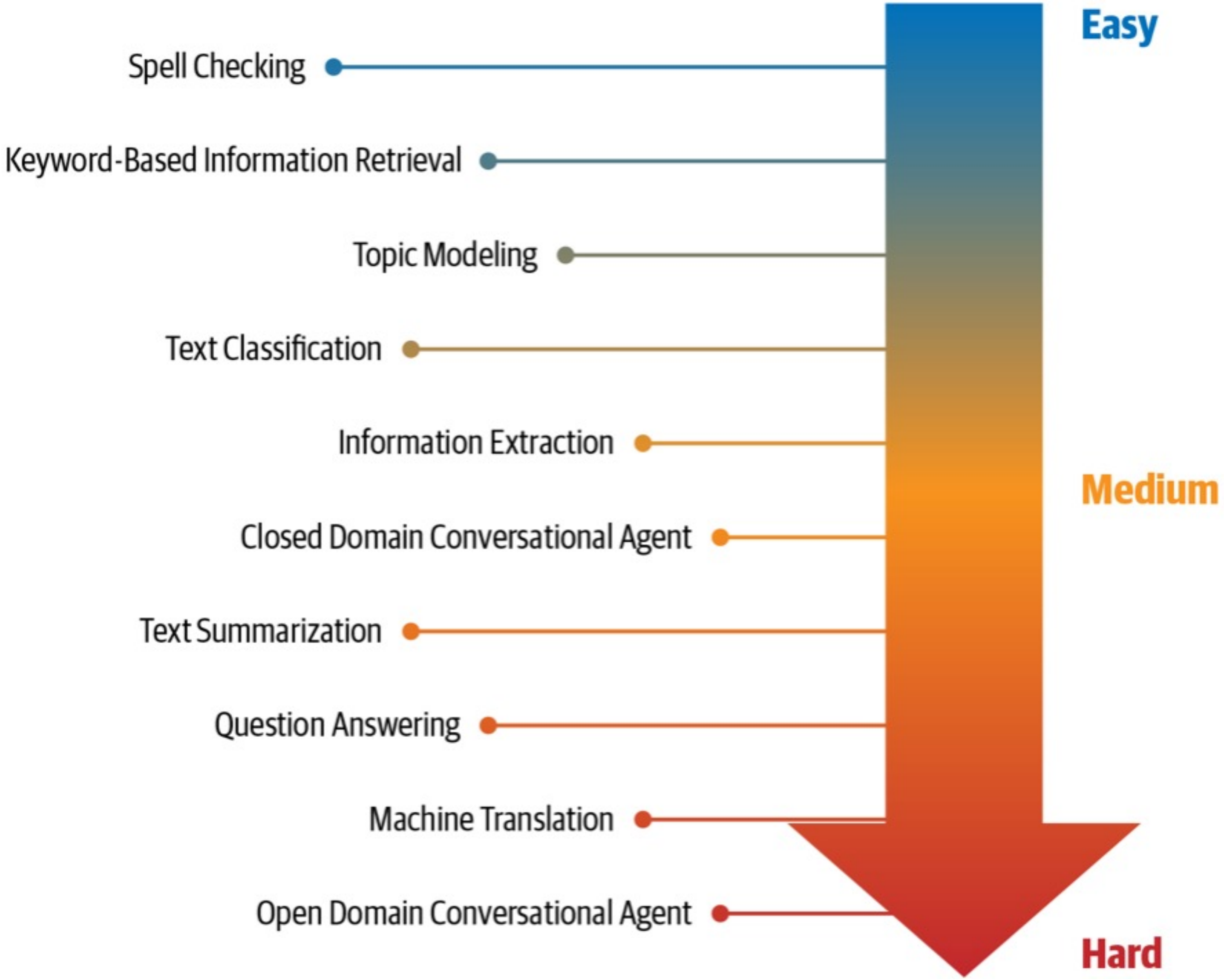
- Raw text
- Sentence Segmentation
- Tokenization
- Part-of-Speech (POS)
- Stop word removal
- Stemming / Lemmatization
- Dependency Parser
- String Metrics & Matching

| word's stem | 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

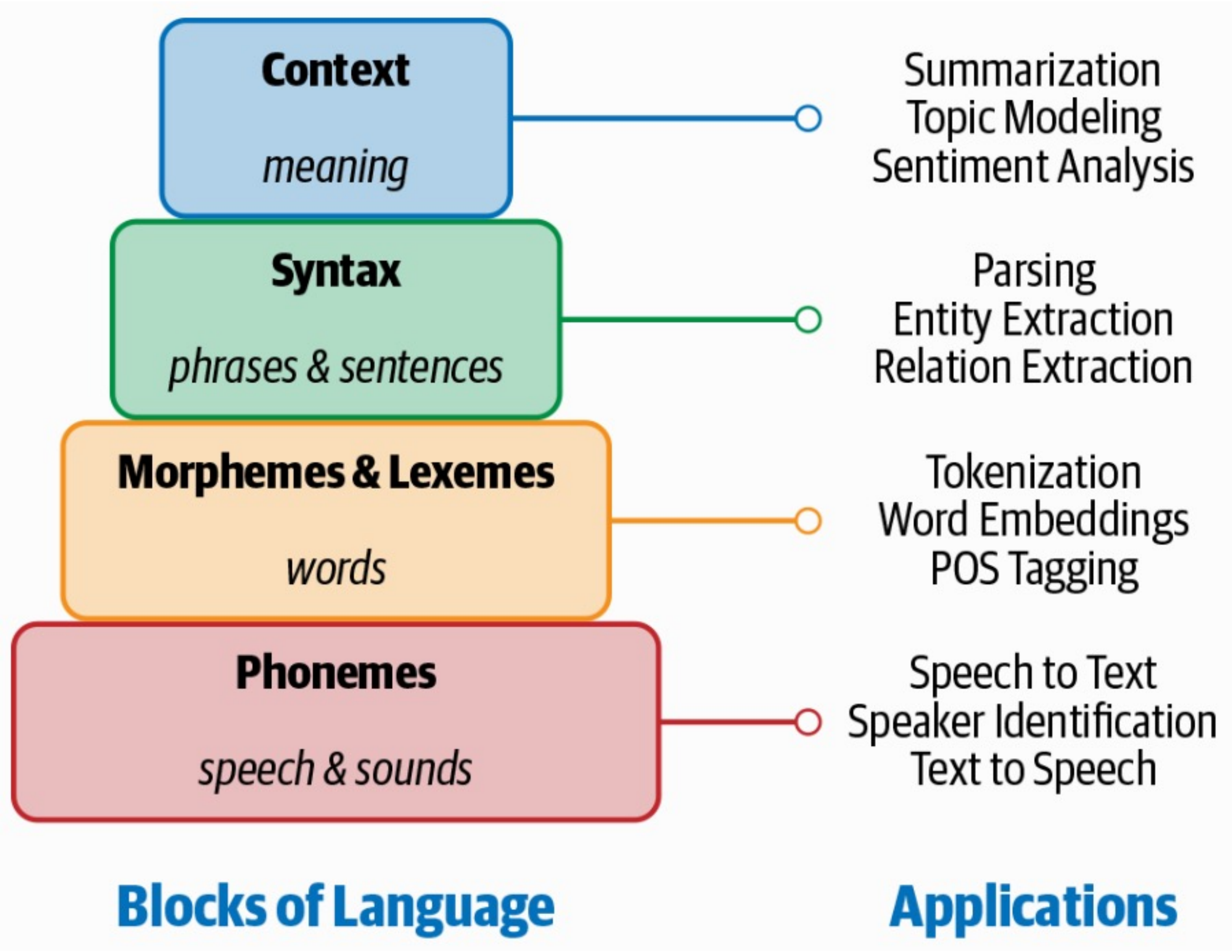
# NLP Tasks




Source: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.

# Building Blocks of Language and Applications




Source: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.

# Morpheme Examples

unbreakable
*un + break + able*

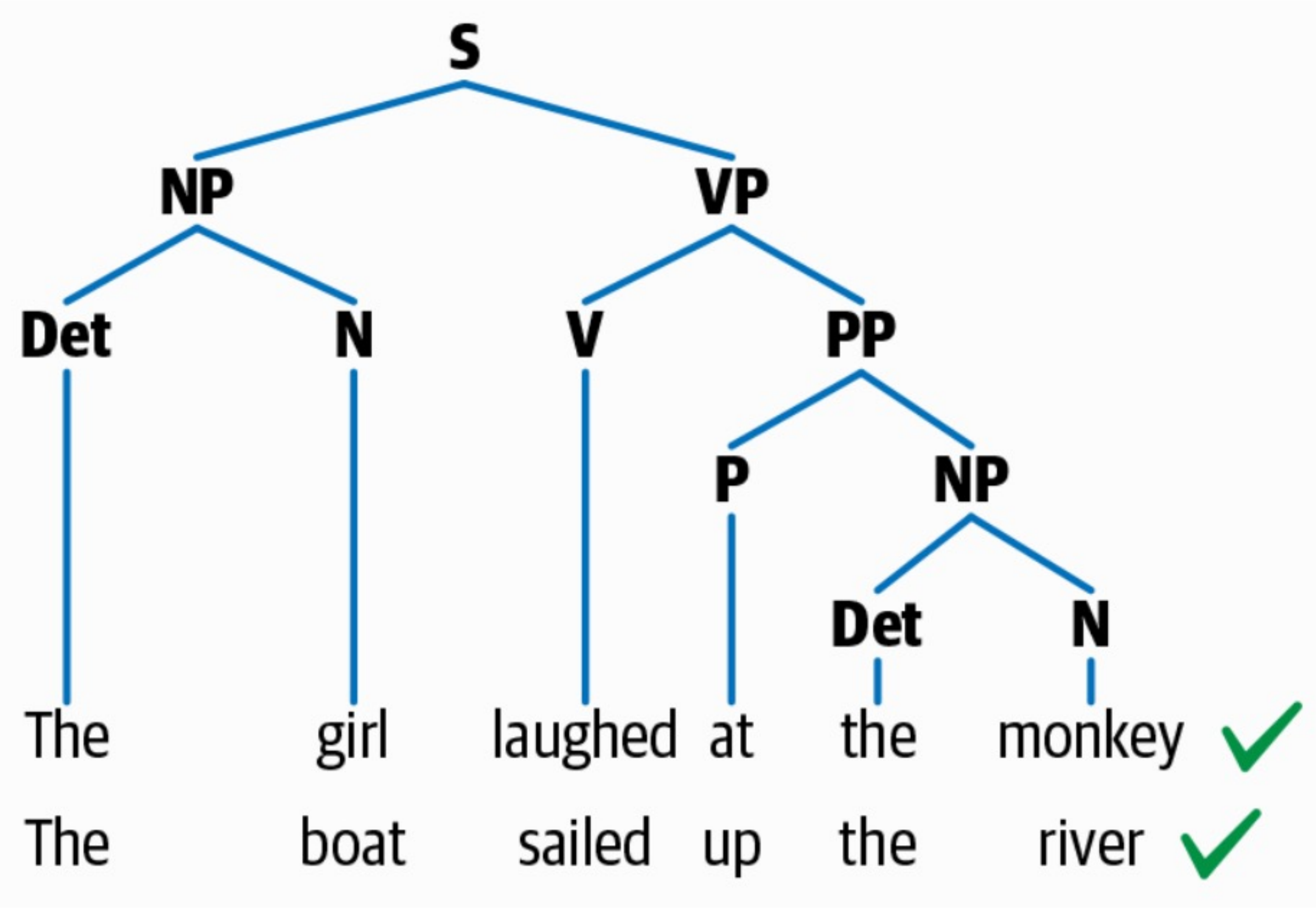
cats
*cat + s*

tumbling
*tumble + ing*

unreliability
*un + rely + able + ity*

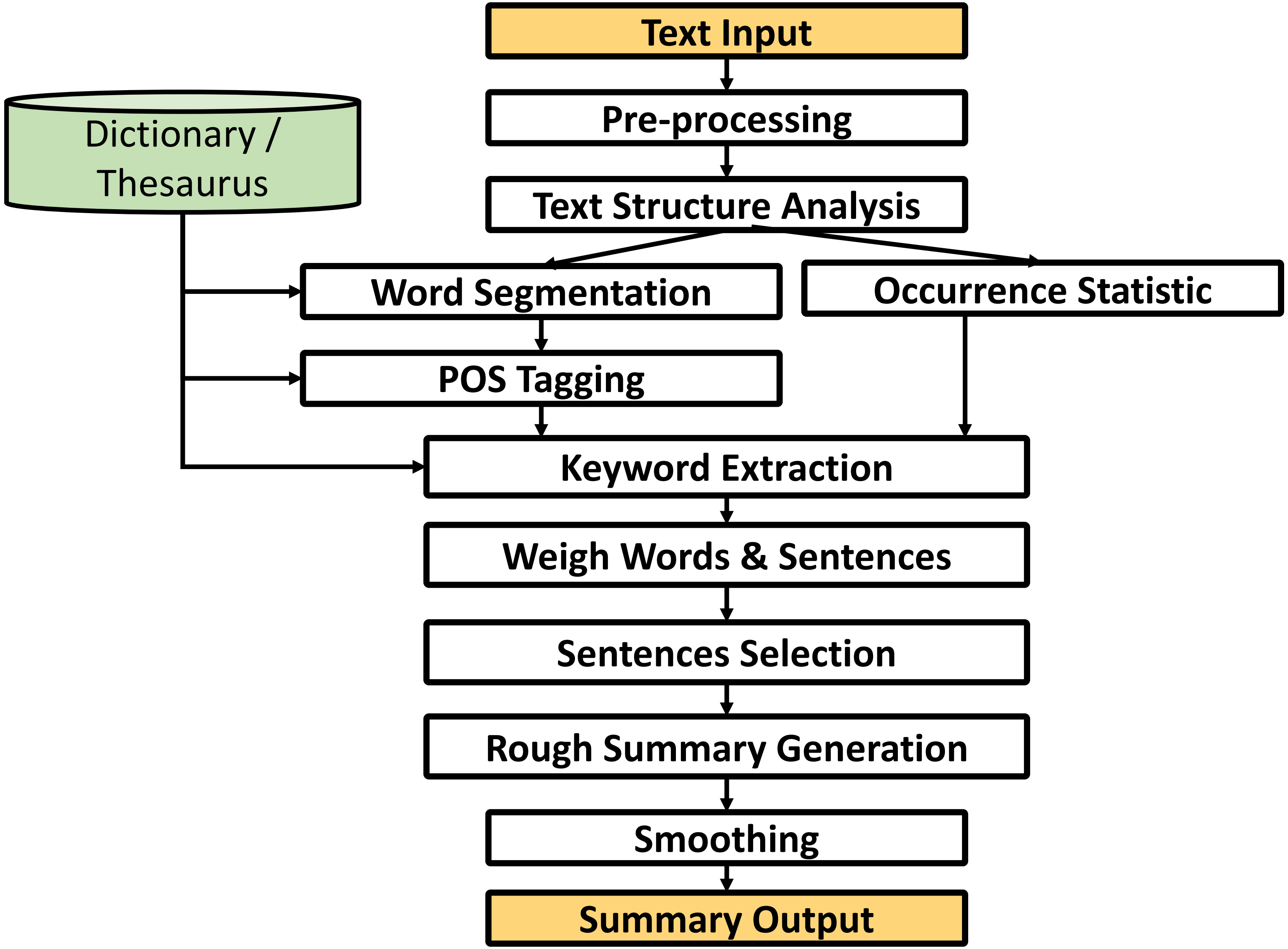
Source: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.

# Syntactic Structure



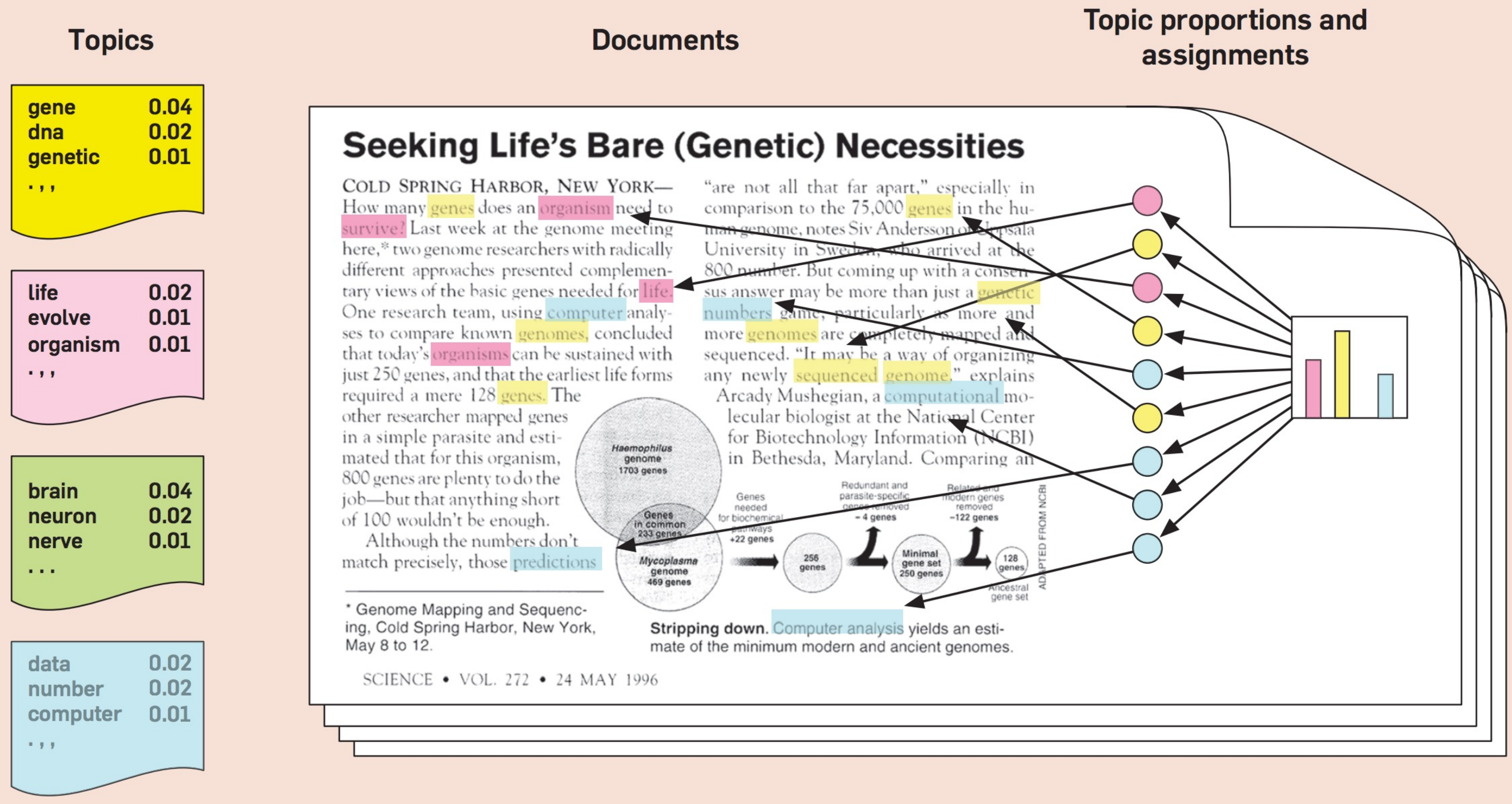

Source: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.

# 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.

# Natural Language Processing (NLP)

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



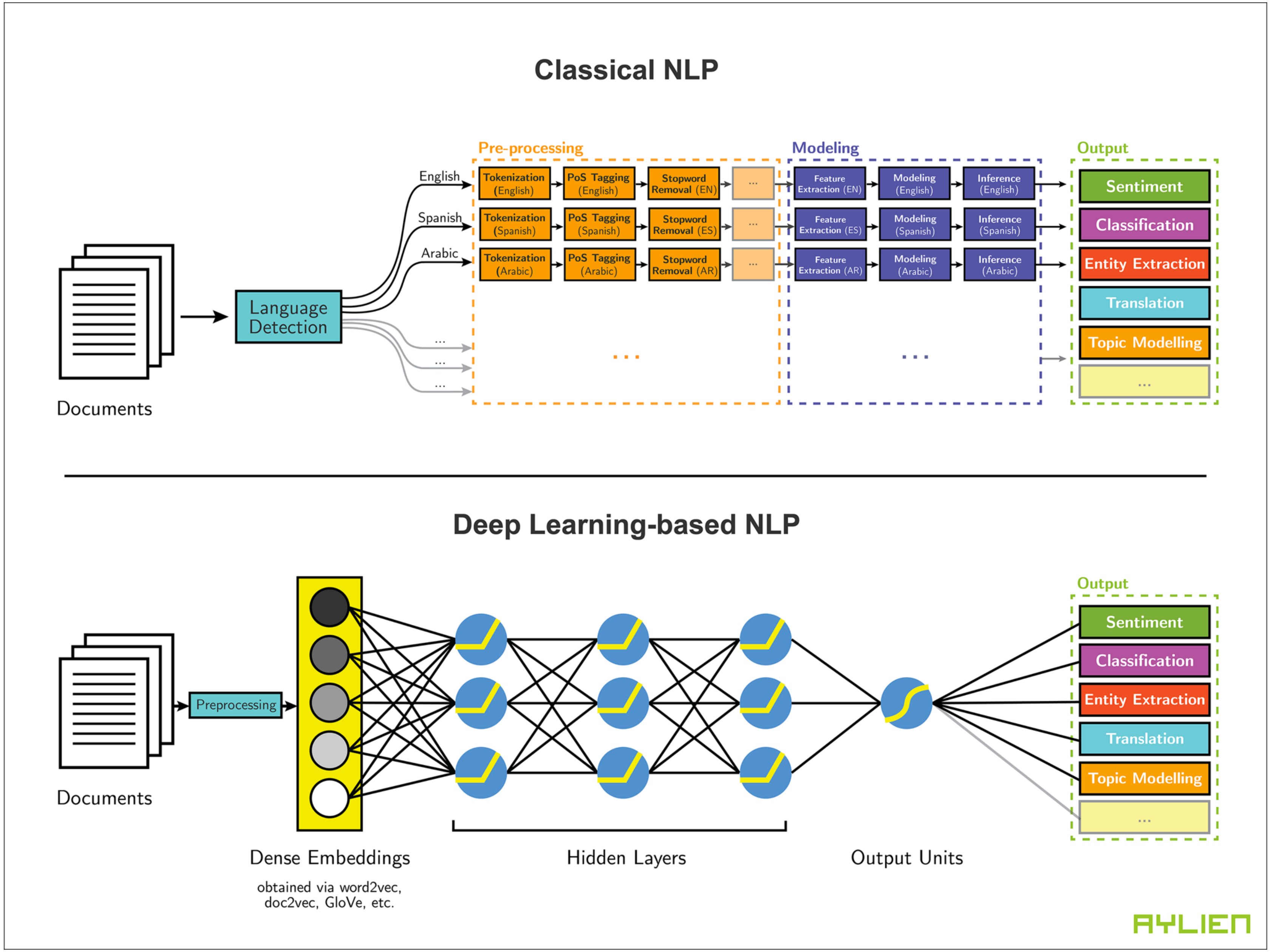
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

# 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

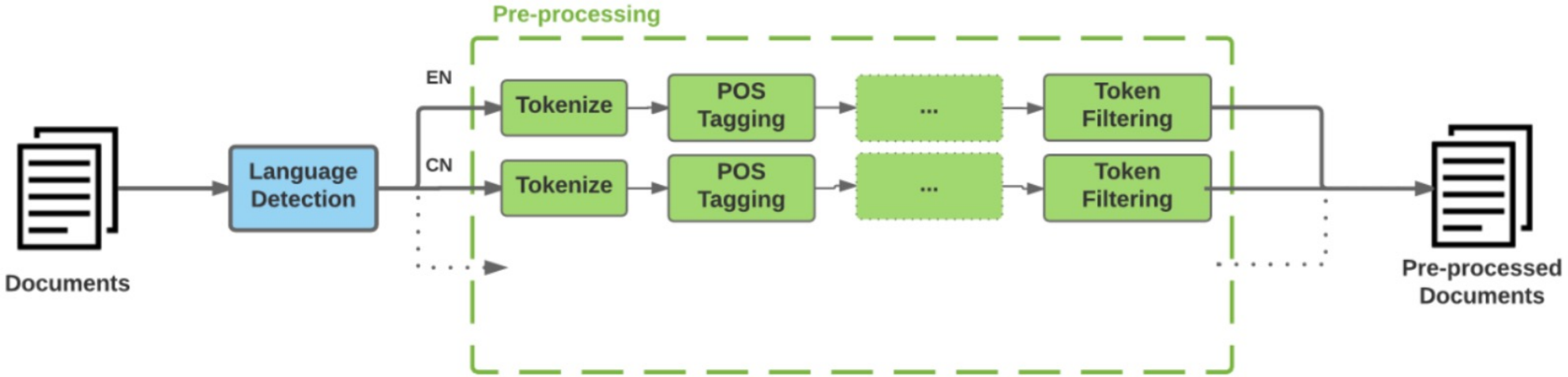
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

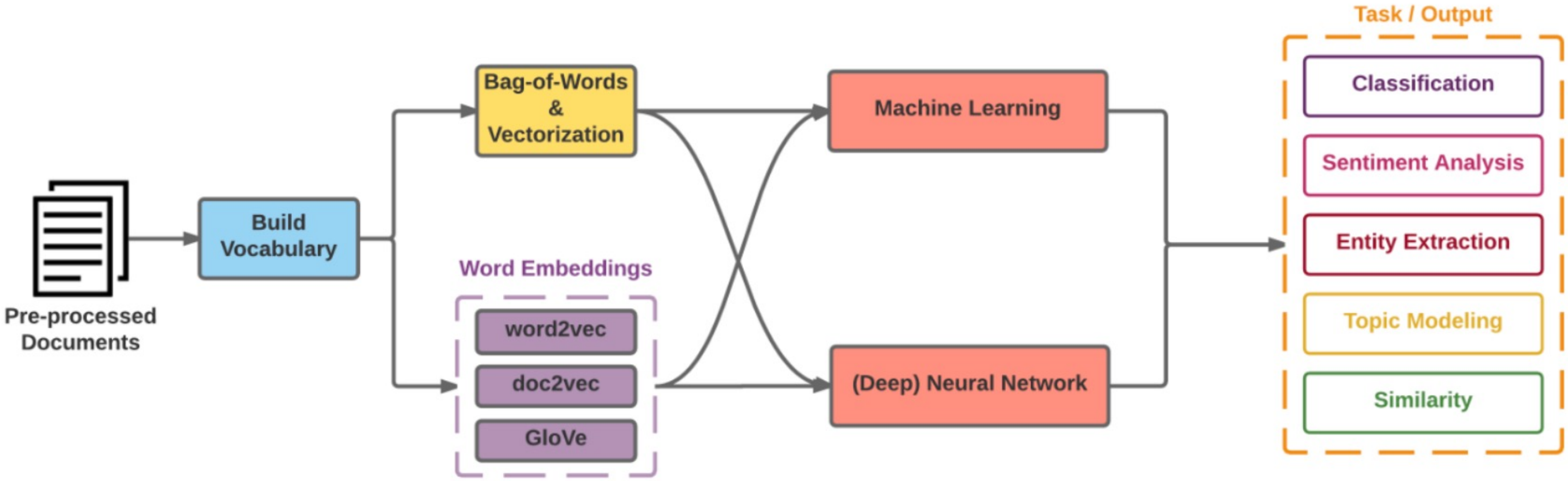
# NLP




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

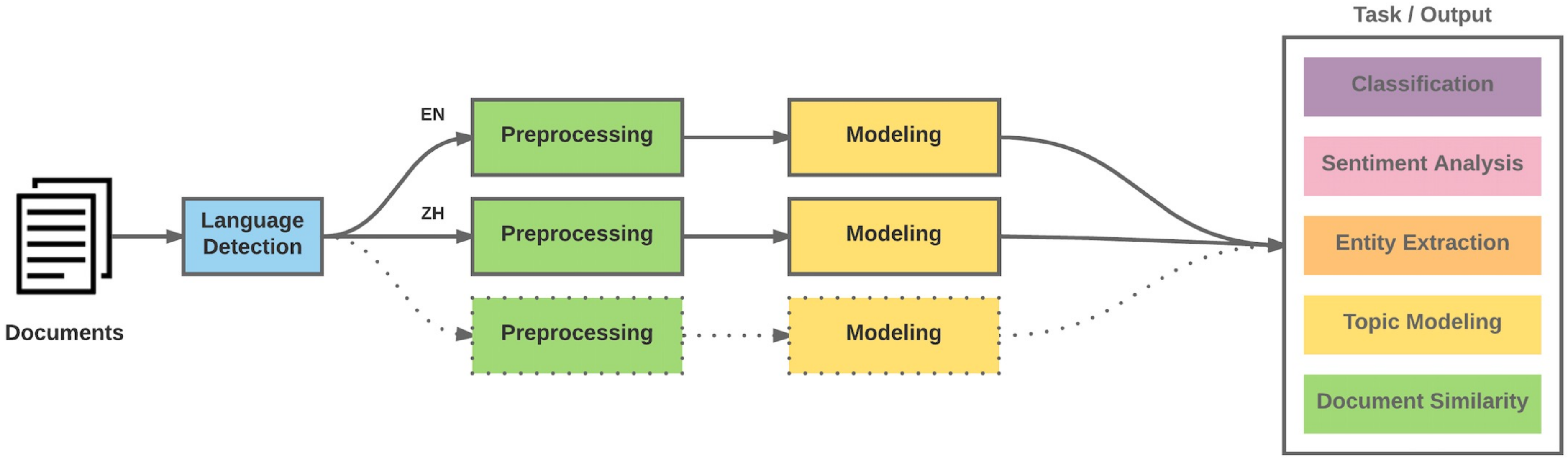
# Modern NLP Pipeline



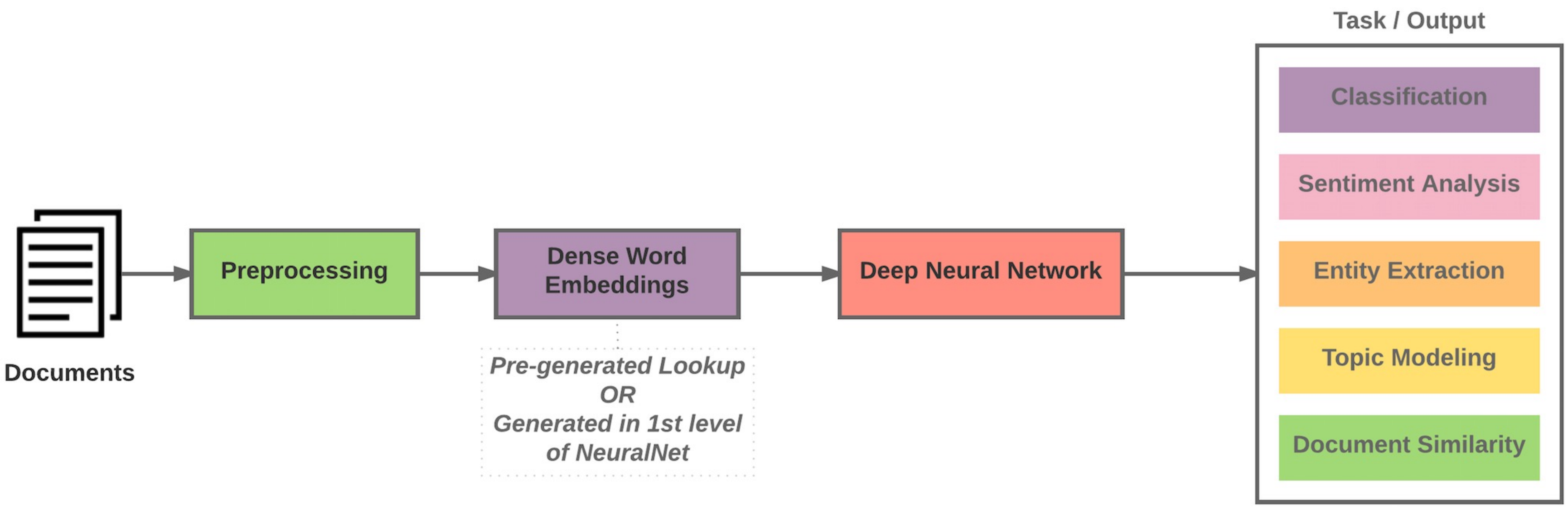

Source: https://github.com/fortiema/talks/blob/master/opendata2016sh/pragmatic-nlp-opendata2016sh.pdf

# Modern NLP Pipeline



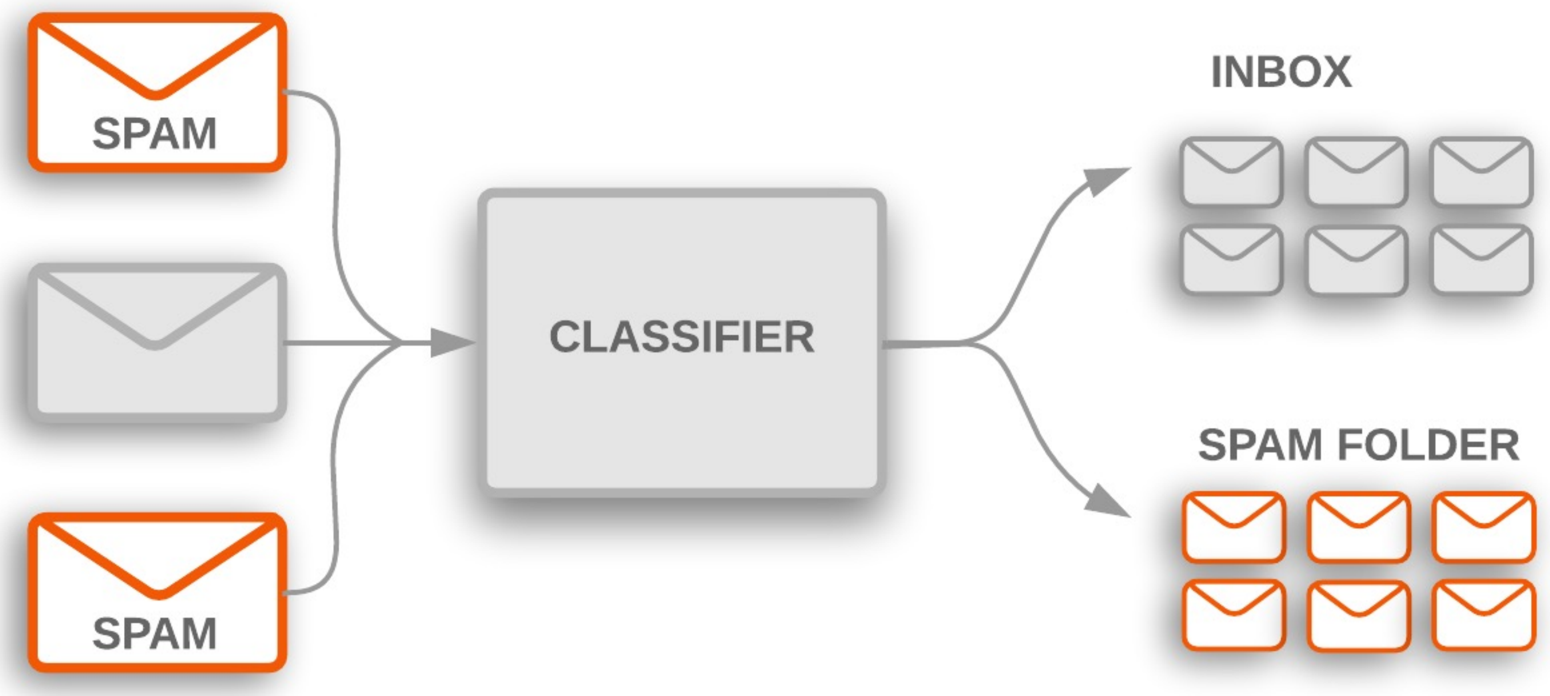

Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/

# Deep Learning NLP




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

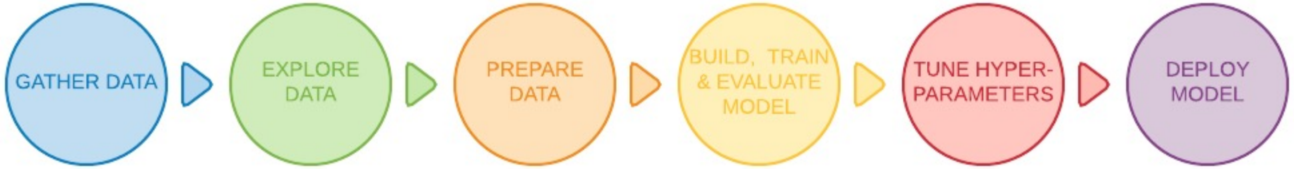
# Text Classification




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

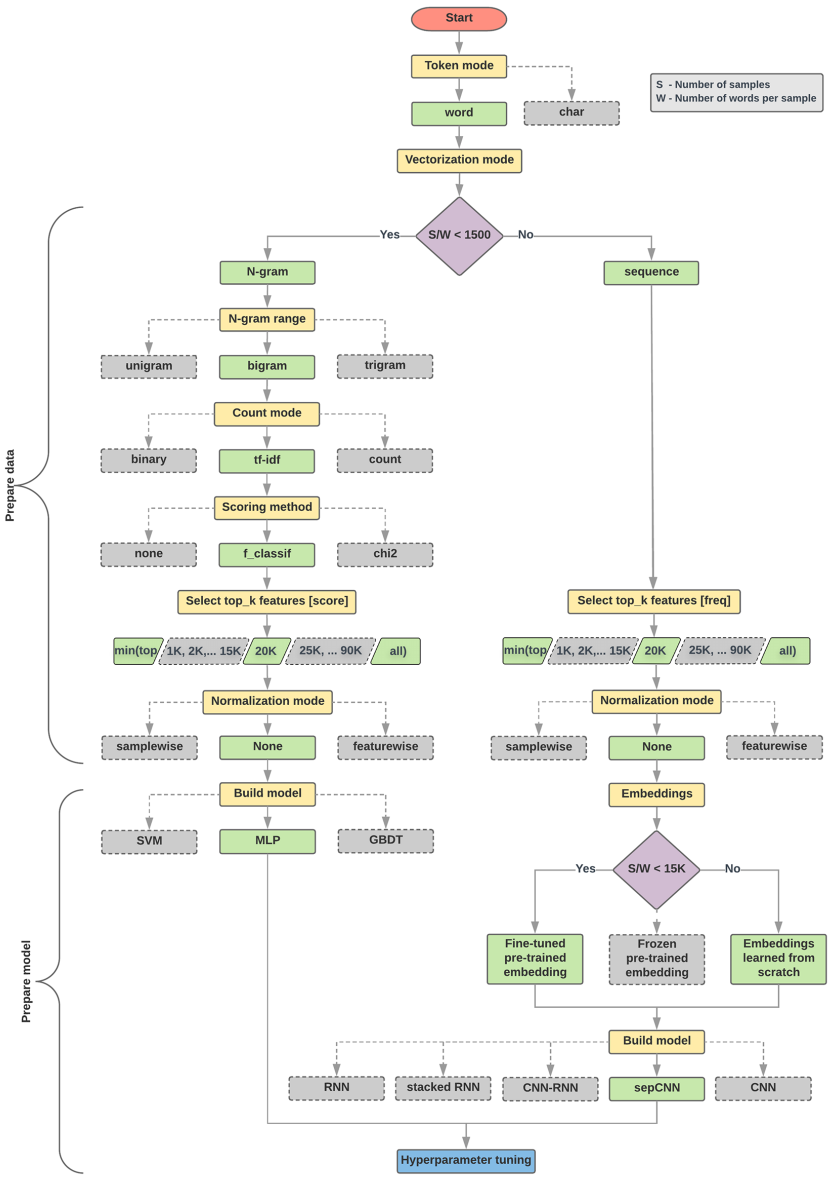
# Text Classification Workflow

- 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



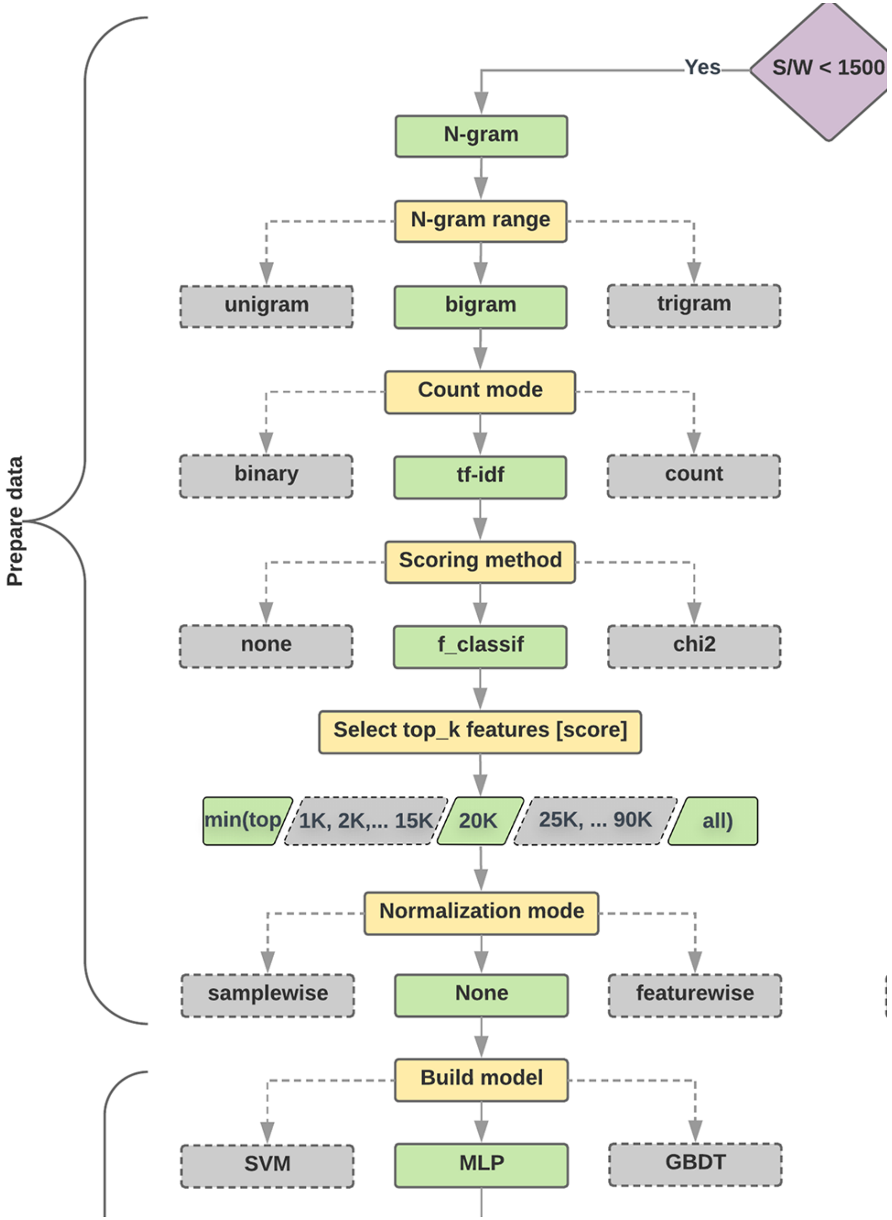

Source: https://developers.google.com/machine-learning/guides/text-classification/

# Text Classification Flowchart



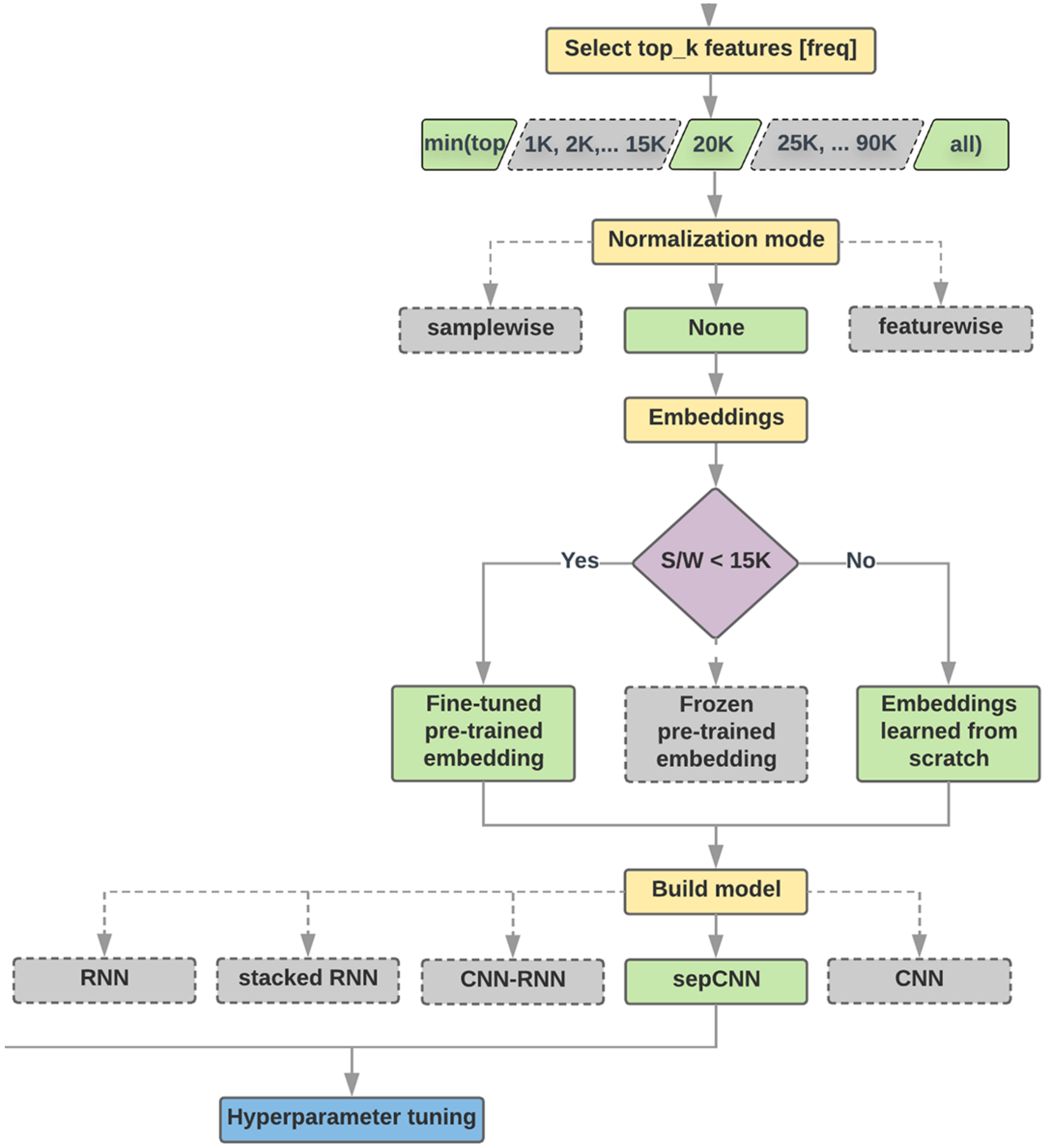

Source: https://developers.google.com/machine-learning/guides/text-classification/step-2-5

# 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

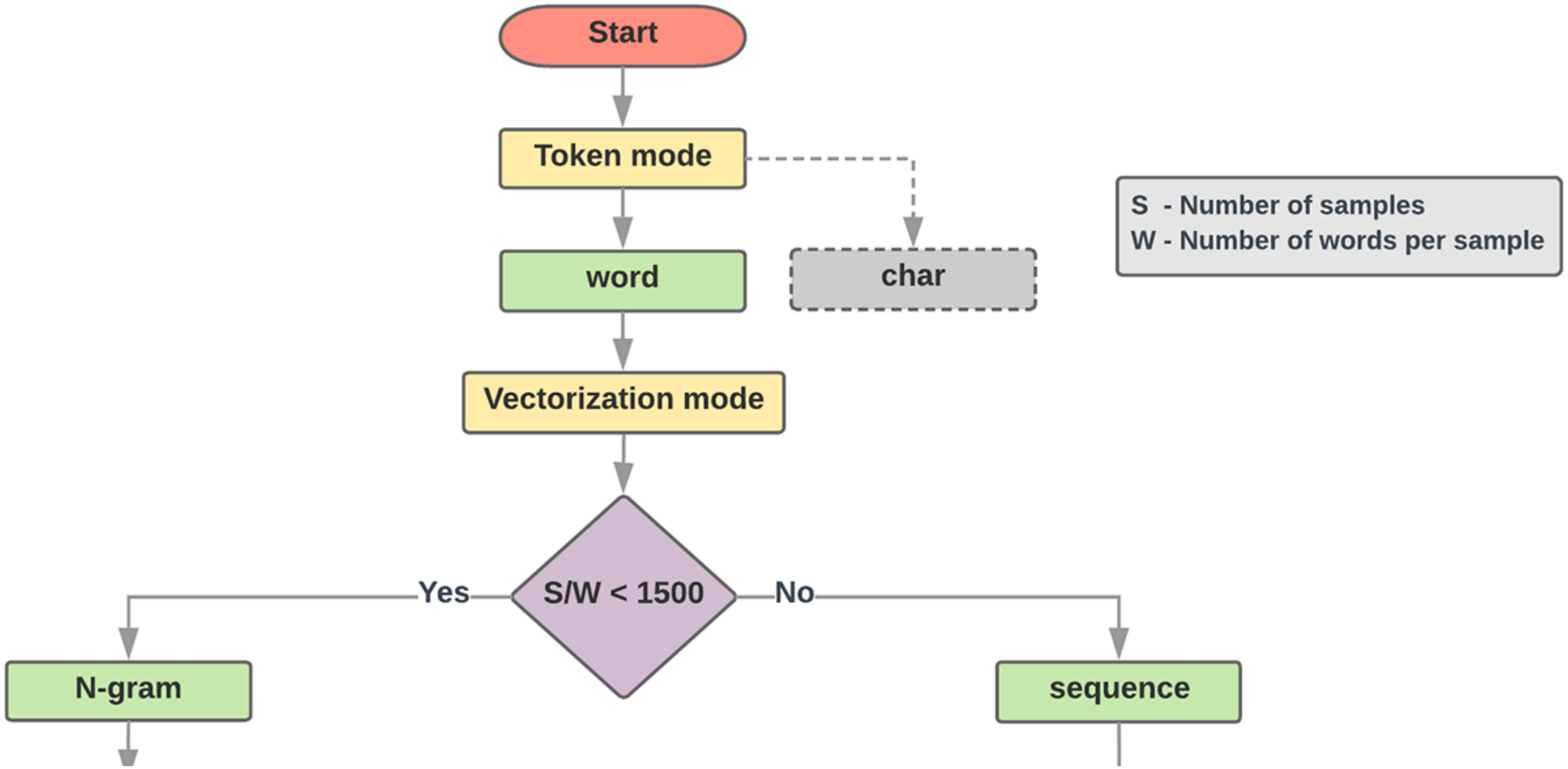



Source: https://developers.google.com/machine-learning/guides/text-classification/step-2-5

# Step 2.5: Choose a Model

## Samples/Words < 1500

## 150,000/100 = 1500



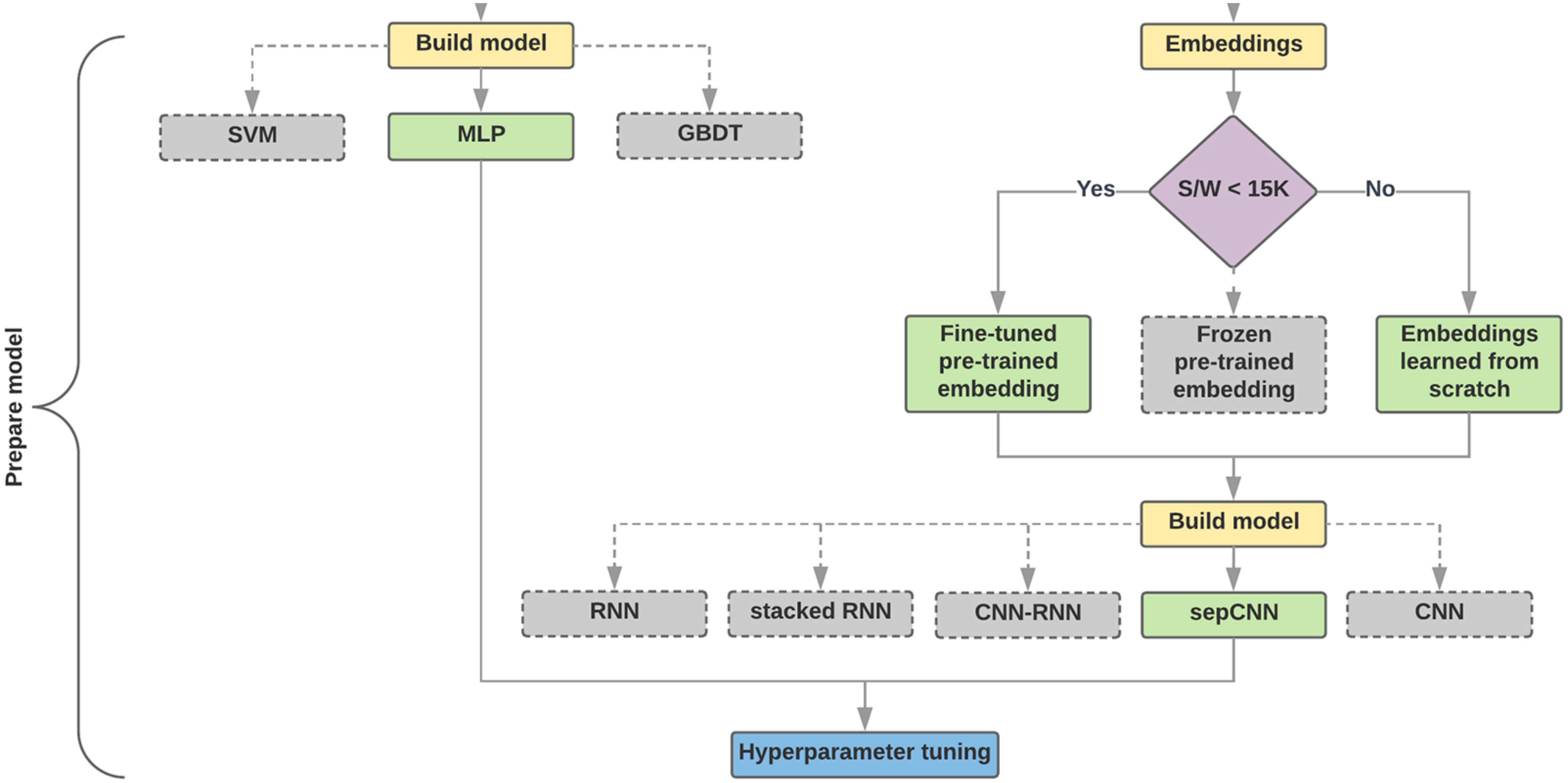

IMDb review dataset,
the samples/words-per-sample ratio is ~ 144

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

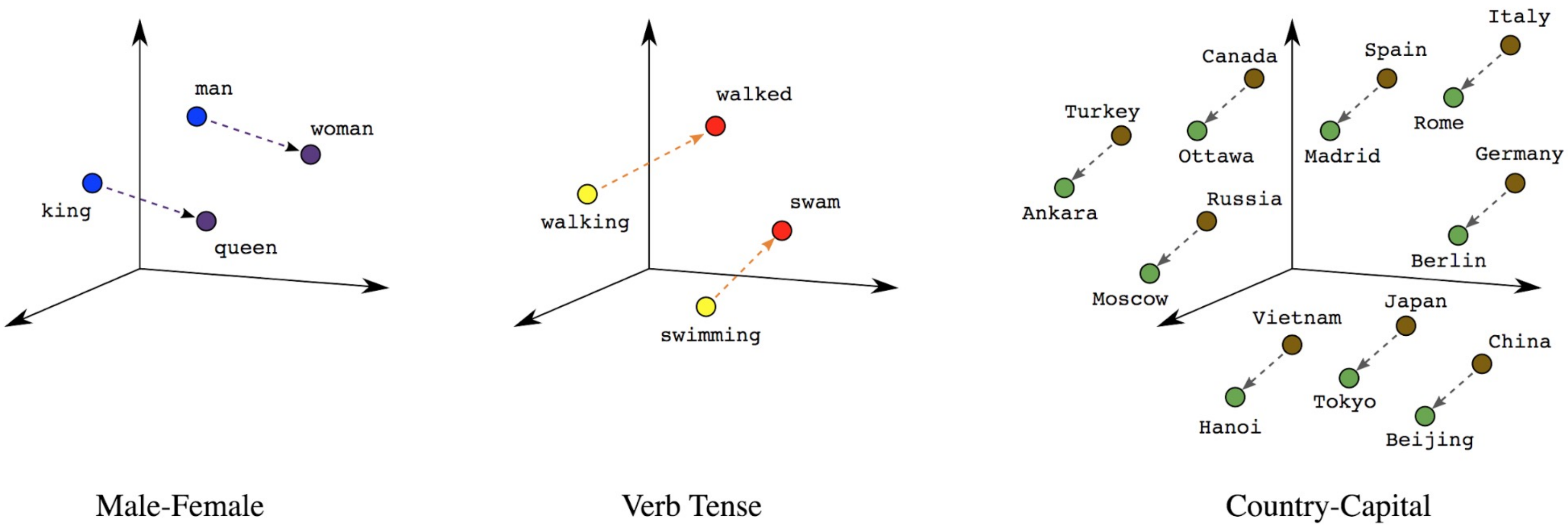
```
                    'The mouse ran up the clock' =

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]
```

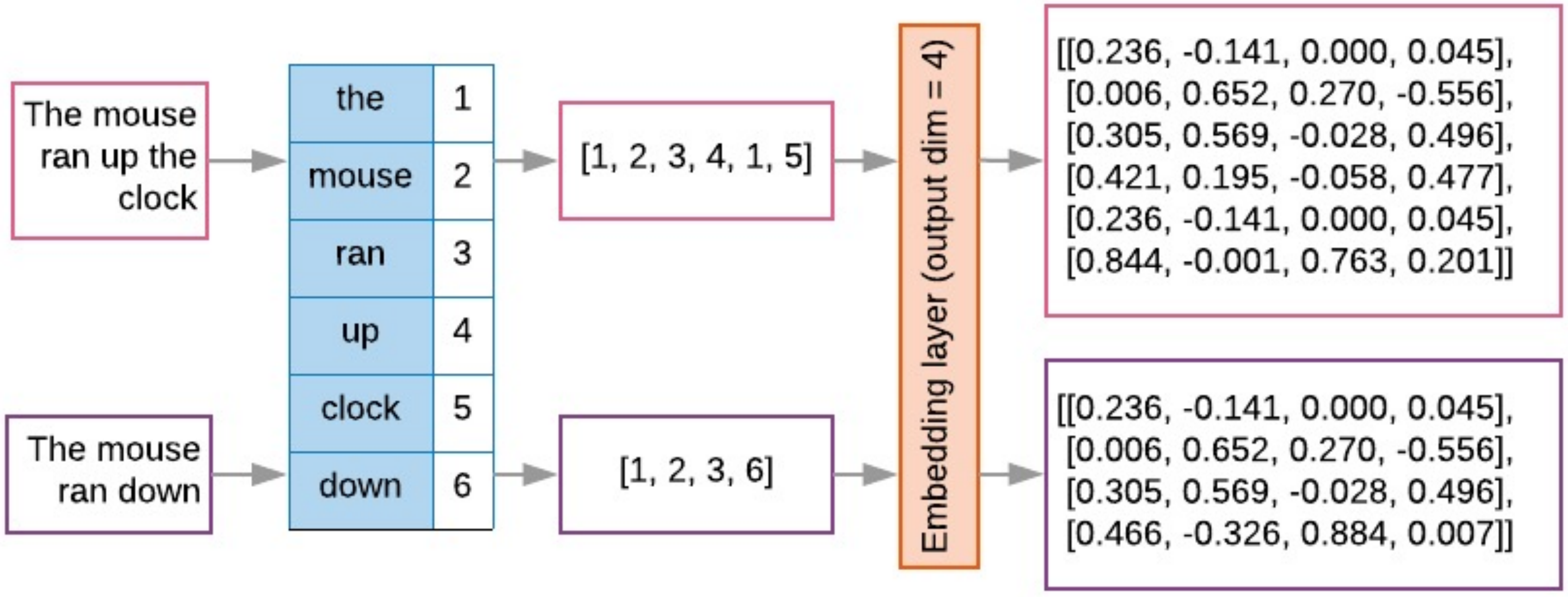
Source: https://developers.google.com/machine-learning/guides/text-classification/step-3

# Word embeddings




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

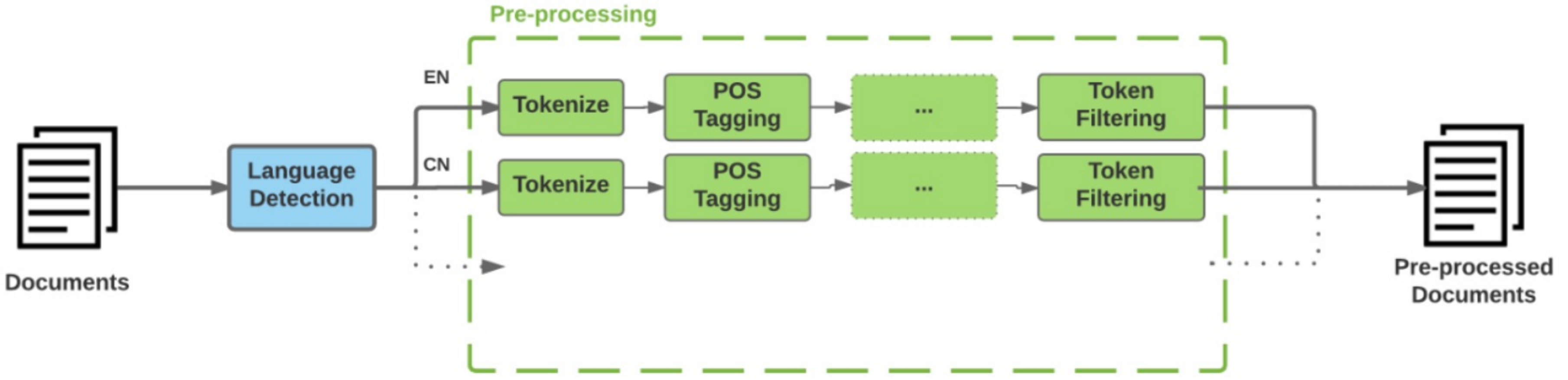
# Vector Representations of Words

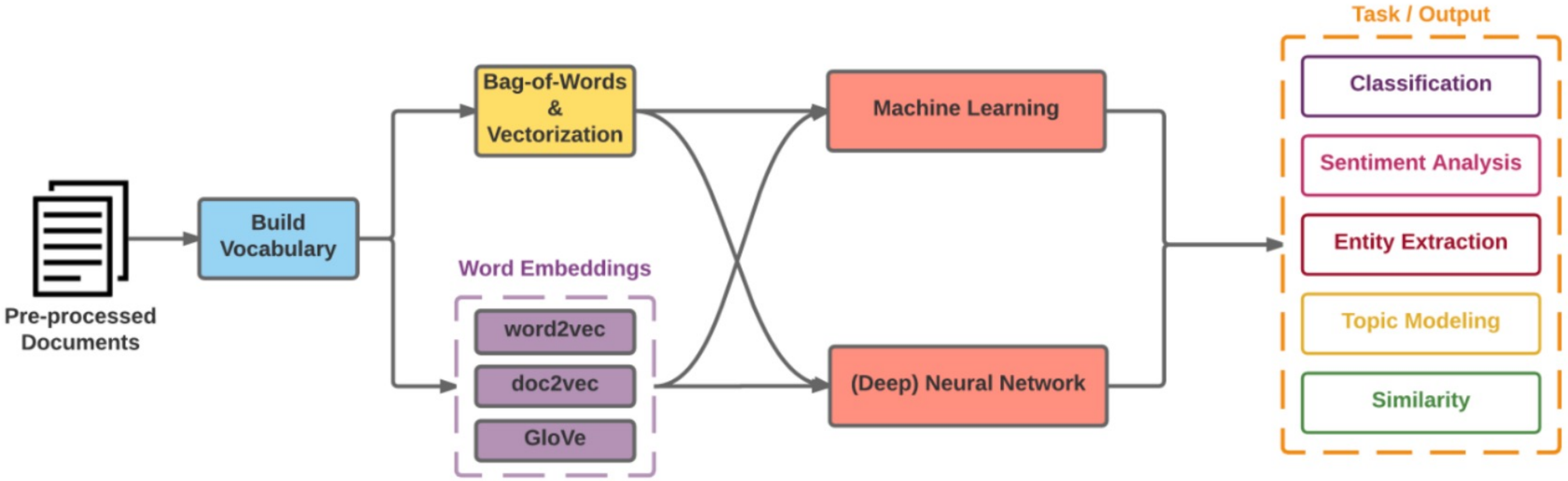
# Word Embeddings

# 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.

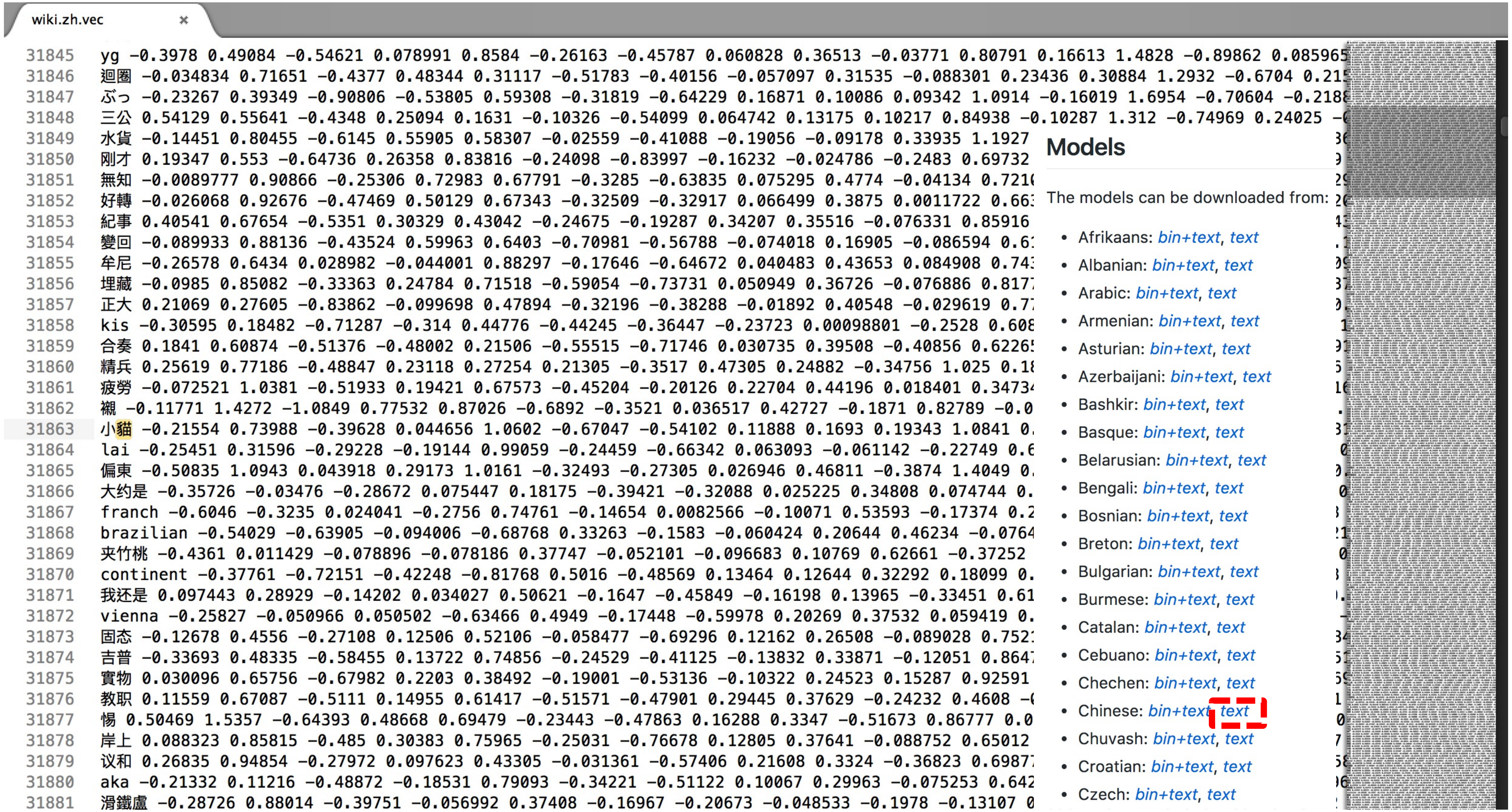
These vectors in dimension 300 were obtained using the skip-gram model with default parameters.

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).

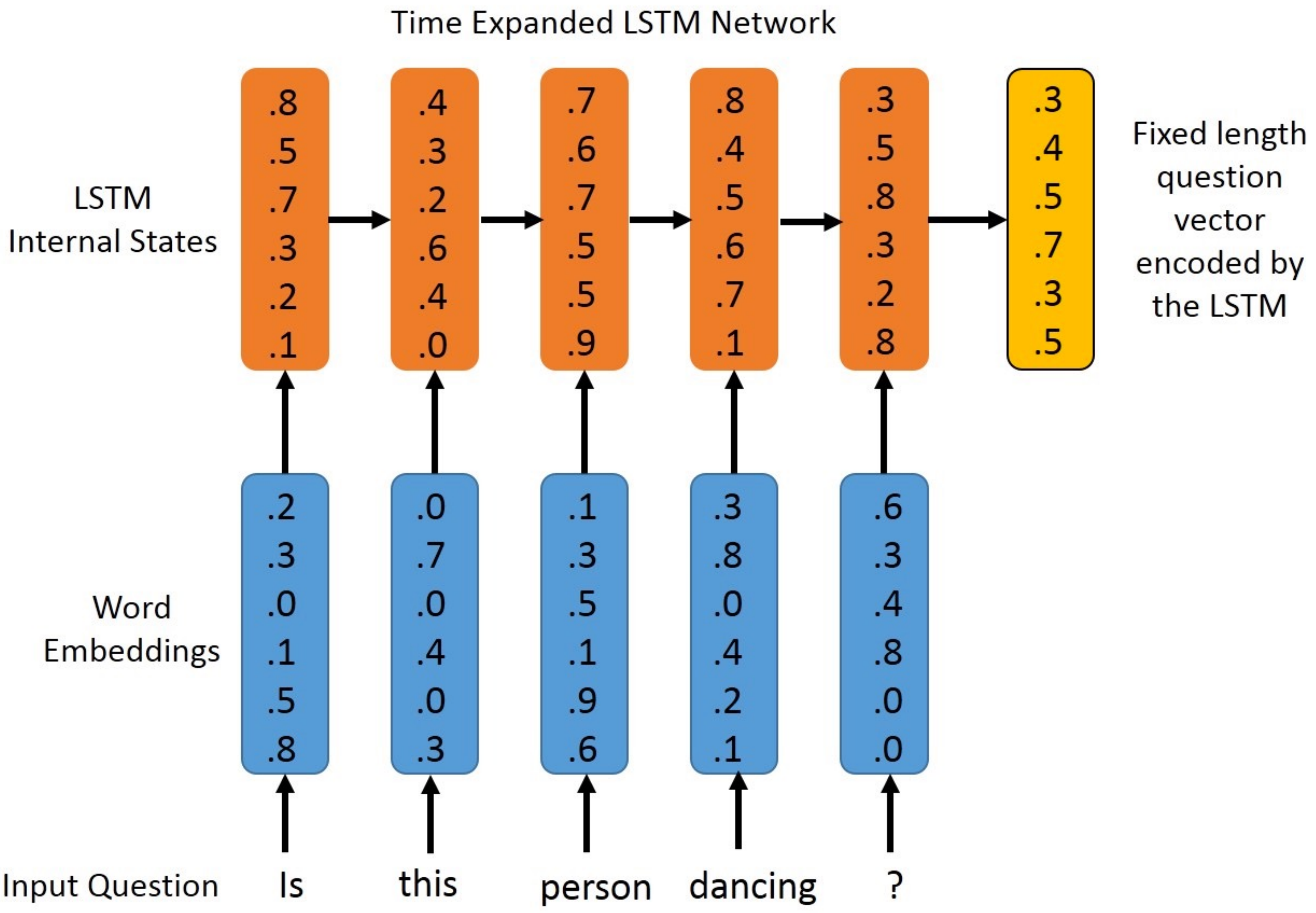
# Facebook Research FastText Word2Vec: wiki.zh.vec

## (861MB) (332647 word 300 vec)

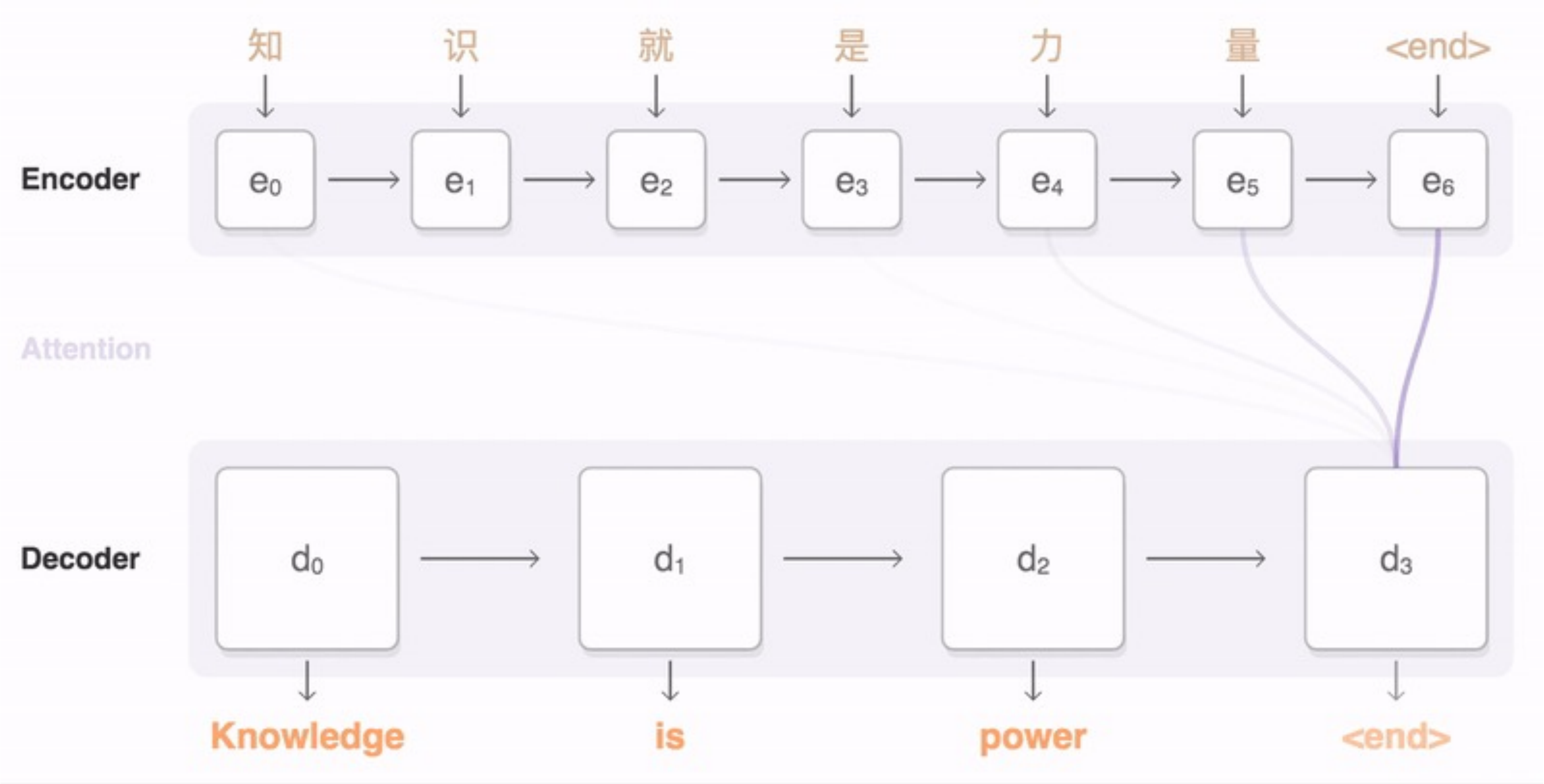



https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

# Word Embeddings in LSTM RNN



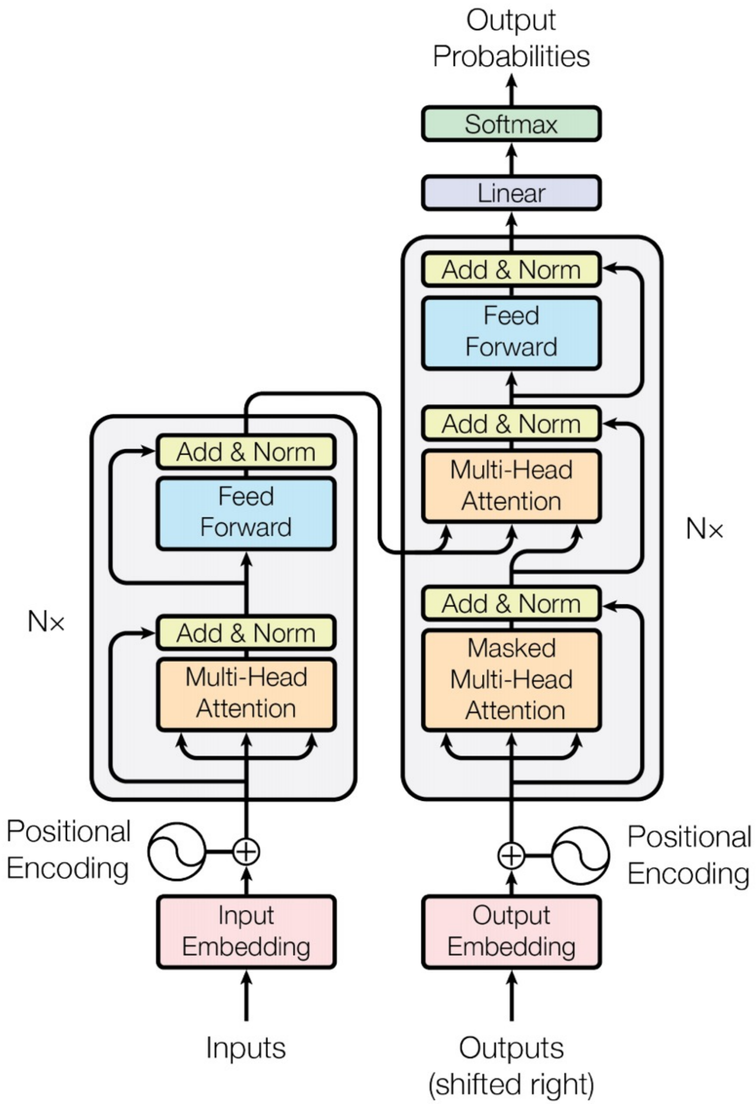
# Sequence to Sequence (Seq2Seq)




Source: https://google.github.io/seq2seq/

# Transformer (Attention is All You Need)

## (Vaswani et al., 2017)

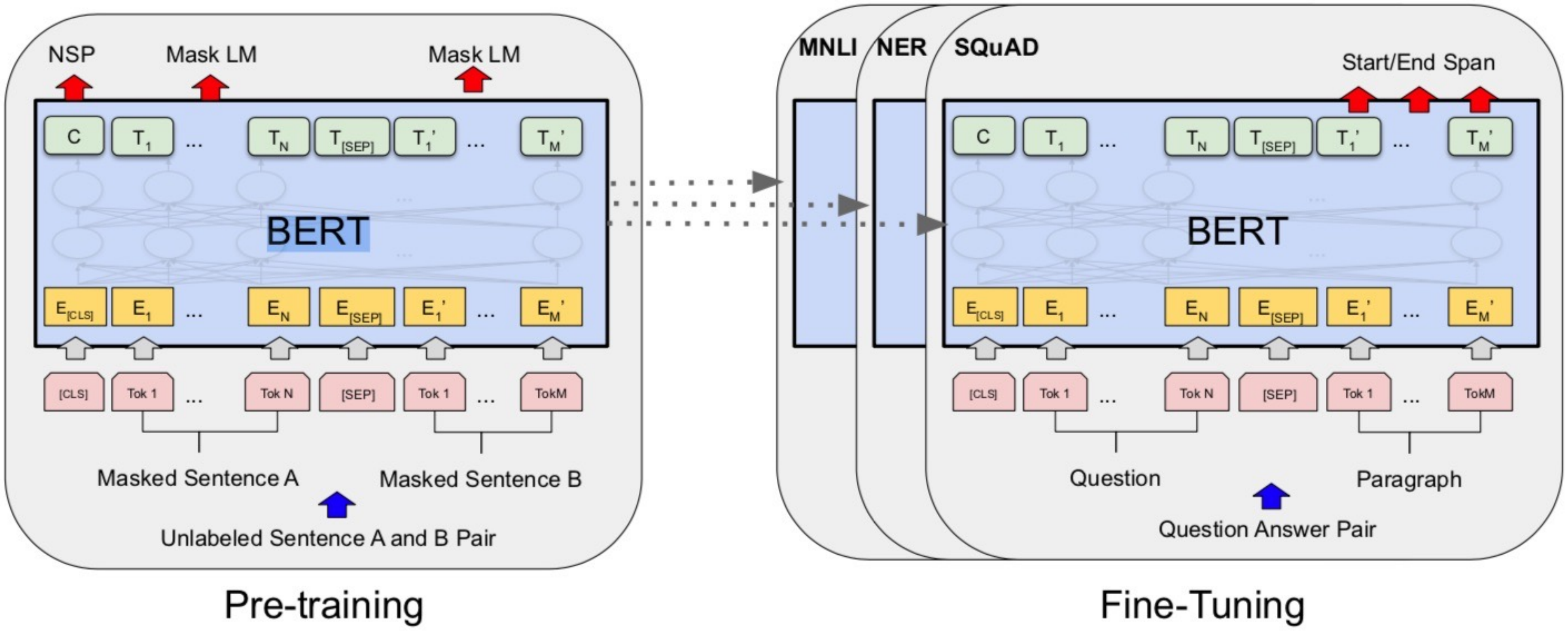



Source: Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

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

BERT (Bidirectional Encoder Representations from Transformers)

## Overall pre-training and fine-tuning procedures for BERT




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: Pre-training of Deep Bidirectional Transformers for Language Understanding

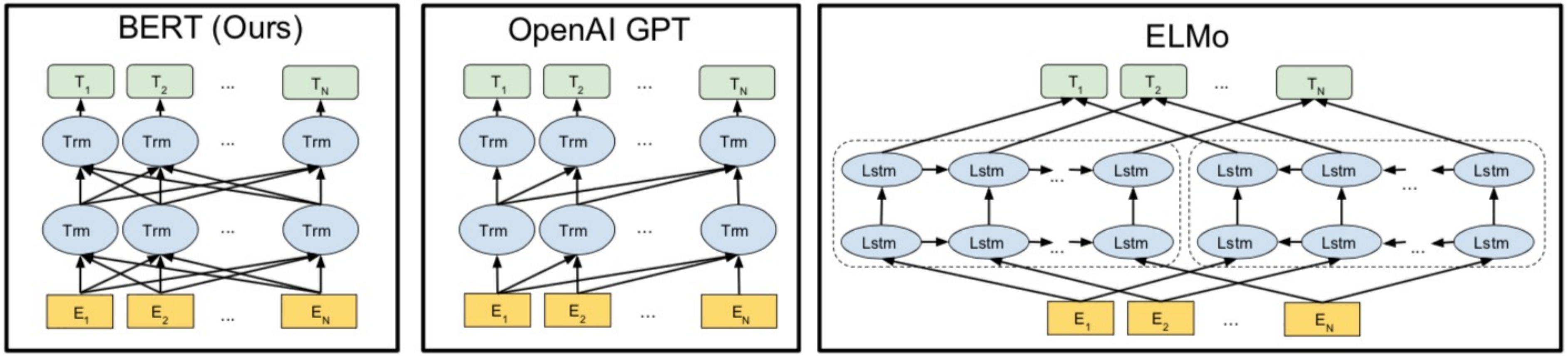
**BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**

**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**

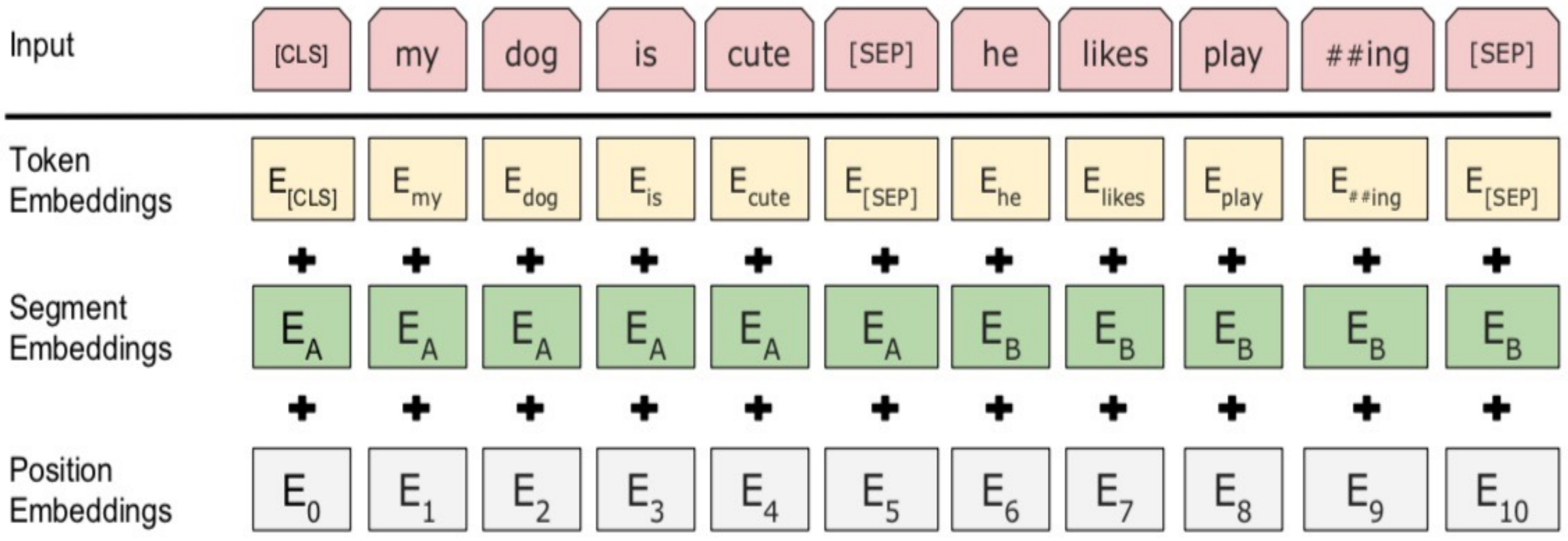
**BERT** uses a bidirectional Transformer.
**OpenAI GPT** uses a left-to-right Transformer.
**ELMo** uses the concatenation of independently trained left-to-right and right- to-left LSTM to generate features for downstream tasks.
Among three, only BERT representations are jointly conditioned on both 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: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT (Bidirectional Encoder Representations from Transformers)

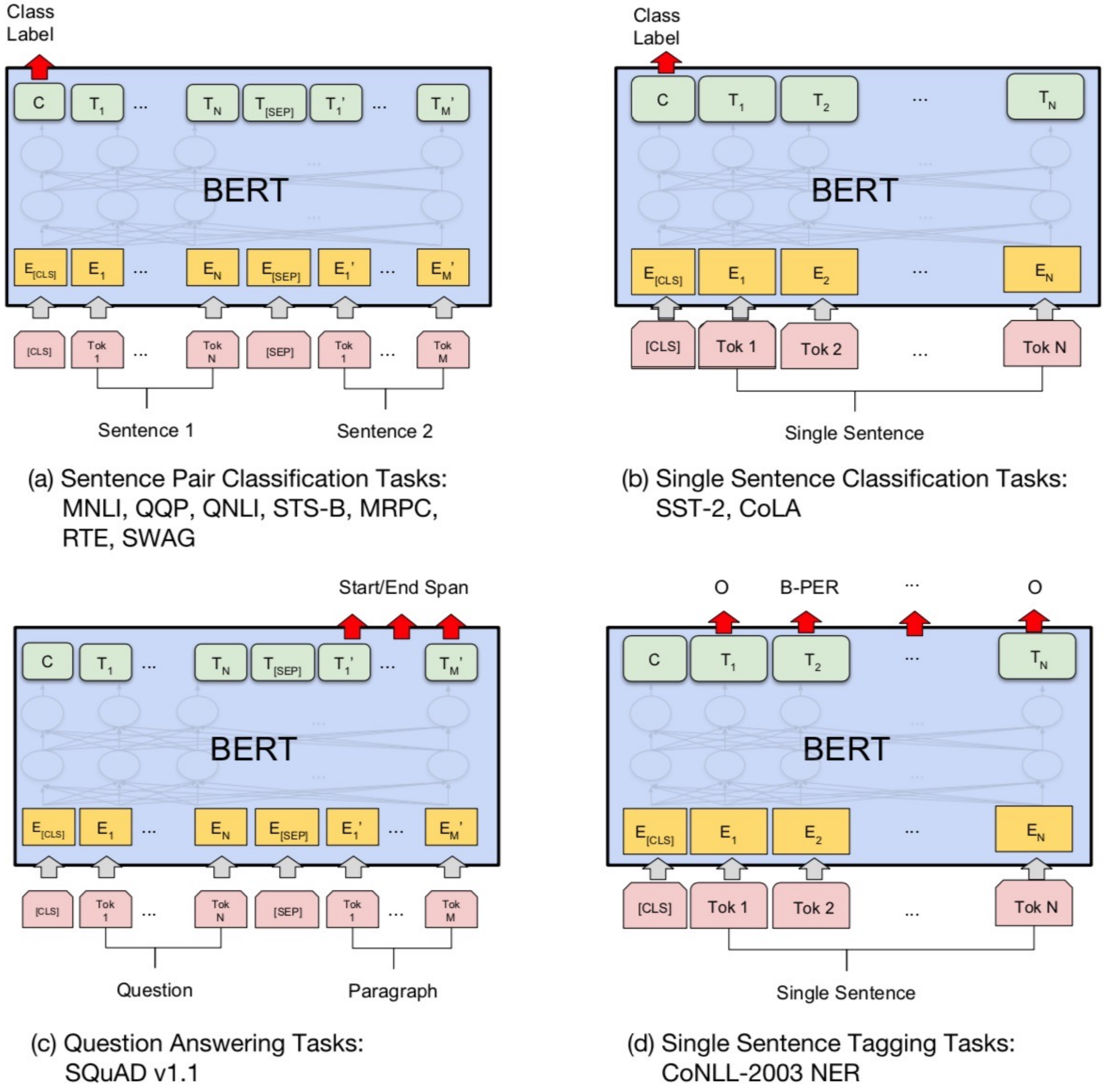
## BERT input representation




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.

# Fine-tuning BERT on NLP Tasks




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

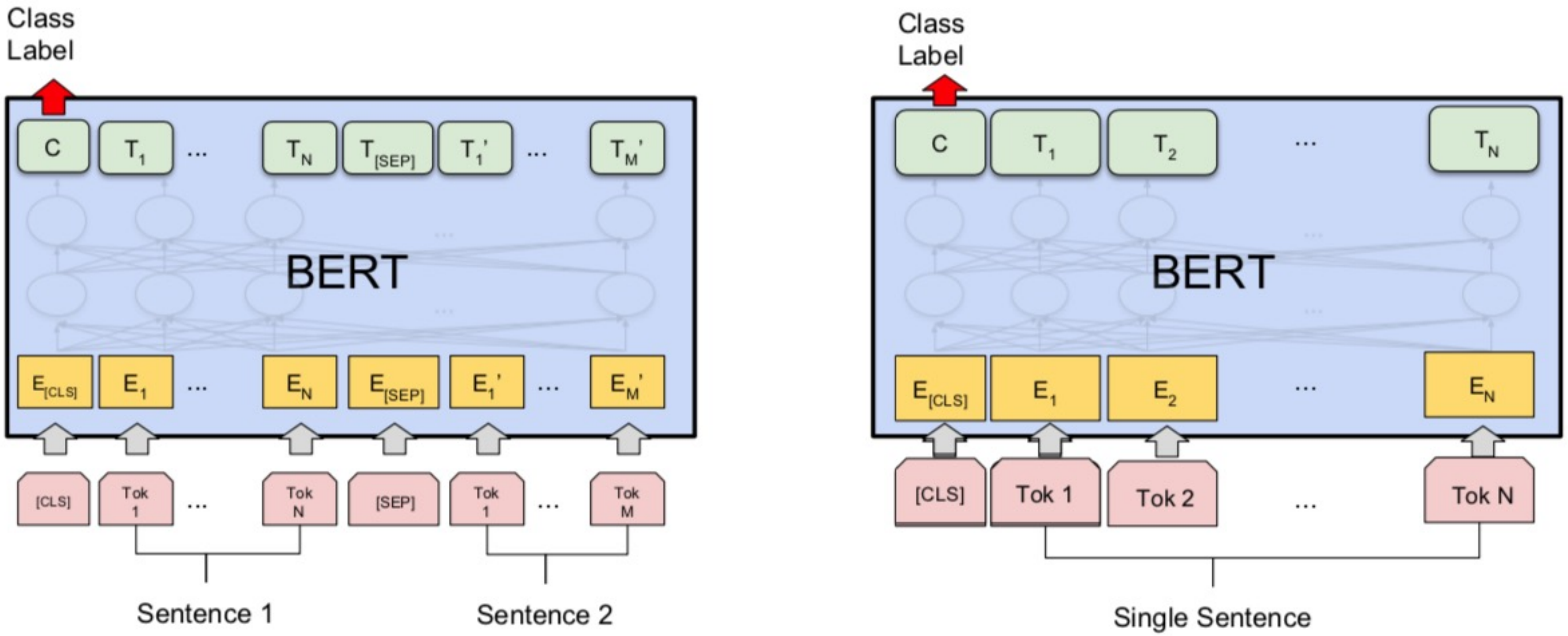
(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

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

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

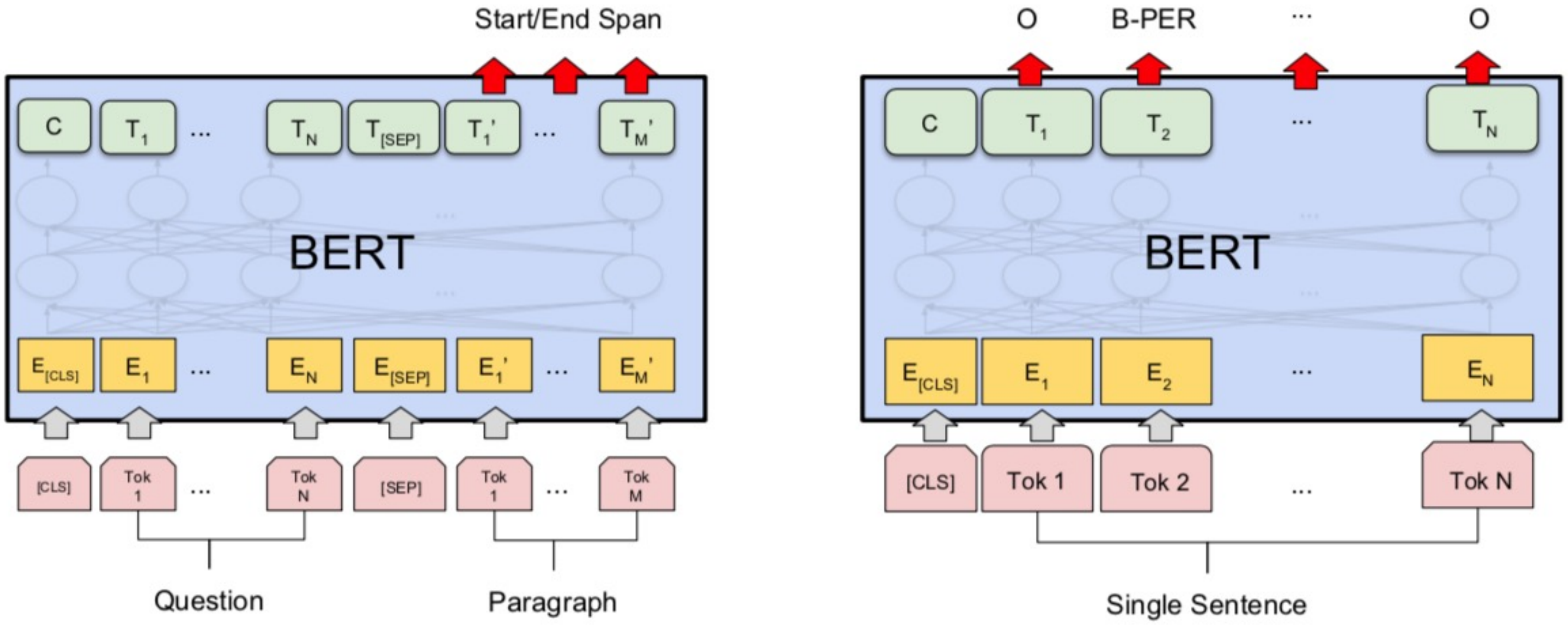



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(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

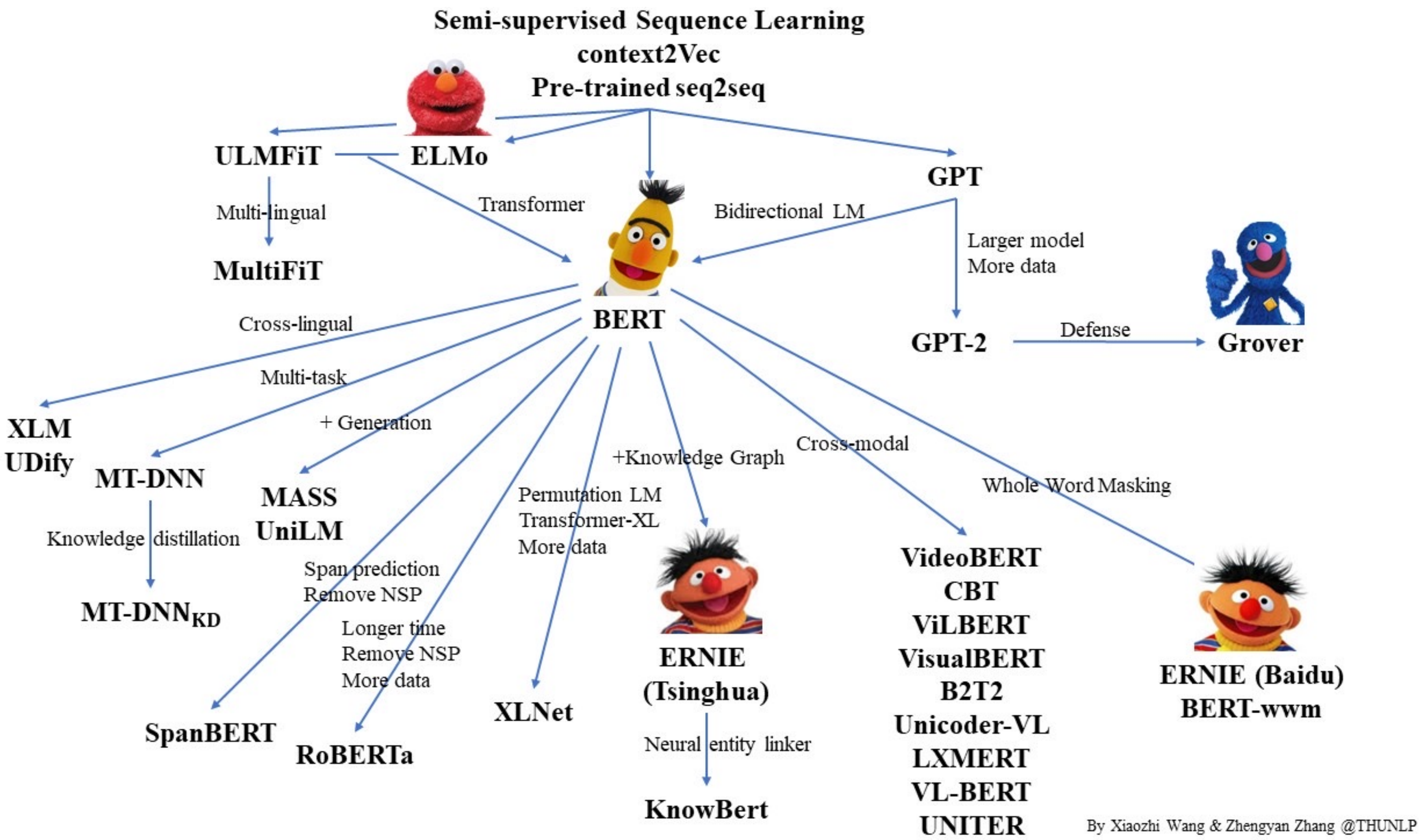
# GLUE Test results

| 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** |

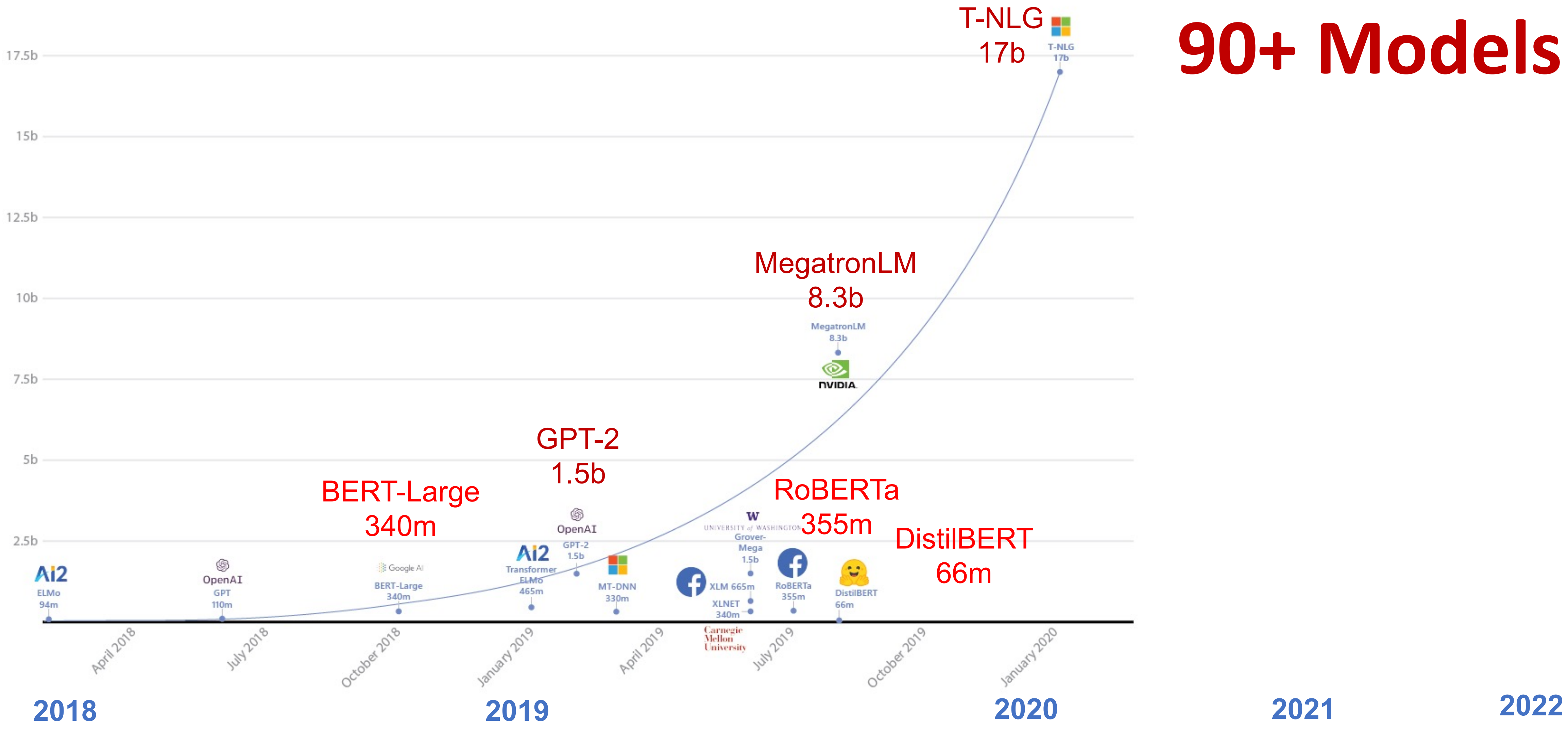
**MNLI**: Multi-Genre Natural Language Inference
**QQP**: Quora Question Pairs
**QNLI**: Question Natural Language Inference
**SST-2**: The Stanford Sentiment Treebank
**CoLA**: The Corpus of Linguistic Acceptability
**STS-B**:The Semantic Textual Similarity Benchmark
**MRPC**: Microsoft Research Paraphrase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

# Pre-trained Language Model (PLM)

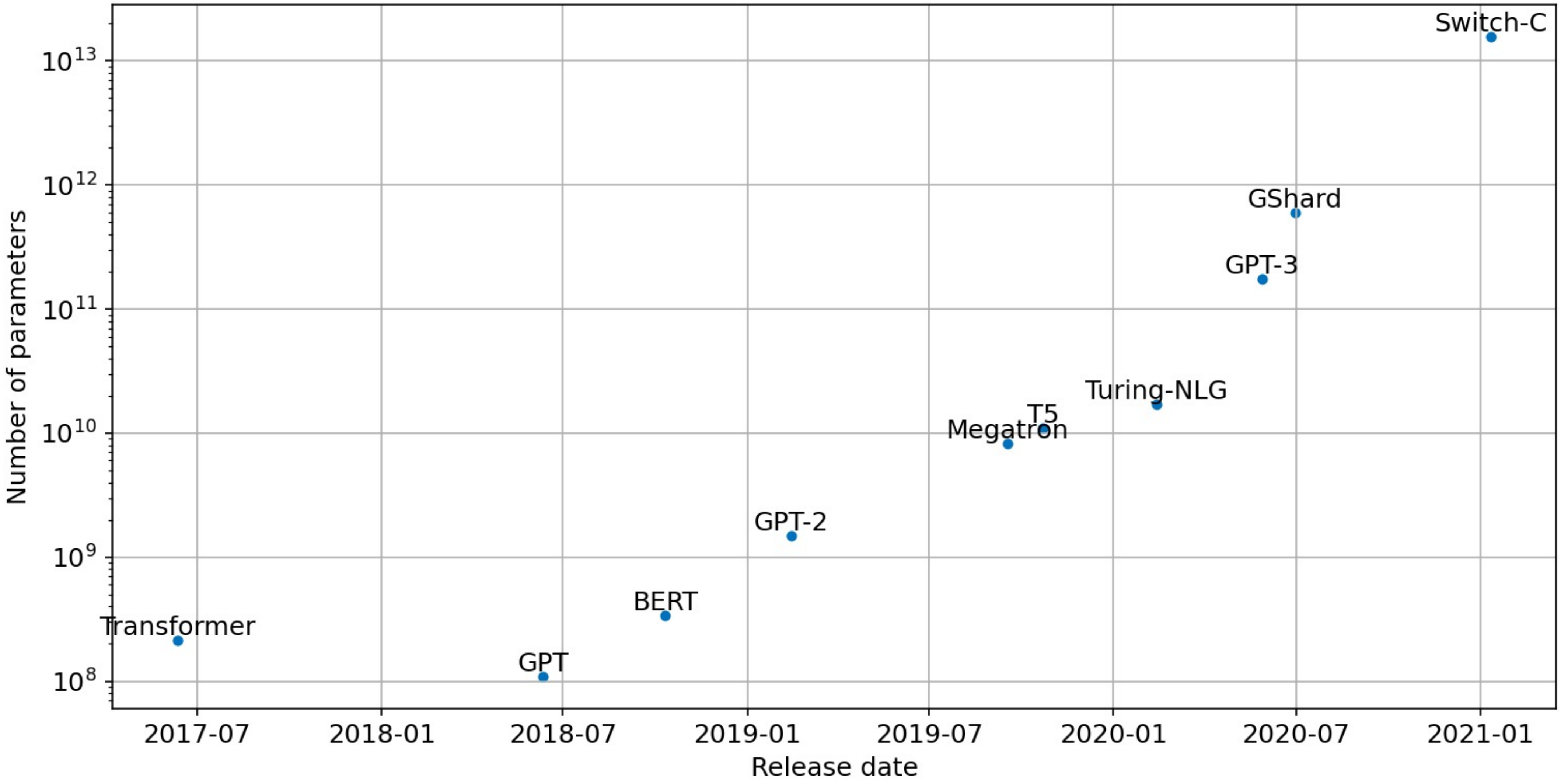


# Transformers Pre-trained Language Model

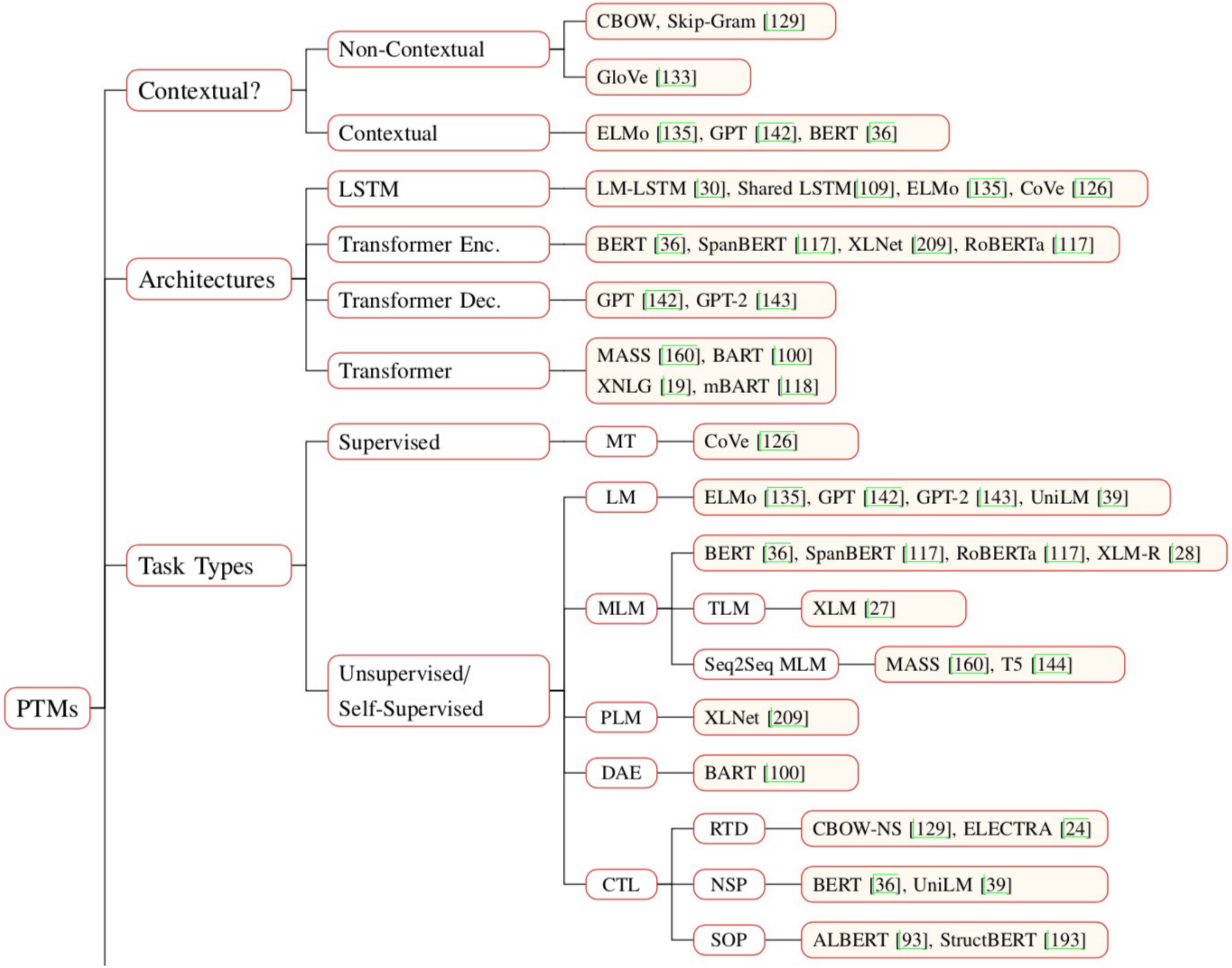



Source: https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

# Scaling Transformers

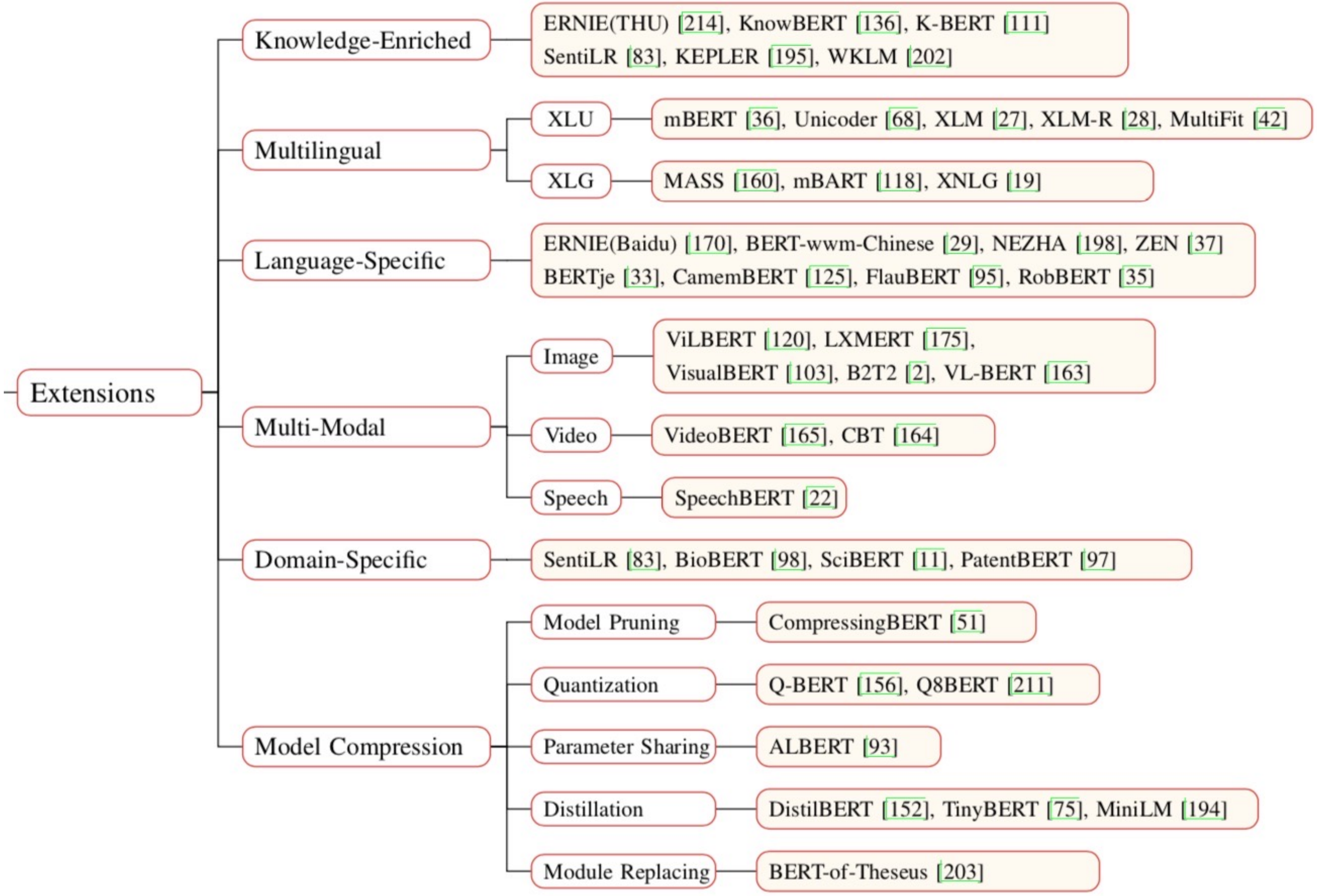


# Pre-trained Models (PTM)




Source: Qiu, Xipeng, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. "Pre-trained Models for Natural Language Processing: A Survey." arXiv preprint arXiv:2003.08271 (2020).

# Pre-trained Models (PTM)




Source: Qiu, Xipeng, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. "Pre-trained Models for Natural Language Processing: A Survey." arXiv preprint arXiv:2003.08271 (2020).

Transformers

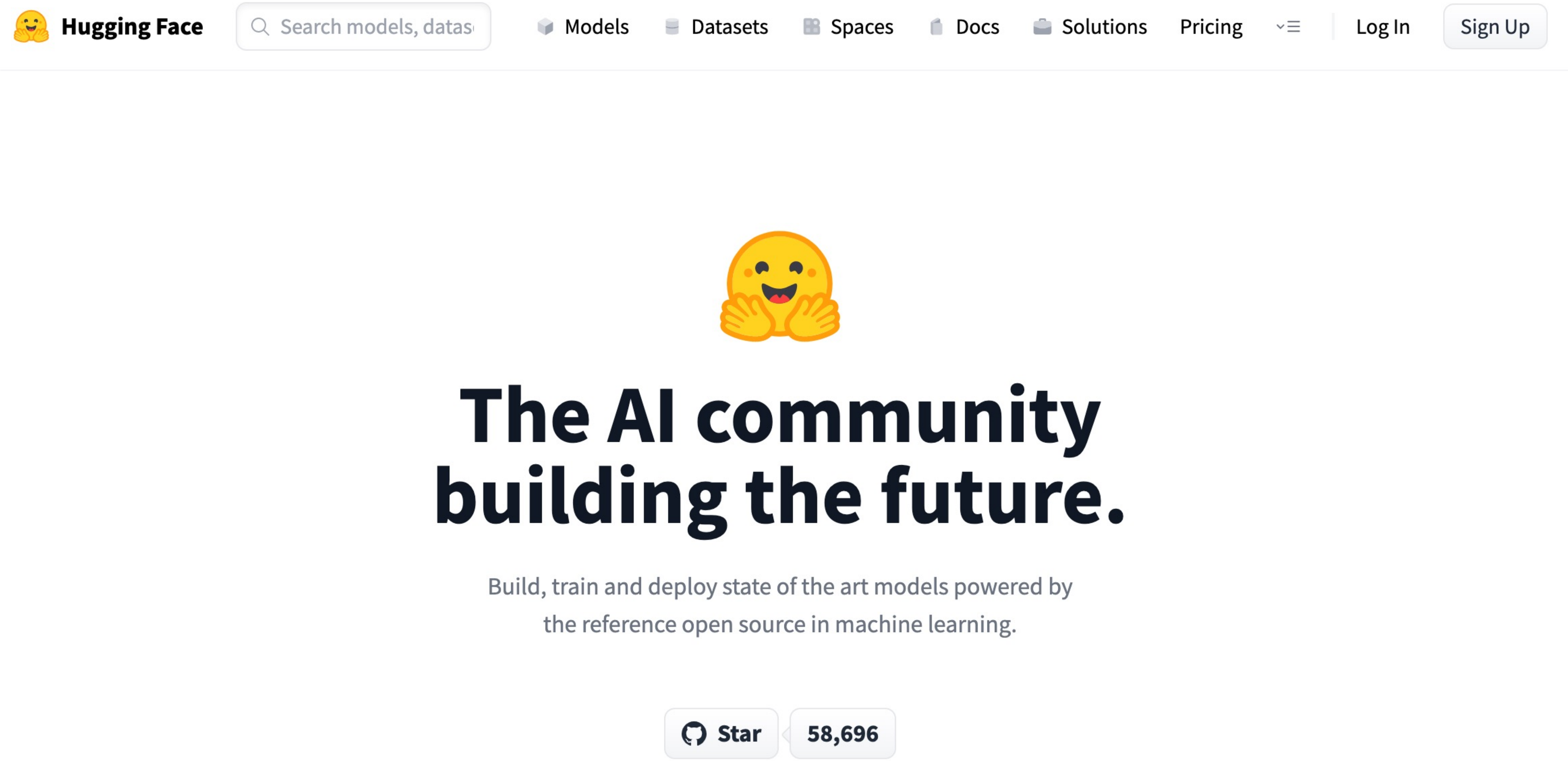
# Transformers

## State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch

- Transformers
  - pytorch-transformers
  - pytorch-pretrained-bert
- provides state-of-the-art general-purpose architectures
  - (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, CTRL...)
  - for Natural Language Understanding (NLU) and
    Natural Language Generation (NLG)
    with over 32+ pretrained models
    in 100+ languages
    and deep interoperability between
    TensorFlow 2.0 and
    PyTorch.

Source: https://github.com/huggingface/transformers

# Hugging Face




https://huggingface.co/

# Hugging Face Transformers

🤗 **Hugging Face** | Search models, datasets, users... | Models | Datasets | Spaces | Docs | Solutions | Pricing | Log In | Sign Up

**Transformers**

Search documentation ⌘K

V4.16.2 | EN | 58,697

GET STARTED

- 🤗 Transformers
- Quick tour
- Installation
- Philosophy
- Glossary

USING 🤗 TRANSFORMERS

- Summary of the tasks
- Summary of the models
- Preprocessing data
- Fine-tuning a pretrained model
- Distributed training with 🤗 Accelerate

## 🤗 Transformers

State-of-the-art Machine Learning for Jax, Pytorch and TensorFlow

🤗 Transformers (formerly known as *pytorch-transformers* and *pytorch-pretrained-bert*) provides thousands of pretrained models to perform tasks on different modalities such as text, vision, and audio.

These models can applied on:

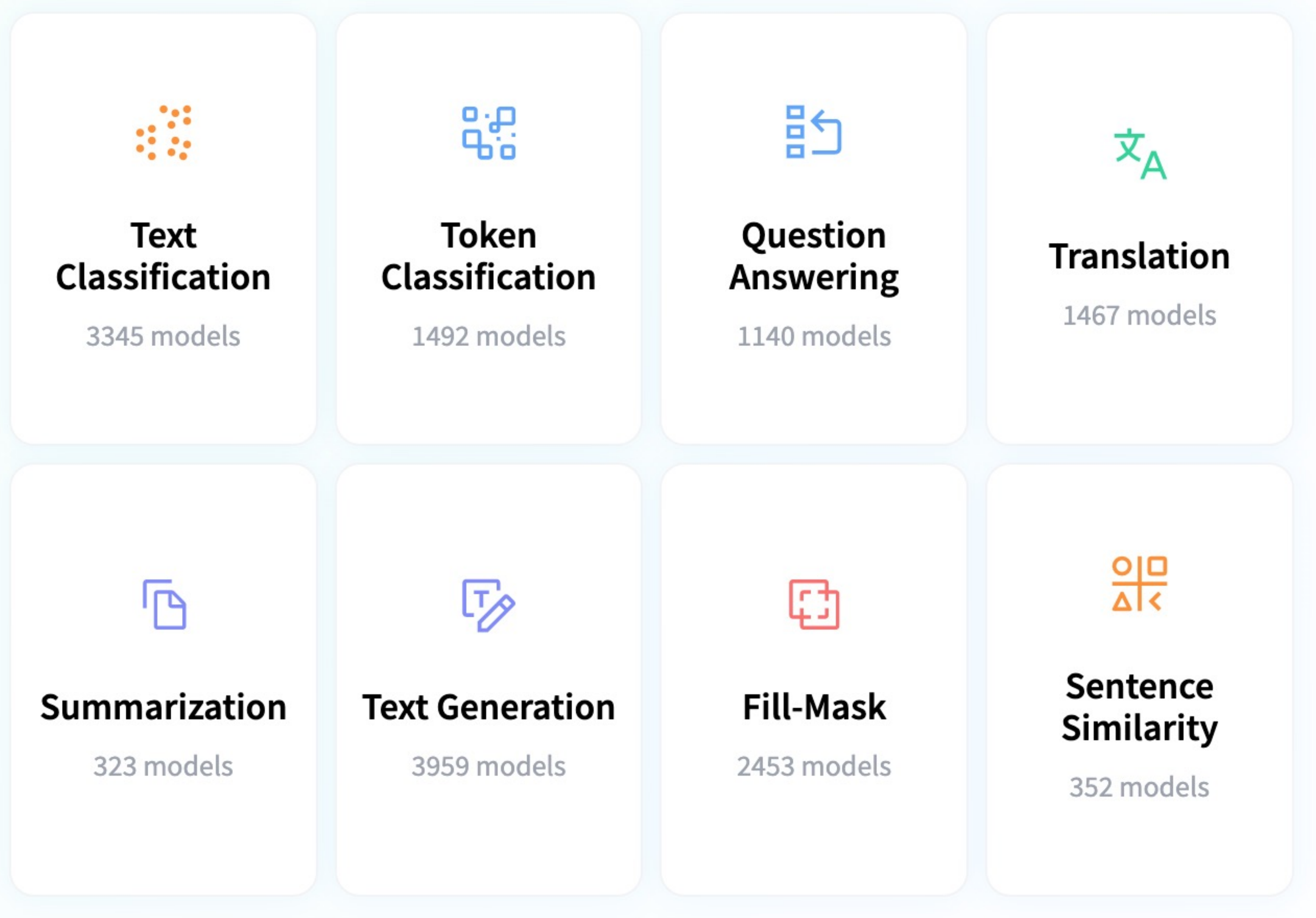
- 📝 Text, for tasks like text classification, information extraction, question answering, summarization, translation, text generation, in over 100 languages.
- 🖼️ Images, for tasks like image classification, object detection, and segmentation.
- 🗣️ Audio, for tasks like speech recognition and audio classification.

Transformer models can also perform tasks on **several modalities combined**, such as table question answering, optical character recognition, information extraction from scanned documents, video classification, and visual question answering.

- 🤗 Transformers
  - If you are looking for custom support from the Hugging Face team
  - Features
  - Contents
    - Supported models
    - Supported frameworks

https://huggingface.co/docs/transformers/index

# Hugging Face Tasks
# Natural Language Processing




https://huggingface.co/tasks

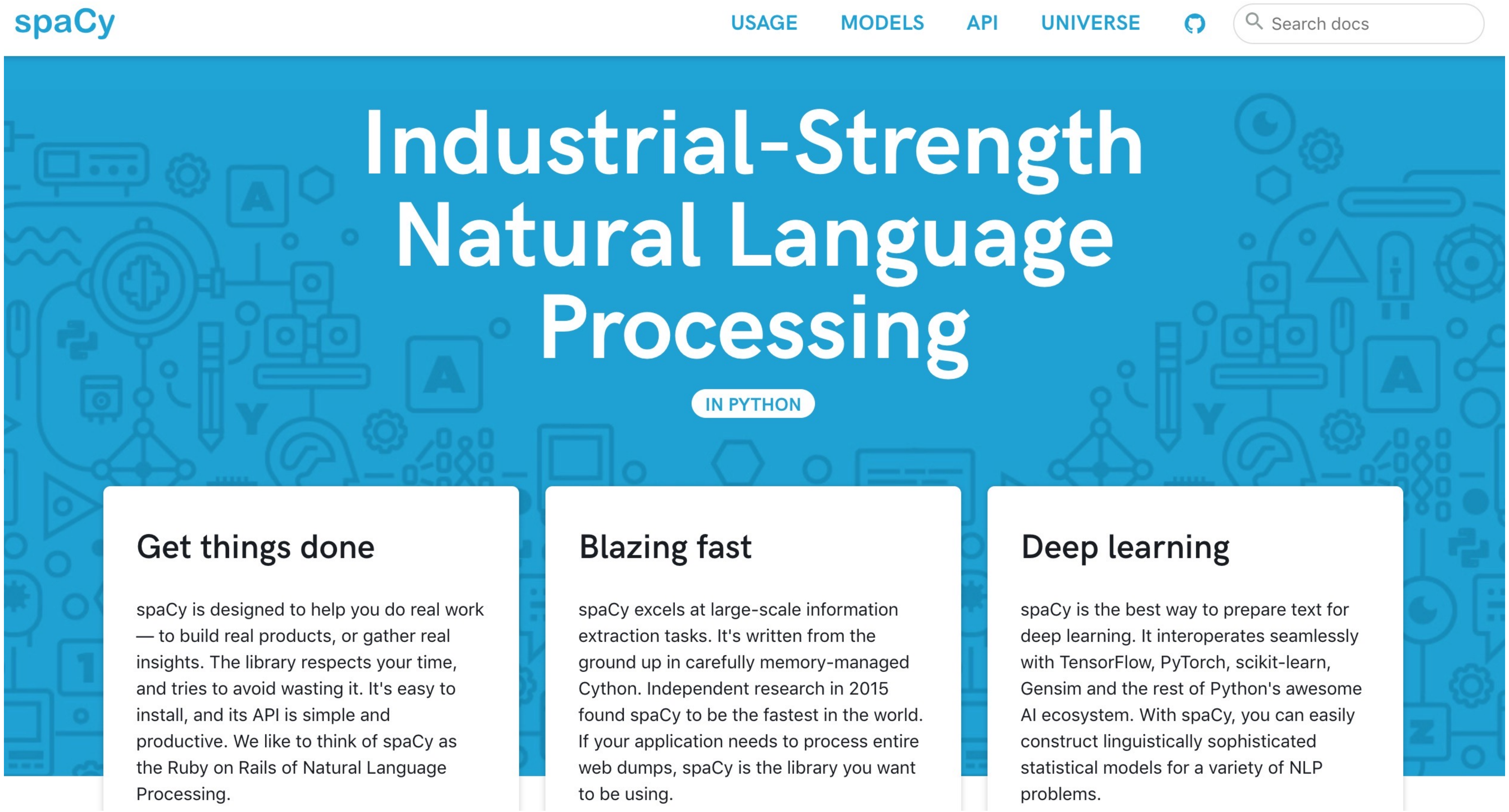
# Text Analytics with Python

# NLP Libraries and Tools

python™

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

# spaCy: Natural Language Processing




https://spacy.io/

# 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.)*


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

Bibliography
Term Index



*Please post any questions about the materials to the nltk-users mailing list. Please report any errors on the issue tracker.*

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

# 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

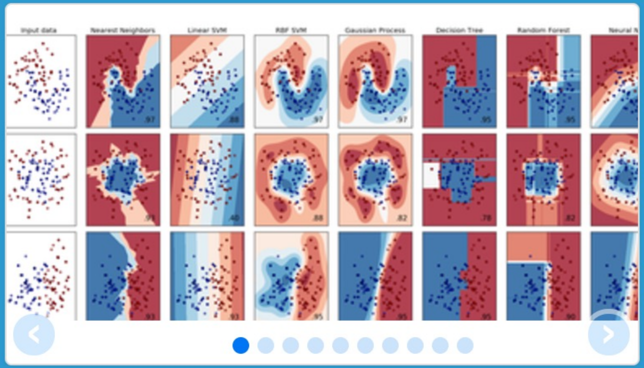
scikit learn

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/

# Python in Google Colab (Python101)

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

python101.ipynb

File Edit View Insert Runtime Tools Help Last edited on May 13

Comment Share A

Table of contents

+ Code + Text

Connect Editing


Text Analytics and Natural Language Processing (NLP)

**Python for Natural Language Processing**

spaCy Chinese Model

Open Chinese Convert (OpenCC, 開放中文轉換)

Jieba 結巴中文分詞

Natural Language Toolkit (NLTK)

Stanza: A Python NLP Library for Many Human Languages

Text Processing and Understanding

NLTK (Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit)

NLP Zero to Hero

Natural Language Processing - Tokenization (NLP Zero to Hero, part 1)

Natural Language Processing - Sequencing - Turning sentence into data (NLP Zero to Hero, part 2)

Natural Language Processing - Training a model to recognize sentiment in text (NLP Zero to Hero, part 3)

Keras preprocessing text

JSON File


```python
text = "Steve Jobs and Steve Wozniak incorporated Apple Computer on January 3, 1977, in Cupertino, California."
doc = nlp(text)
displacy.render(doc, style="ent", jupyter=True)
```

Steve Jobs **PERSON** and Steve Wozniak **PERSON** incorporated Apple Computer **ORG** on January 3, 1977 **DATE** , in Cupertino **GPE** , California **GPE** .

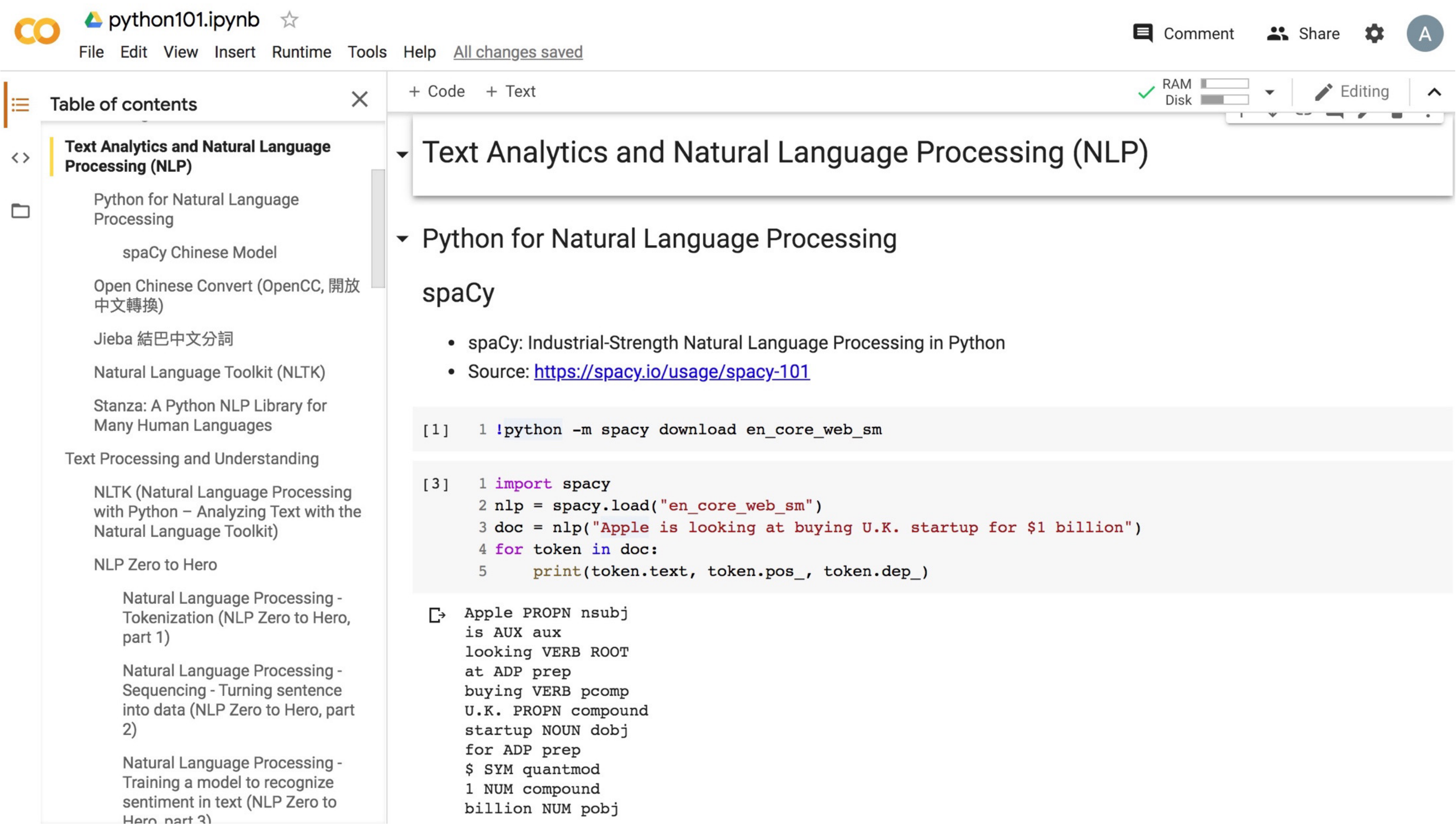
```python
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Stanford University is located in California. It is a great university.")
import pandas as pd
cols = ("text", "lemma", "pos", "tag", "pos_explain", "stopword")
rows = []
for t in doc:
    row = [t.text, t.lemma_, t.pos_, t.tag_,spacy.explain(t.pos_), t.is_stop]
    rows.append(row)
df = pd.DataFrame(rows, columns=cols)
df
```

| | text | lemma | pos | tag | pos_explain | stopword |
|---|---|---|---|---|---|---|
| 0 | Stanford | Stanford | PROPN | NNP | proper noun | False |
| 1 | University | University | PROPN | NNP | proper noun | False |
| 2 | is | be | VERB | VBZ | verb | True |
| 3 | located | locate | VERB | VBN | verb | False |
| 4 | in | in | ADP | IN | adposition | True |
| 5 | California | California | PROPN | NNP | proper noun | False |
| 6 | . | . | PUNCT | . | punctuation | False |
| 7 | It | -PRON- | PRON | PRP | pronoun | True |

https://tinyurl.com/aintpupython101

# Python in Google Colab (Python101)

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




https://tinyurl.com/aintpupython101

# Python in Google Colab (Python101)

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

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")
import pandas as pd
cols = ("text", "lemma", "POS", "explain", "stopword")
rows = []
for t in doc:
    row = [t.text, t.lemma_, t.pos_, spacy.explain(t.pos_), t.is_stop]
    rows.append(row)
df = pd.DataFrame(rows, columns=cols)
df
```

| | text | lemma | POS | explain | stopword |
|---|---|---|---|---|---|
| 0 | Apple | Apple | PROPN | proper noun | False |
| 1 | is | be | VERB | verb | True |
| 2 | looking | look | VERB | verb | False |
| 3 | at | at | ADP | adposition | True |
| 4 | buying | buy | VERB | verb | False |
| 5 | U.K. | U.K. | PROPN | proper noun | False |
| 6 | startup | startup | NOUN | noun | False |
| 7 | for | for | ADP | adposition | True |
| 8 | $ | $ | SYM | symbol | False |
| 9 | 1 | 1 | NUM | numeral | False |
| 10 | billion | billion | NUM | numeral | False |

https://tinyurl.com/aintpupython101

# Python in Google Colab (Python101)

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

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```python
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Stanford University is located in California. It is a great university.")
import pandas as pd
cols = ("text", "lemma", "POS", "explain", "stopword")
rows = []
for t in doc:
    row = [t.text, t.lemma_, t.pos_, spacy.explain(t.pos_), t.is_stop]
    rows.append(row)
df = pd.DataFrame(rows, columns=cols)
df
```

| | text | lemma | POS | explain | stopword |
|---|---|---|---|---|---|
| 0 | Stanford | Stanford | PROPN | proper noun | False |
| 1 | University | University | PROPN | proper noun | False |
| 2 | is | be | VERB | verb | True |
| 3 | located | locate | VERB | verb | False |
| 4 | in | in | ADP | adposition | True |
| 5 | California | California | PROPN | proper noun | False |
| 6 | . | . | PUNCT | punctuation | False |
| 7 | It | -PRON- | PRON | pronoun | True |
| 8 | is | be | VERB | verb | True |
| 9 | a | a | DET | determiner | True |
| 10 | great | great | ADJ | adjective | False |
| 11 | university | university | NOUN | noun | False |
| 12 | . | . | PUNCT | punctuation | False |

https://tinyurl.com/aintpupython101

# Python in Google Colab (Python101)

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

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```python
import spacy
nlp = spacy.load("en_core_web_sm")
text = "Stanford University is located in California. It is a great university."
doc = nlp(text)
for ent in doc.ents:
    print(ent.text, ent.label_)
```

```
Stanford University ORG
California GPE
```

```python
from spacy import displacy
text = "Stanford University is located in California. It is a great university."
doc = nlp(text)
displacy.render(doc, style="ent", jupyter=True)
```

Stanford University **ORG** is located in California **GPE** . It is a great university.

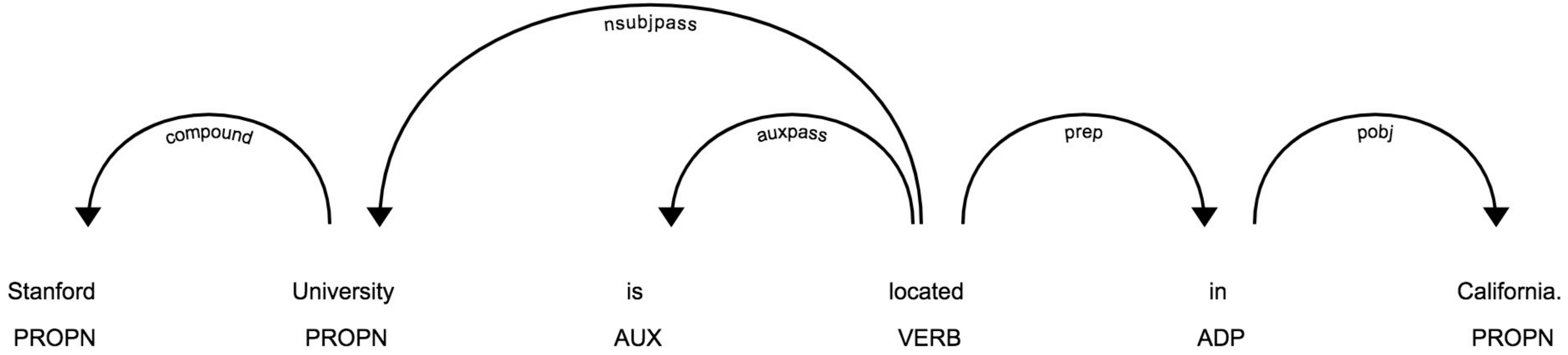
https://tinyurl.com/aintpupython101

# Python in Google Colab (Python101)

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

```python
from spacy import displacy
text = "Stanford University is located in California. It is a great university."
doc = nlp(text)
displacy.render(doc, style="ent", jupyter=True)
displacy.render(doc, style="dep", jupyter=True)
```

Stanford University **ORG** is located in California **GPE** . It is a great university.




https://tinyurl.com/aintpupython101

# Python in Google Colab (Python101)

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

python101.ipynb

File Edit View Insert Runtime Tools Help Last edited on May 13

Comment Share

Table of contents


Text Analytics and Natural Language Processing (NLP)
**Python for Natural Language Processing**
spaCy Chinese Model
Open Chinese Convert (OpenCC, 開放中文轉換)
Jieba 結巴中文分詞
Natural Language Toolkit (NLTK)
Stanza: A Python NLP Library for Many Human Languages
Text Processing and Understanding
NLTK (Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit)
NLP Zero to Hero
Natural Language Processing - Tokenization (NLP Zero to Hero, part 1)
Natural Language Processing - Sequencing - Turning sentence into data (NLP Zero to Hero, part 2)
Natural Language Processing - Training a model to recognize sentiment in text (NLP Zero to Hero, part 3)
Keras preprocessing text
JSON File


+ Code + Text Connect Editing

```python
text = "Steve Jobs and Steve Wozniak incorporated Apple Computer on January 3, 1977, in Cupertino, California."
doc = nlp(text)
displacy.render(doc, style="ent", jupyter=True)
```

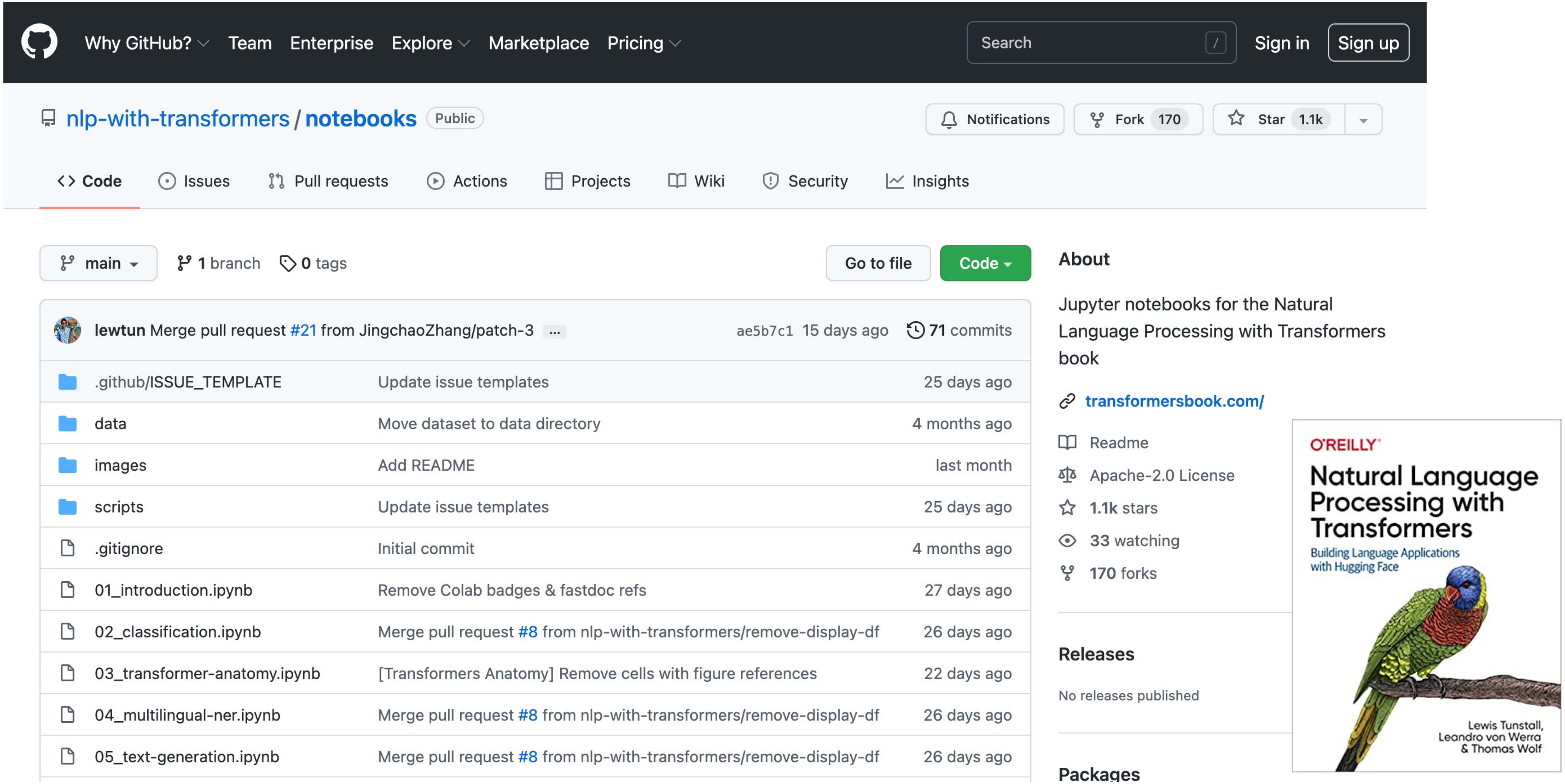
Steve Jobs PERSON and Steve Wozniak PERSON incorporated Apple Computer ORG on January 3, 1977 DATE , in Cupertino GPE , California GPE .

```python
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Stanford University is located in California. It is a great university.")
import pandas as pd
cols = ("text", "lemma", "pos", "tag", "pos_explain", "stopword")
rows = []
for t in doc:
    row = [t.text, t.lemma_, t.pos_, t.tag_,spacy.explain(t.pos_), t.is_stop]
    rows.append(row)
df = pd.DataFrame(rows, columns=cols)
df
```

| | text | lemma | pos | tag | pos_explain | stopword |
|---|---|---|---|---|---|---|
| **0** | Stanford | Stanford | PROPN | NNP | proper noun | False |
| **1** | University | University | PROPN | NNP | proper noun | False |
| **2** | is | be | VERB | VBZ | verb | True |
| **3** | located | locate | VERB | VBN | verb | False |
| **4** | in | in | ADP | IN | adposition | True |
| **5** | California | California | PROPN | NNP | proper noun | False |
| **6** | . | . | PUNCT | . | punctuation | False |
| **7** | It | -PRON- | PRON | PRP | pronoun | True |

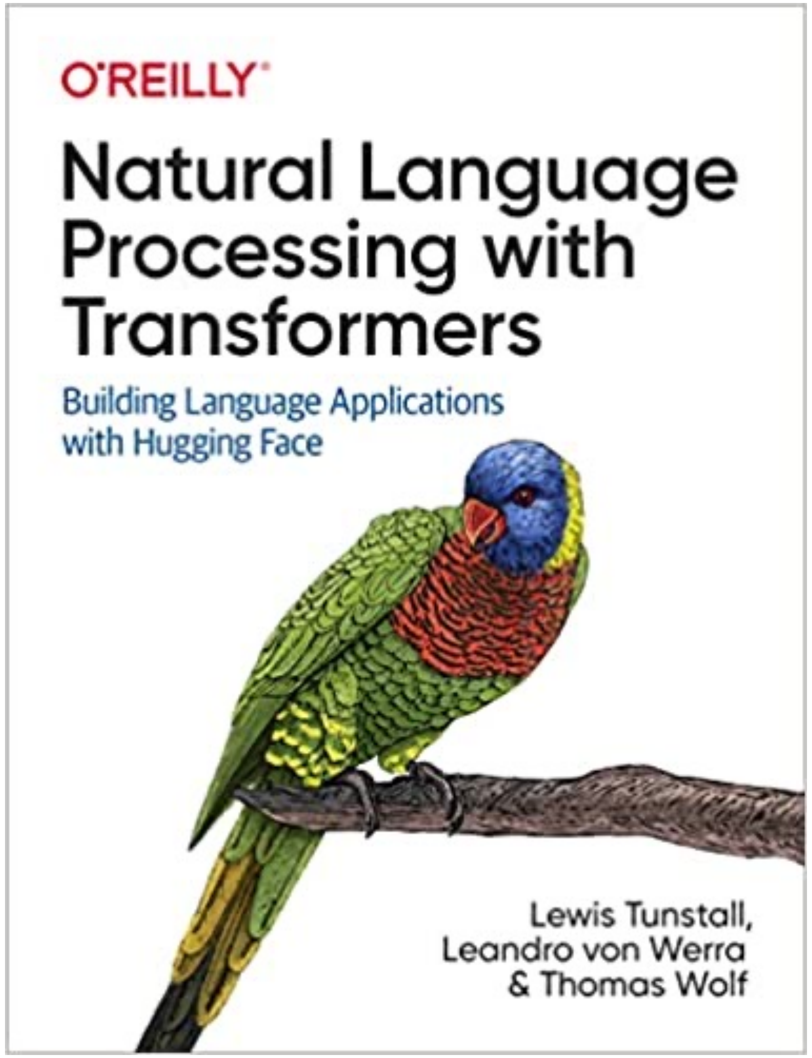
https://tinyurl.com/aintpupython101

# NLP with Transformers Github




https://github.com/nlp-with-transformers/notebooks

# NLP with Transformers Github Notebooks




### Running on a cloud platform

To run these notebooks on a cloud platform, just click on one of the badges in the table below:

| Chapter | Colab | Kaggle | Gradient | Studio Lab |
|---|---|---|---|---|
| Introduction | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Text Classification | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Transformer Anatomy | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Multilingual Named Entity Recognition | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Text Generation | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Summarization | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Question Answering | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Making Transformers Efficient in Production | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Dealing with Few to No Labels | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Training Transformers from Scratch | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |
| Future Directions | Open in Colab | Open in Kaggle | Run on Gradient | Open Studio Lab |

Nowadays, the GPUs on Colab tend to be K80s (which have limited memory), so we recommend using Kaggle, Gradient, or SageMaker Studio Lab. These platforms tend to provide more performant GPUs like P100s, all for free!

https://github.com/nlp-with-transformers/notebooks

# NLP Benchmark Datasets

| Task | Dataset | Link |
|---|---|---|
| Machine Translation | WMT 2014 EN-DE<br>WMT 2014 EN-FR | http://www-lium.univ-lemans.fr/~schwenk/cslm_joint_paper/ |
| Text Summarization | CNN/DM<br>Newsroom<br>DUC<br>Gigaword | https://cs.nyu.edu/~kcho/DMQA/<br>https://summari.es/<br>https://www-nlpir.nist.gov/projects/duc/data.html<br>https://catalog.ldc.upenn.edu/LDC2012T21 |
| Reading Comprehension<br>Question Answering<br>Question Generation | ARC<br>CliCR<br>CNN/DM<br>NewsQA<br>RACE<br>SQuAD<br>Story Cloze Test<br>NarativeQA<br>Quasar<br>SearchQA | http://data.allenai.org/arc/<br>http://aclweb.org/anthology/N18-1140<br>https://cs.nyu.edu/~kcho/DMQA/<br>https://datasets.maluuba.com/NewsQA<br>http://www.qizhexie.com/data/RACE_leaderboard<br>https://rajpurkar.github.io/SQuAD-explorer/<br>http://aclweb.org/anthology/W17-0906.pdf<br>https://github.com/deepmind/narrativeqa<br>https://github.com/bdhingra/quasar<br>https://github.com/nyu-dl/SearchQA |
| Semantic Parsing | AMR parsing<br>ATIS (SQL Parsing)<br>WikiSQL (SQL Parsing) | https://amr.isi.edu/index.html<br>https://github.com/jkkummerfeld/text2sql-data/tree/master/data<br>https://github.com/salesforce/WikiSQL |
| Sentiment Analysis | IMDB Reviews<br>SST<br>Yelp Reviews<br>Subjectivity Dataset | http://ai.stanford.edu/~amaas/data/sentiment/<br>https://nlp.stanford.edu/sentiment/index.html<br>https://www.yelp.com/dataset/challenge<br>http://www.cs.cornell.edu/people/pabo/movie-review-data/ |
| Text Classification | AG News<br>DBpedia<br>TREC<br>20 NewsGroup | http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html<br>https://wiki.dbpedia.org/Datasets<br>https://trec.nist.gov/data.html<br>http://qwone.com/~jason/20Newsgroups/ |
| Natural Language Inference | SNLI Corpus<br>MultiNLI<br>SciTail | https://nlp.stanford.edu/projects/snli/<br>https://www.nyu.edu/projects/bowman/multinli/<br>http://data.allenai.org/scitail/ |
| Semantic Role Labeling | Proposition Bank<br>OneNotes | http://propbank.github.io/<br>https://catalog.ldc.upenn.edu/LDC2013T19 |

Source: Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavvaf, and Edward A. Fox (2020). "Natural Language Processing Advancements By Deep Learning: A Survey." arXiv preprint arXiv:2003.01200.

# Summary

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

# References


- Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Denis Rothman (2021), Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more, Packt Publishing.
- Savaş Yıldırım and Meysam Asgari-Chenaghlu (2021), Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques, Packt Publishing.
- Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.
- Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson.
- Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress.
- Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018), Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning, O'Reilly.
- Charu C. Aggarwal (2018), Machine Learning for Text, Springer.
- Gabe Ignatow and Rada F. Mihalcea (2017), An Introduction to Text Mining: Research Design, Data Collection, and Analysis, SAGE Publications.
- Rajesh Arumugam (2018), Hands-On Natural Language Processing with Python: A practical guide to applying deep learning architectures to your NLP applications, Packt.
- Jake VanderPlas (2016), Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
- 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.
- 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/
- 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/
- Bing Liu (2009), Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer.
- The Super Duper NLP Repo, https://notebooks.quantumstat.com/
- Min-Yuh Day (2022), Python 101, https://tinyurl.com/aintpupython101