

文字探勘 (Text Mining)

文本摘要和主題模型

(Text Summarization and Topic Models)

1082TM07

MBA, BDABI, TKU (E3611) (8480) (Spring 2020)

Mon, 7, 8, 9 (14:10-17:00) (B206)



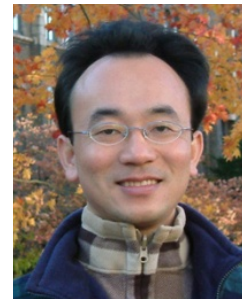
Chichang Jou

周清江

Associate Professor

副教授

cjou@mail.tku.edu.tw



Min-Yuh Day

戴敏育

Associate Professor

副教授

myday@mail.tku.edu.tw

Dept. of Information Management, Tamkang University

淡江大學 資訊管理學系

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2020/03/02	文字探勘課程介紹 (Course Orientation on Text Mining)
2	2020/03/09	文字探勘基礎：自然語言處理 (Foundations of Text Mining: Natural Language Processing; NLP)
3	2020/03/16	Python自然語言處理 (Python for Natural Language Processing)
4	2020/03/23	處理和理解文本 (Processing and Understanding Text)
5	2020/03/30	文本表達特徵工程 (Feature Engineering for Text Representation)
6	2020/04/06	人工智慧文本分析個案研究 I (Case Study on Artificial Intelligence for Text Analytics I)

課程大綱 (Syllabus)

- | 週次 (Week) | 日期 (Date) | 內容 (Subject/Topics) |
|-----------|------------|--|
| 7 | 2020/04/13 | 文本分類
(Text Classification) |
| 8 | 2020/04/20 | 文本摘要和主題模型
(Text Summarization and Topic Models) |
| 9 | 2020/04/27 | 期中報告 (Midterm Project Report) |
| 10 | 2020/05/04 | 文本相似度和分群
(Text Similarity and Clustering) |
| 11 | 2020/05/11 | 語意分析和命名實體識別
(Semantic Analysis and Named Entity Recognition; NER) |
| 12 | 2020/05/18 | 情感分析
(Sentiment Analysis) |

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
13	2020/05/25	人工智慧文本分析個案研究 II (Case Study on Artificial Intelligence for Text Analytics II)
14	2020/06/01	深度學習和通用句子嵌入模型 (Deep Learning and Universal Sentence-Embedding Models)
15	2020/06/08	問答系統與對話系統 (Question Answering and Dialogue Systems)
16	2020/06/15	期末報告 I (Final Project Presentation I)
17	2020/06/22	期末報告 II (Final Project Presentation II)
18	2020/06/29	教師彈性補充教學

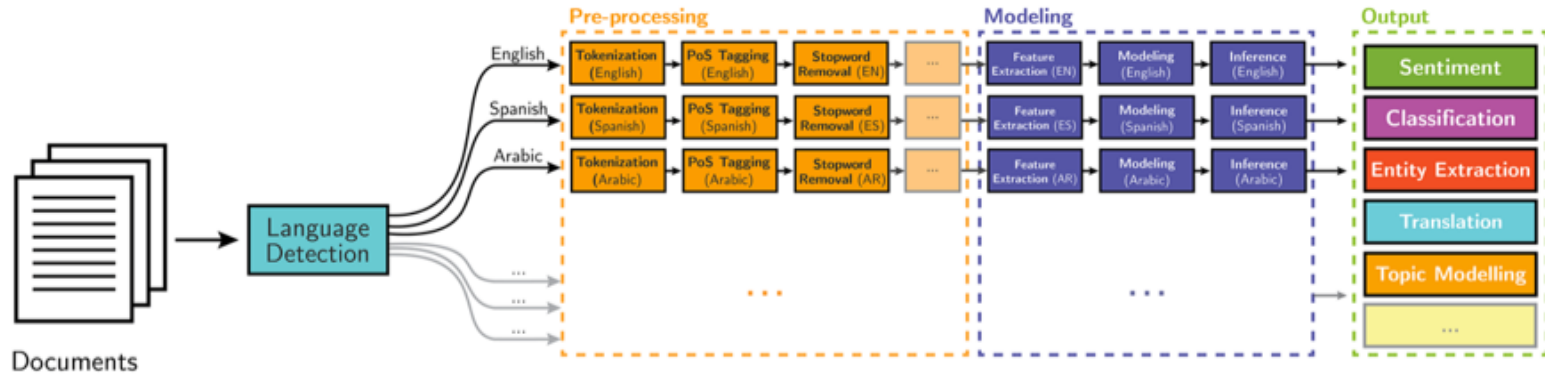
Outline

- Text Summarization
- Topic Models
 - Topic Modeling
 - Latent Dirichlet Allocation (LDA)

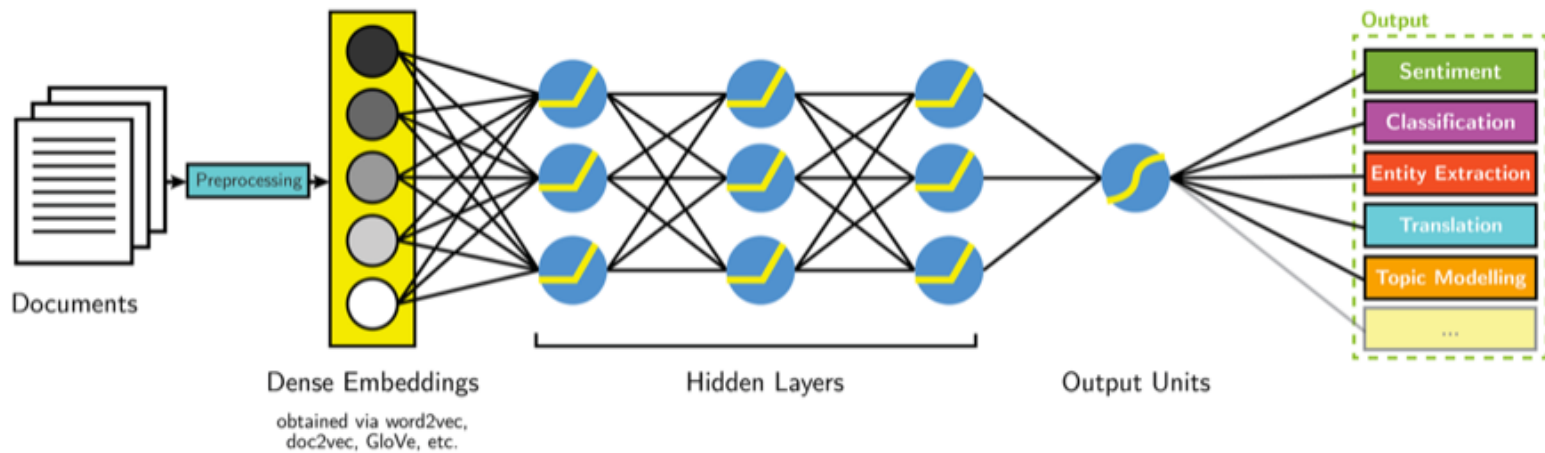
**Text
Summarization
and
Topic Models**

NLP

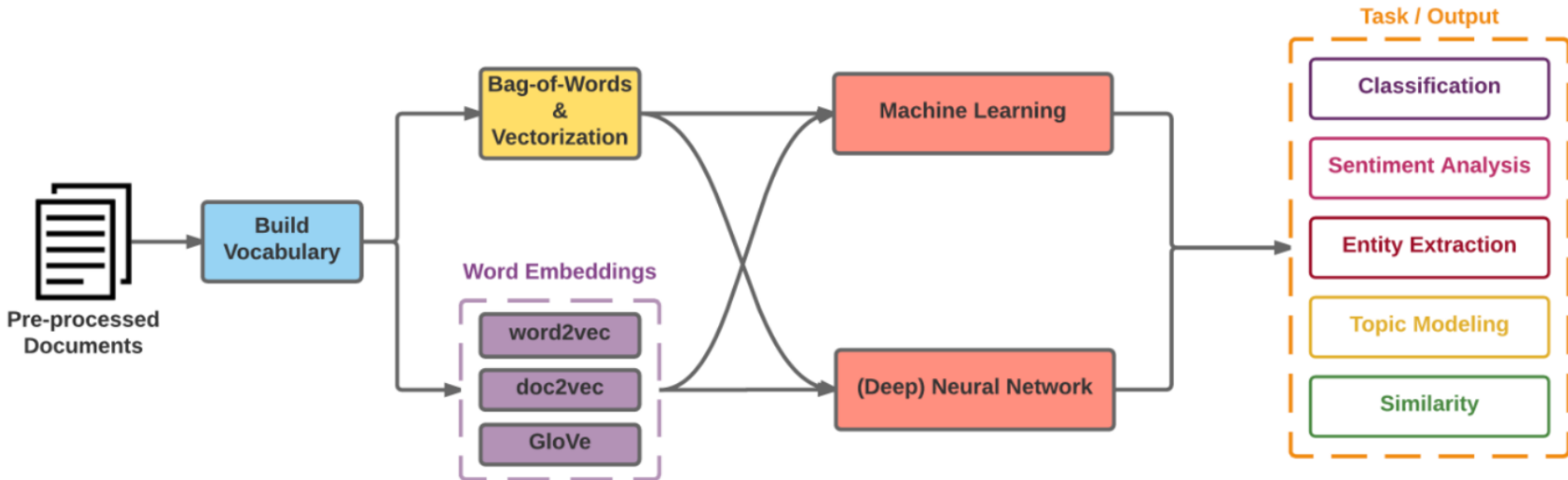
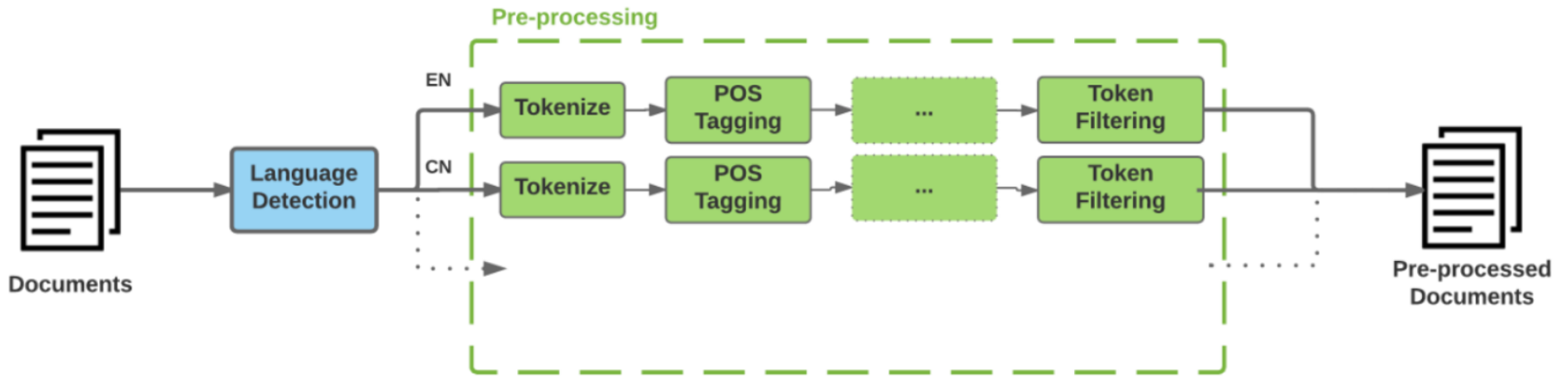
Classical NLP



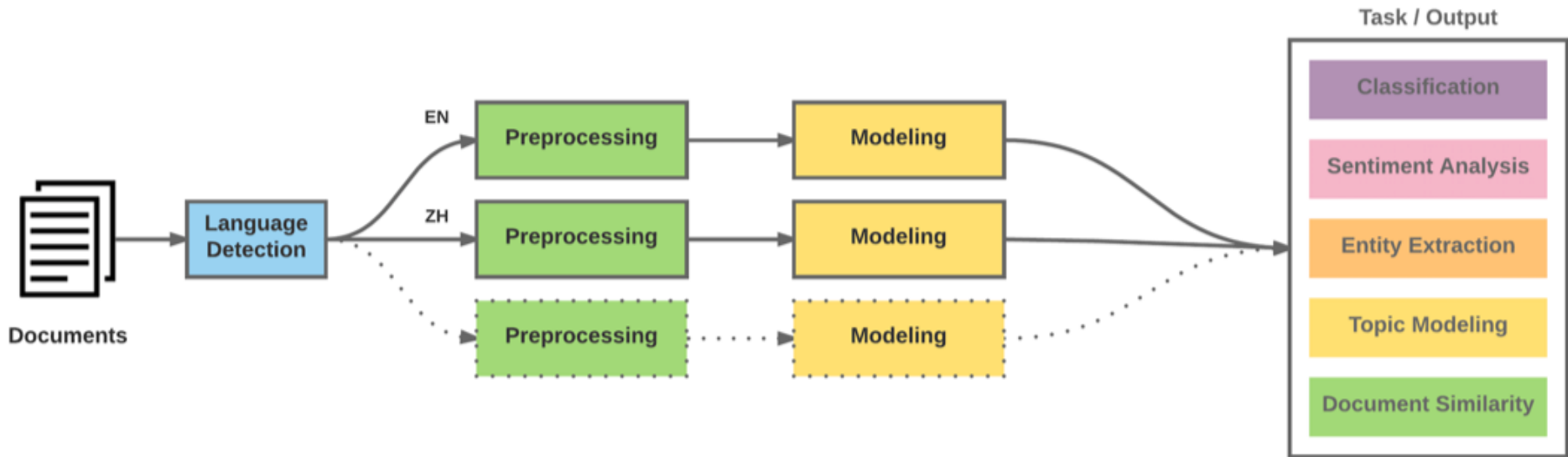
Deep Learning-based NLP



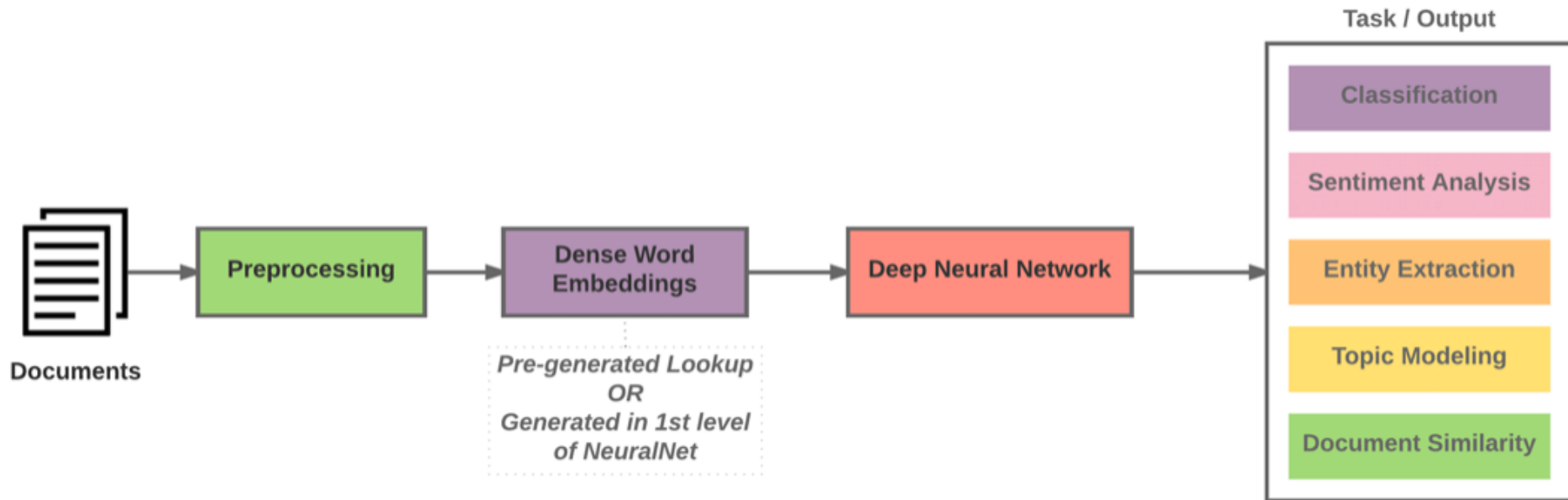
Modern NLP Pipeline



Modern NLP Pipeline



Deep Learning NLP



Natural Language Processing (NLP) and Text Mining

Raw text

Sentence Segmentation

Tokenization

Part-of-Speech (POS)

Stop word removal

Stemming / Lemmatization

Dependency Parser

String Metrics & Matching

word's stem

am → am

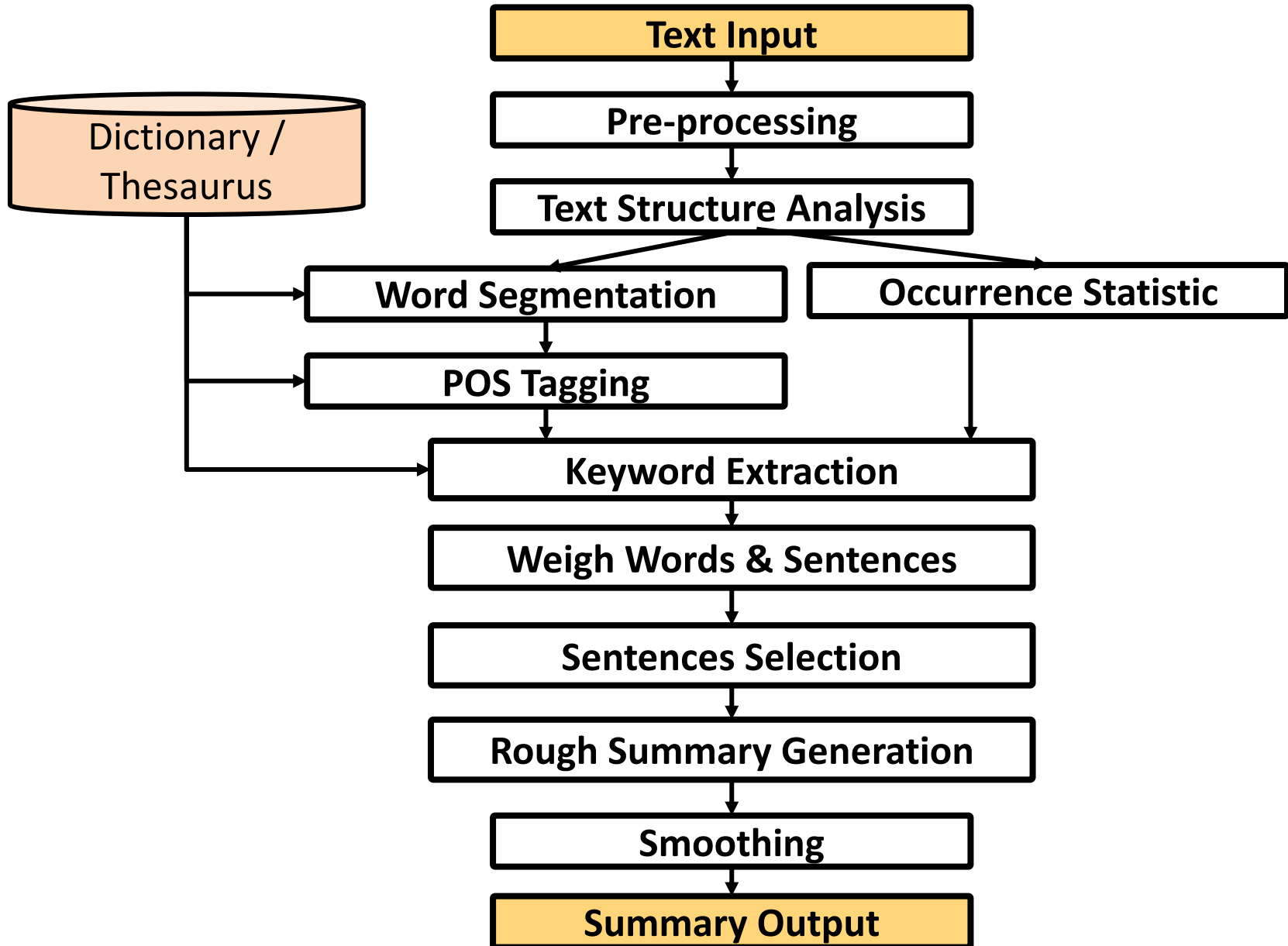
having → hav

word's lemma

am → be

having → have

Text Summarization



Topic Modeling

Topics

gene 0.04
dna 0.02
genetic 0.01
...

life 0.02
evolve 0.01
organism 0.01
...

brain 0.04
neuron 0.02
nerve 0.01
...

data 0.02
number 0.02
computer 0.01
...

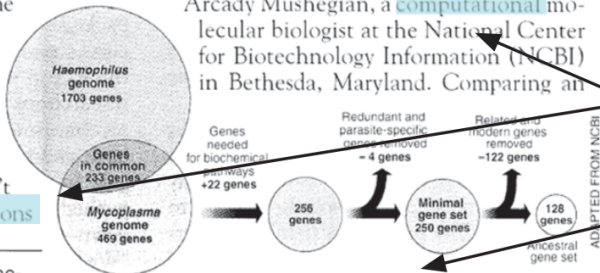
Documents

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

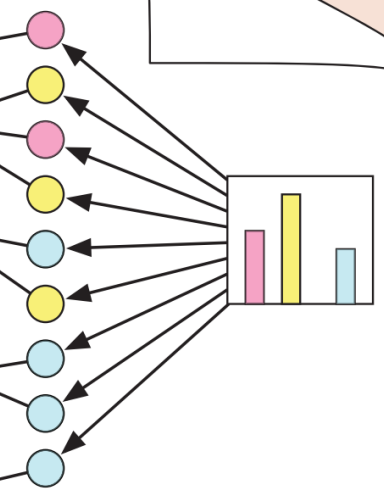


* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

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

SCIENCE • VOL. 272 • 24 MAY 1996

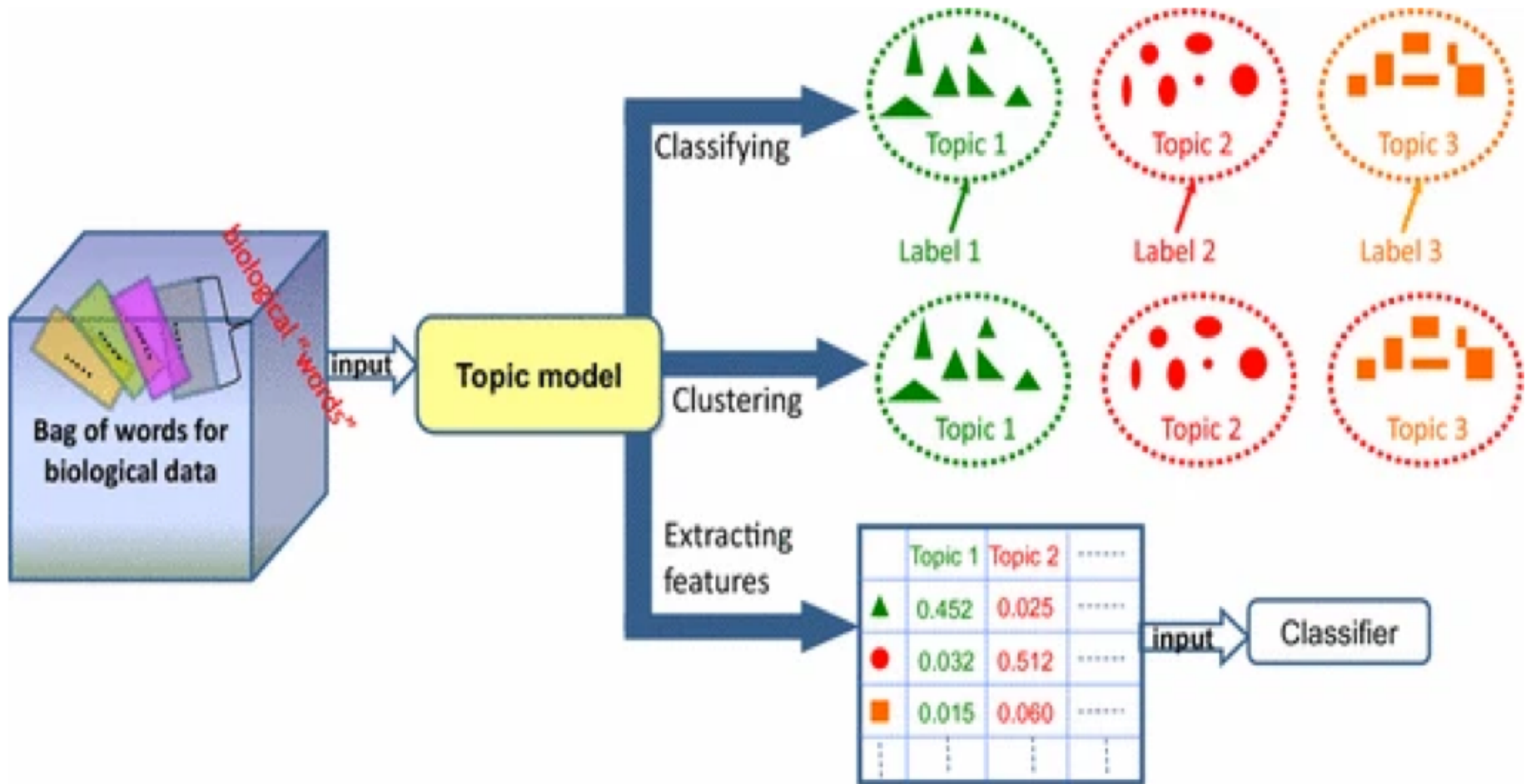
Topic proportions and assignments



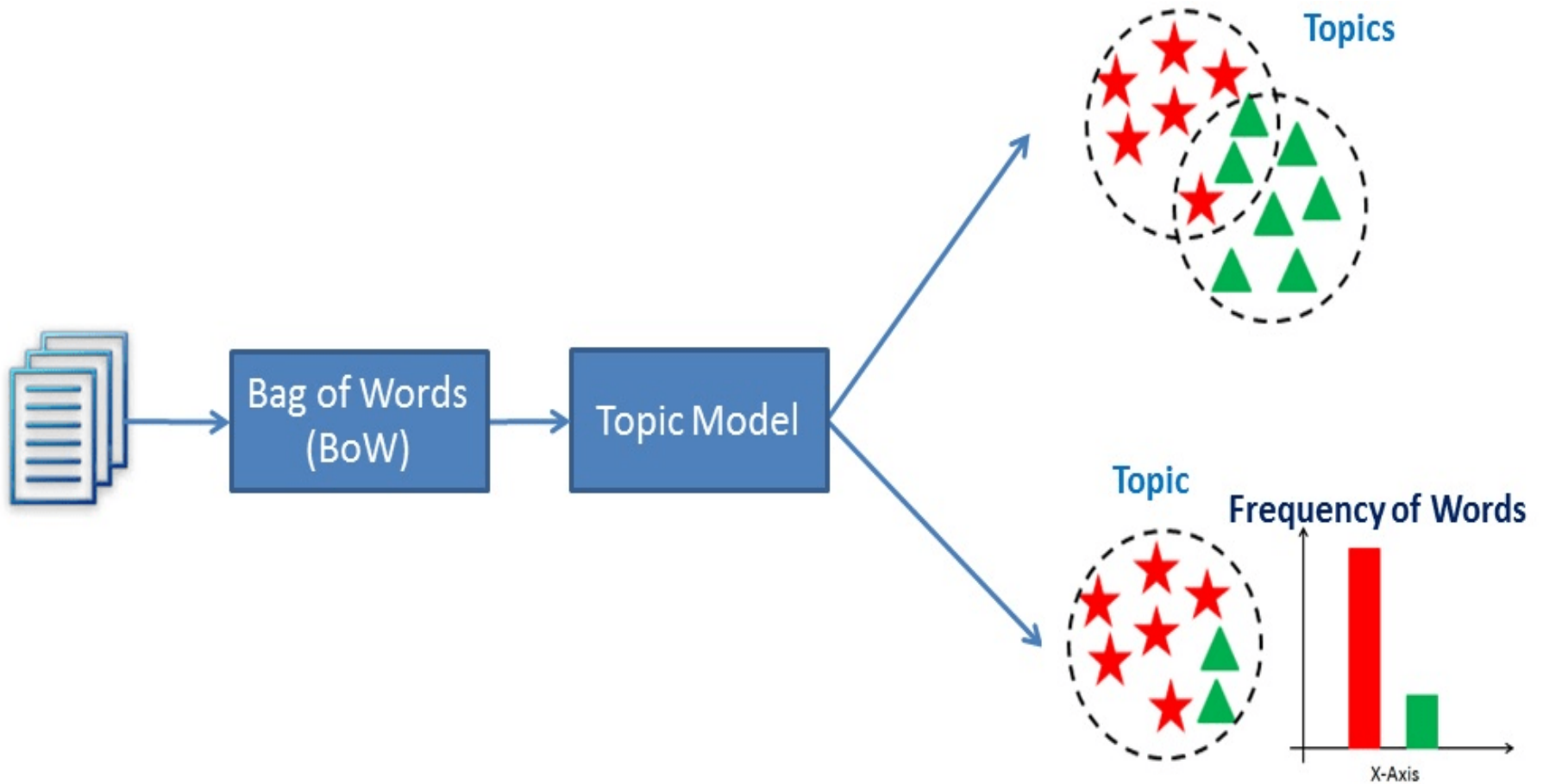
Text Summarization and Information Extraction

- Key-phrase extraction
 - extracting key influential phrases from the documents.
- Topic modeling
 - Extract various diverse concepts or topics present in the documents, retaining the major themes.
- Document summarization
 - Summarize entire text documents to provide a gist that retains the important parts of the whole corpus.

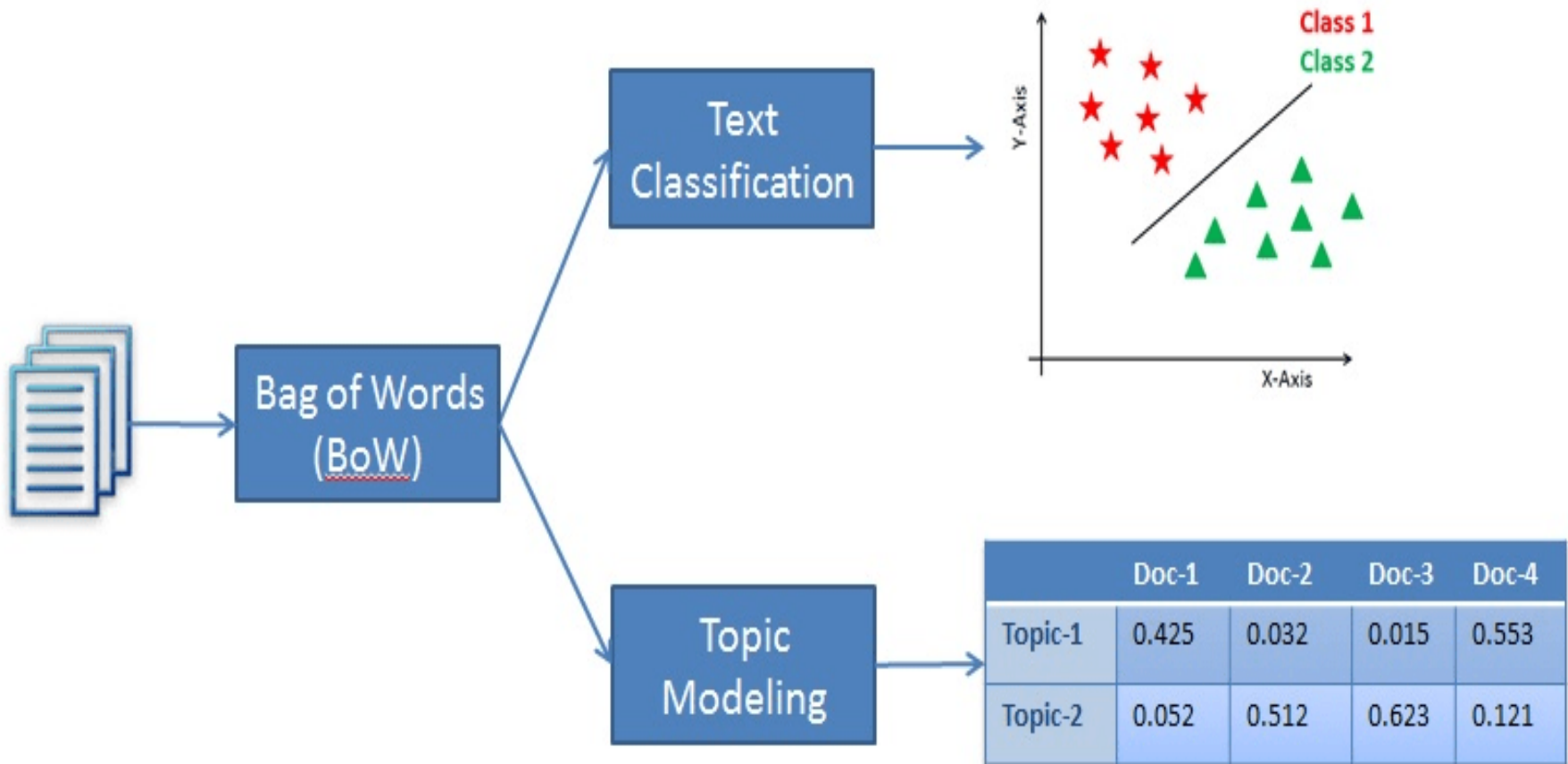
Topic Model in Bioinformatics



Topic Modeling



Topic Modeling (Unsupervised Learning) VS. Text Classification (Supervised Learning)



Topic Modeling

Term Document Matrix to Topic Distribution

Term Document Matrix

	Doc-1	Doc-2	Doc-3	Doc-4
Term-1				
Term-2				
Term-3				
Term-4				

$m \times m$ Matrix

Word Assignment to Topics

	Topic-1	Topic-2
Term-1		
Term-2		
Term-3		
Term-4		

$m \times n$ Singular Matrix

Topic Importance

	Topic-1	Topic-2
Topic-1		
Topic-2		

$n \times n$ Diagonal Matrix

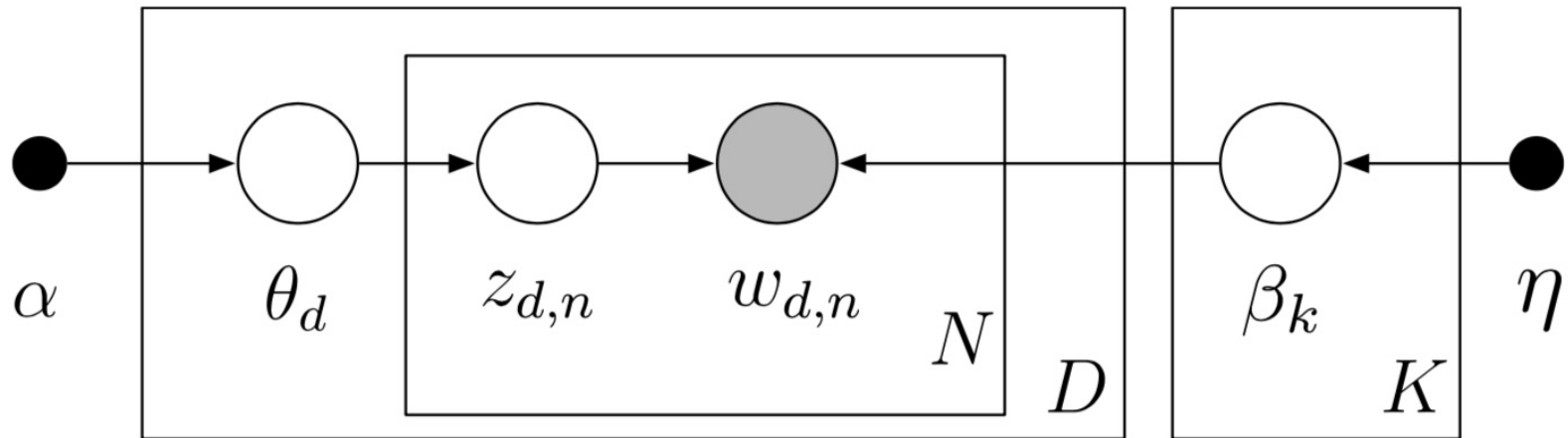
Topic Distribution Across Documents

	Doc-1	Doc-2	Doc-3	Doc-4
Topic-1				
Topic-2				

$n \times m$ Singular Matrix

Topic Modeling

Latent Dirichlet Allocation (LDA)



D documents

N words

K topics

Latent Dirichlet Allocation (Blei et al., 2003)

Latent Dirichlet Allocation

David M. Blei

*Computer Science Division
University of California
Berkeley, CA 94720, USA*

BLEI@CS.BERKELEY.EDU

Andrew Y. Ng

*Computer Science Department
Stanford University
Stanford, CA 94305, USA*

ANG@CS.STANFORD.EDU

Michael I. Jordan

*Computer Science Division and Department of Statistics
University of California
Berkeley, CA 94720, USA*

JORDAN@CS.BERKELEY.EDU

Editor: John Lafferty

Abstract

We describe *latent Dirichlet allocation* (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document. We present efficient approximate inference techniques based on variational methods and an EM algorithm for empirical Bayes parameter estimation. We report results in document modeling, text classification, and collaborative filtering, comparing to a mixture of unigrams model and the probabilistic LSI model.

Python in Google Colab (Python101)

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

The screenshot shows the Google Colab interface for a notebook named 'python101.ipynb'. The top navigation bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help', with a status 'All changes saved'. On the right, there are icons for 'Comment', 'Share', 'Settings', and a user profile 'A'. Below the navigation bar, the interface is split into a left sidebar and a main workspace.

The left sidebar contains a 'Table of contents' panel with the following items:

- Build the model
- Train the model
- Evaluate the model
- Create a graph of accuracy and loss over time
- Text Classification: BBC News Articles
- Text Summarization and Topic Modeling
 - Text Summarization
 - Text Summarization with Gensim Summarization**
 - Topic Modeling
 - Topic Modeling with Gensim LSI model
 - Topic Modeling with Gensim LDA model
 - Topic Modeling with Scikit-learn LDA and NMF
 - Topic Modeling Visualization

The main workspace shows a code cell titled 'Text Summarization with Gensim Summarization'. The code in the cell is:

```
1 from pprint import pprint as print
2 from gensim.summarization import summarize

[ ] 1 text = (
2     "Thomas A. Anderson is a man living two lives. By day he is an "
3     "average computer programmer and by night a hacker known as "
4     "Neo. Neo has always questioned his reality, but the truth is "
5     "far beyond his imagination. Neo finds himself targeted by the "
6     "police when he is contacted by Morpheus, a legendary computer "
7     "hacker branded a terrorist by the government. Morpheus awakens "
8     "Neo to the real world, a ravaged wasteland where most of "
9     "humanity have been captured by a race of machines that live "
10    "off of the humans' body heat and electrochemical energy and "
11    "who imprison their minds within an artificial reality known as "
12    "the Matrix. As a rebel against the machines, Neo must return to "
13    "the Matrix and confront the agents: super-powerful computer "
14    "programs devoted to snuffing out Neo and the entire human "
15    "rebellion. "
16 )
17 print(text)
```

Below the code cell, the output is displayed as a string:

```
('Thomas A. Anderson is a man living two lives. By day he is an average '
'computer programmer and by night a hacker known as Neo. Neo has always '
'questioned his reality, but the truth is far beyond his imagination. Neo '
```

<https://tinyurl.com/imtkupython101>

Python in Google Colab (Python101)

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

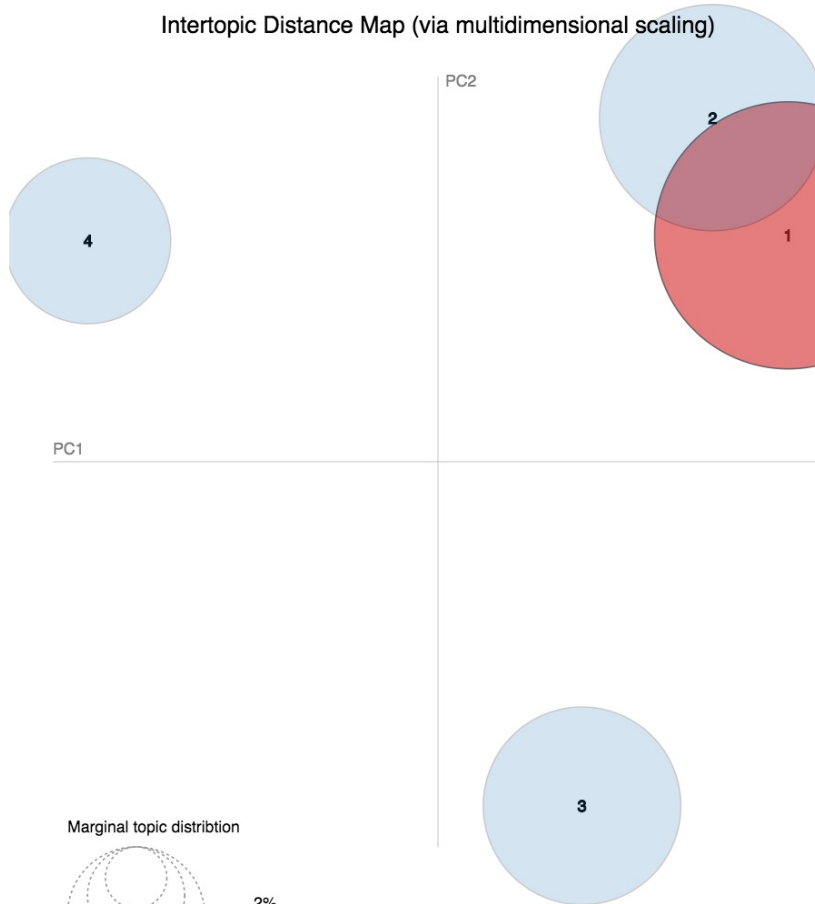
Selected Topic: Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

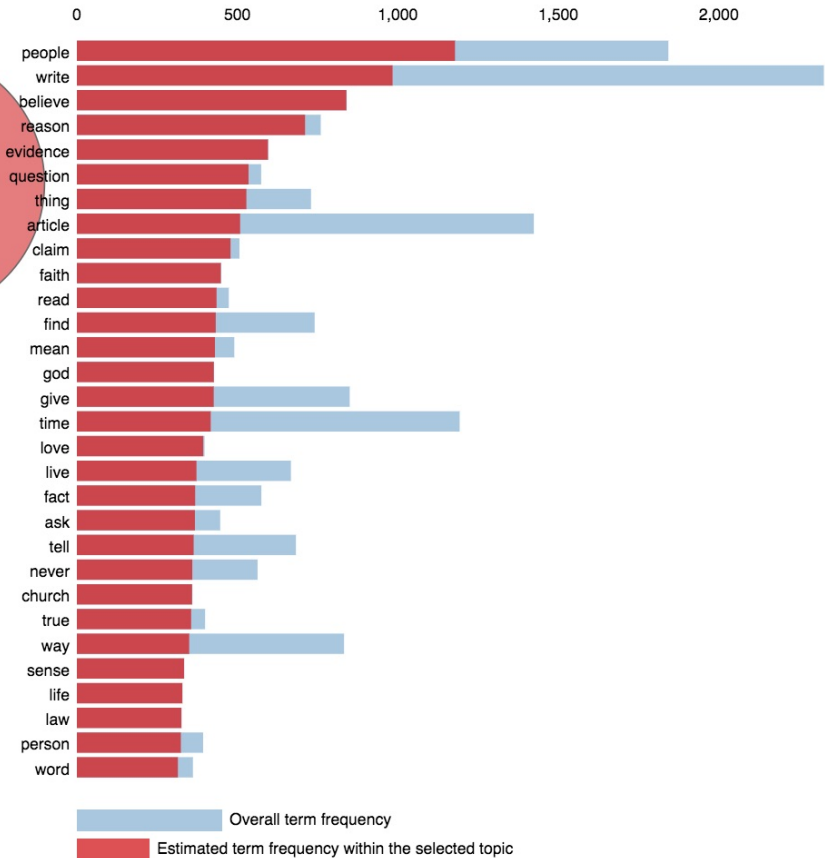
$\lambda = 1$

0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)



Top-30 Most Relevant Terms for Topic 1 (37.7% of tokens)



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = $\lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

<https://tinyurl.com/imtkupython101>

Summary

- Text Summarization
- Topic Models
 - Topic Modeling
 - Latent Dirichlet Allocation (LDA)

References

- Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress. <https://github.com/Apress/text-analytics-w-python-2e>
- Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018), Applied Text Analysis with Python, O'Reilly Media. <https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/>
- Selva Prabhakaran (2020), Topic modeling visualization – How to present the results of LDA models?, <https://www.machinelearningplus.com/nlp/topic-modeling-visualization-how-to-present-results-lda-models/>
- Min-Yuh Day (2020), Python 101, <https://tinyurl.com/imtkupython101>