Foundations of Text Analytics: Natural Language Processing (NLP)

Min-Yuh Day, Ph.D, Associate Professor
Institute of Information Management, National Taipei University

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

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

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

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2022/02/22</td>
<td>Introduction to Artificial Intelligence for Text Analytics</td>
</tr>
<tr>
<td>2</td>
<td>2022/03/01</td>
<td>Foundations of Text Analytics: Natural Language Processing (NLP)</td>
</tr>
<tr>
<td>3</td>
<td>2022/03/08</td>
<td>Python for Natural Language Processing</td>
</tr>
<tr>
<td>4</td>
<td>2022/03/15</td>
<td>Natural Language Processing with Transformers</td>
</tr>
<tr>
<td>5</td>
<td>2022/03/22</td>
<td>Case Study on Artificial Intelligence for Text Analytics I</td>
</tr>
<tr>
<td>6</td>
<td>2022/03/29</td>
<td>Text Classification and Sentiment Analysis</td>
</tr>
</tbody>
</table>
# Syllabus

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>2022/04/05</td>
<td>Tomb-Sweeping Day (Holiday, No Classes)</td>
</tr>
<tr>
<td>8</td>
<td>2022/04/12</td>
<td>Midterm Project Report</td>
</tr>
<tr>
<td>9</td>
<td>2022/04/19</td>
<td>Multilingual Named Entity Recognition (NER), Text Similarity and Clustering</td>
</tr>
<tr>
<td>10</td>
<td>2022/04/26</td>
<td>Text Summarization and Topic Models</td>
</tr>
<tr>
<td>11</td>
<td>2022/05/03</td>
<td>Text Generation</td>
</tr>
<tr>
<td>12</td>
<td>2022/05/10</td>
<td>Case Study on Artificial Intelligence for Text Analytics II</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Subject/Topics</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>13</td>
<td>2022/05/17</td>
<td>Question Answering and Dialogue Systems</td>
</tr>
<tr>
<td>14</td>
<td>2022/05/24</td>
<td>Deep Learning, Transfer Learning, Zero-Shot, and Few-Shot Learning for Text Analytics</td>
</tr>
<tr>
<td>15</td>
<td>2022/05/31</td>
<td>Final Project Report I</td>
</tr>
<tr>
<td>16</td>
<td>2022/06/07</td>
<td>Final Project Report II</td>
</tr>
<tr>
<td>17</td>
<td>2022/06/14</td>
<td>Self-learning</td>
</tr>
<tr>
<td>18</td>
<td>2022/06/21</td>
<td>Self-learning</td>
</tr>
</tbody>
</table>
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

Text Mining “Knowledge Discovery in Textual Data”

Document Matching
Link Analysis
Search Engines
POS Tagging
Lemmatization
Word Disambiguation

Web Content Mining
Web Structure Mining
Web Usage Mining
Classification
Clustering
Association

Data Mining

Natural Language Processing

Statistics
Machine Learning
Management Science
Artificial Intelligence
Computer Science
Other Disciplines

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

Artificial intelligence
Machine learning
Natural Language Processing
Deep learning

Artificial Intelligence (AI)
Evolution of Computerized Decision Support to Analytics/Data Science

The timeline in Figure 1.8 shows the terminology used to describe analytics since the 1970s. During the 1970s, the primary focus of information systems support for decision making focused on providing structured, periodic reports that a manager could use for decision making (or ignore them). Businesses began to create routine reports to inform decision makers (managers) about what had happened in the previous period (e.g., day, week, month, quarter). Although it was useful to know what had happened in the past, managers needed more than this: They needed a variety of reports at different levels of granularity to better understand and address changing needs and challenges of the business. These were usually called management information systems (MIS). In the early 1970s, Scott-Morton first articulated the major concepts of DSS. He defined DSSs as "interactive computer-based systems, which help decision makers utilize data and models to solve unstructured problems" (Gorry and Scott-Morton, 1971). The following is another classic DSS definition, provided by Keen and Scott-Morton (1978):

"Decision support systems couple the intellectual resources of individuals with the capabilities of the computer to improve the quality of decisions. It is a computer-based support system for management decision makers who deal with semistructured problems."

Note that the term decision support system, like management information system and several other terms in the field of IT, is a content-free expression (i.e., it means different things to different people). Therefore, there is no universally accepted definition of DSS.

During the early days of analytics, data was often obtained from the domain experts using manual processes (i.e., interviews and surveys) to build mathematical or knowledge-based models to solve constrained optimization problems. The idea was to do the best with limited resources. Such decision support models were typically called operations research (OR). The problems that were too complex to solve optimally (using linear or nonlinear mathematical programming techniques) were tackled using heuristic methods such as simulation models. (We will introduce these as prescriptive analytics later in this chapter and in a bit more detail in Chapter 6.)

In the late 1970s and early 1980s, in addition to the mature OR models that were being used in many industries and government systems, a new and exciting line of models had emerged: rule-based expert systems. These systems promised to capture experts’ knowledge in a format that computers could process (via a collection of if–then–else rules or heuristics) so that these could be used for consultation much the same way that one...
The Rise of AI

1.1 Origin & Definition of AI

Artificial intelligence (AI) is not new. The term was coined in 1956 by John McCarthy, a Stanford computer science professor who organized an academic conference on the topic at Dartmouth College in the summer of that year. The field of AI has gone through a series of boom-bust cycles since then, characterized by technological breakthroughs that stirred activity and excitement about the topic, followed by subsequent periods of disillusionment and disinterest known as ‘AI Winters’ as technical limitations were discovered. As you can see in figure 1, today we are once again in an ‘AI Spring’.

Artificial intelligence can be defined as human intelligence exhibited by machines; systems that approximate, mimic, replicate, automate, and eventually improve on human thinking. Throughout the past half-century a few key components of AI were established as essential: the ability to perceive, understand, learn, problem solve, and reason. Countless working definitions of AI have been proposed over the years but the unifying thread in all of them is that computers with the right software can be used to solve the kind of problems that humans solve, interact with humans and the world as humans do, and create ideas like humans. In other words, while the mechanisms that give rise to AI are ‘artificial’, the intelligence to which AI is intended to approximate is indistinguishable from human intelligence. In the early days of the science, processing inputs from the outside world required extensive programming, which limited early AI systems to a very narrow set of inputs and conditions. However since then, computer science has worked to advance the capability of AI-enabled computing systems.

Board games have long been a proving ground for AI research, as they typically involve a finite number of players, rules, objectives, and possible moves. This essentially means that games – one by one, including checkers, backgammon, and even Jeopardy! to name a few – have been taken over by AI. Most famously, in 1997 IBM's Deep Blue defeated Garry Kasparov, the then reigning world champion of chess. This trajectory persists with the ancient Chinese game of Go, and the defeat of reigning world champion Lee Sedol by DeepMind's AlphaGo in March 2016.

**Understanding Artificial Intelligence**

- Theory and applications of AI.
- Real-world problems are complicated.
- Facial recognition, translation.
- Combinatorial explosion.
- Disappointing results: failure to achieve scale.
- Collapse of dedicated hardware vendors.
- Limited computer processing power.
- Limited database storage capacity.
- Limited network ability.

**AI Winter I**

- 1950: Eliza, the first chatbot is developed by Joseph Weizenbaum at MIT.
- 1960: Edward Feigenbaum develops the first Expert System, giving rebirth to AI.

**AI Winter II**

- 2000: Apple integrates Siri, a personal voice assistant into the iPhone.
- 2010: YouTube recognizes cats from videos.
- 2014: AlphaGo defeats Lee Sedol.
- 2016: YouTube recognizes cats from videos.

**Timeline**

- 1956: The Turing Test.
- 1956: Dartmouth College conference led by John McCarthy coins the term ‘artificial intelligence’.
- 1997: IBM’s Deep Blue defeats Garry Kasparov, the world’s reigning chess champion.
- 2000: IBM’s Watson Q&A machine wins Jeopardy!
- 2011: Apple integrates Siri, a personal voice assistant into the iPhone.
- 2014: AlphaGo defeats Lee Sedol.
- 2016: YouTube recognizes cats from videos.

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

“... the science and engineering of making intelligent machines”

(John McCarthy, 1955)

Artificial Intelligence

“... technology that thinks and acts like humans”
Artificial Intelligence

“... intelligence exhibited by machines or software”

<table>
<thead>
<tr>
<th>Thinking Humanly</th>
<th>Thinking Rationally</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acting Humanly</td>
<td>Acting Rationally</td>
</tr>
</tbody>
</table>

## 4 Approaches of AI

| 1. | Acting Humanly: The Turing Test Approach (1950) |
| 2. | Thinking Humanly: The Cognitive Modeling Approach |
| 3. | Thinking Rationally: The “Laws of Thought” Approach |
| 4. | Acting Rationally: The Rational Agent Approach |

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

Text Analytics
and
Text Mining
Dipanjan Sarkar (2019),

Text Analytics with Python:
A Practitioner’s Guide to Natural Language Processing,

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


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

Source: https://www.amazon.com/Practical-Natural-Language-Processing-Pragmatic/dp/1492054054
<table>
<thead>
<tr>
<th>FOUNDATIONS</th>
<th>Covered in Chapters 1 to 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML for NLP</td>
<td>NLP Pipelines</td>
</tr>
<tr>
<td>Data</td>
<td>Gathering</td>
</tr>
<tr>
<td>Multilingual</td>
<td>NLP</td>
</tr>
<tr>
<td>Text</td>
<td>Representation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CORE TASKS</th>
<th>Covered in Chapters 3 to 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Classification</td>
</tr>
<tr>
<td>Information</td>
<td>Extraction</td>
</tr>
<tr>
<td>Conversational</td>
<td>Agents</td>
</tr>
<tr>
<td>Information</td>
<td>Retrieval</td>
</tr>
<tr>
<td>Question</td>
<td>Answering</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GENERAL APPLICATIONS</th>
<th>Covered in Chapters 4 to 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spam</td>
<td>Classification</td>
</tr>
<tr>
<td>Calendar Event</td>
<td>Extraction</td>
</tr>
<tr>
<td>Personal Assistants</td>
<td></td>
</tr>
<tr>
<td>Search Engines</td>
<td>Jeopardy!</td>
</tr>
<tr>
<td>Jeopardy!</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INDUSTRY SPECIFIC</th>
<th>Covered in Chapters 8 to 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Media</td>
<td>Analysis</td>
</tr>
<tr>
<td>Retail Data</td>
<td>Extraction</td>
</tr>
<tr>
<td>Health Records</td>
<td>Analysis</td>
</tr>
<tr>
<td>Financial Analysis</td>
<td></td>
</tr>
<tr>
<td>Legal Entity</td>
<td>Extraction</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AI PROJECT PLAYBOOK</th>
<th>Covered in Chapters 2 &amp; 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Processes</td>
<td></td>
</tr>
<tr>
<td>Best Practices</td>
<td></td>
</tr>
<tr>
<td>Model Iterations</td>
<td></td>
</tr>
<tr>
<td>MLOps</td>
<td>AI Teams &amp; Hiring</td>
</tr>
</tbody>
</table>


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

Statistics

Database Systems

Natural Language Processing

Information Retrieval

Machine Learning

Pattern Recognition

Visualization

Algorithms

High-performance Computing

Applications

Text Mining

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

Example of Opinion: review segment on iPhone

“I bought an iPhone a few days ago.

It was such a nice phone.

The touch screen was really cool.

The voice quality was clear too.

However, my mother was mad with me as I did not tell her before I bought it.

She also thought the phone was too expensive, and wanted me to return it to the shop. …”

“(1) I bought an 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. ...”

Sentiment Analysis

- Subjectivity Classification
- Sentiment Classification
- Review Usefulness Measurement
- Opinion Spam Detection
- Lexicon Creation
- Aspect Extraction
- Application

Polarity Determination
Vagueness resolution in opinionated text
Multi- & Cross-Lingual SC
Cross-domain SC

Approaches
- Machine Learning based
- Lexicon based
- Hybrid approaches
- Ontology based
- Non-Ontology based

Tasks

Sentiment Classification Techniques

Sentiment Analysis

- Machine Learning Approach
  - Supervised Learning
    - Decision Tree Classifiers
    - Linear Classifiers
    - Rule-based Classifiers
    - Probabilistic Classifiers
    - Support Vector Machine (SVM)
    - Neural Network (NN)
    - Deep Learning (DL)
    - Naïve Bayes (NB)
    - Bayesian Network (BN)
    - Maximum Entropy (ME)

- Lexicon-based Approach
  - Unsupervised Learning
    - Dictionary-based Approach
      - Statistical
      - Semantic
  - Corpus-based Approach

P–N Polarity and S–O Polarity Relationship

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

Web Content Mining
Source: unstructured textual content of the Web pages (usually in HTML format)

Web Structure Mining
Source: the unified resource locator (URL) links contained in the Web pages

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

Search Engines
Page Rank
Search Engine Optimization
Marketing Attribution

Sentiment Analysis
Information Retrieval
Social Network Analysis
Customer Analytics

Semantic Webs
Graph Mining
Social Media Analytics
360 Customer View

Web Analytics
Social Analytics
Clickstream Analysis
Weblog Analysis
Voice of the Customer

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

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

Social Analytics

- Social Network Analysis (SNA)
- Social Media Analytics

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

Natural Language Processing
(NLP)
Text mining

Text Data Mining

Intelligent Text Analysis

Knowledge-Discovery in Text (KDT)

Text Mining
(text data mining)

the process of deriving high-quality information from text

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

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

An example of Text Mining

Analyse Text
- Information Extraction
- Classification
- Summarization
- Clustering

Retrieve and preprocess document

Document Collection

Overview of Information Extraction based Text Mining Framework

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

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

- Spell Checking
- Keyword-Based Information Retrieval
- Topic Modeling
- Text Classification
- Information Extraction
- Closed Domain Conversational Agent
- Text Summarization
- Question Answering
- Machine Translation
- Open Domain Conversational Agent

Building Blocks of Language and Applications

- **Context**
  - meaning

- **Syntax**
  - phrases & sentences

- **Morphemes & Lexemes**
  - words

- **Phonemes**
  - speech & sounds

**Blocks of Language**

**Applications**

- Summarization
- Topic Modeling
- Sentiment Analysis
- Parsing
- Entity Extraction
- Relation Extraction
- Tokenization
- Word Embeddings
- POS Tagging
- Speech to Text
- Speaker Identification
- Text to Speech

Morpheme Examples

unbreakable
un + break + able

cats
cat + s

tumbling
tumble + ing

unreliability
un + rely + able + ity

Syntactic Structure

Text Summarization

Topic Modeling

Natural Language Processing (NLP)

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

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

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

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

Classical NLP

Deep Learning-based NLP

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

Modern NLP Pipeline

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

[Diagram showing a process flow from Documents to Preprocessing, then to Dense Word Embeddings, followed by a Deep Neural Network. Tasks/Outputs include Classification, Sentiment Analysis, Entity Extraction, Topic Modeling, and Document Similarity.]
Text Classification

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

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

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

Text Classification S/W>=1500: Sequence

Select top_k features [freq]

min(top_1K, 2K, ... 15K, 20K, 25K, ... 90K, all)

Normalization mode

samplewise

None

featurewise

Embeddings

Yes

S/W < 15K

Fine-tuned pre-trained embedding

Frozen pre-trained embedding

Embeddings learned from scratch

No

Build model

RNN

stacked RNN

CNN-RNN

sepCNN

CNN

Hyperparameter tuning

Step 2.5: Choose a Model

Samples/Words < 1500

150,000/100 = 1500

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

Samples/Words < 15,000

1,500,000 / 100 = 15,000

Step 3: Prepare Your Data

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

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

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

'The mouse ran up the clock’ =

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

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

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

![Diagram showing Word embeddings]

- **Male-Female**
- **Verb Tense**
- **Country-Capital**

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

The mouse ran up the clock

The mouse ran down

The mouse ran up the clock

The mouse ran down

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

Word Embeddings

Word2Vec

GloVe
Modern NLP Pipeline
Facebook Research FastText

Pre-trained word vectors
Word2Vec
wiki.zh.vec (861MB)
332647 word
300 vec

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

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


https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md
Facebook Research FastText Word2Vec: wiki.zh.vec (861MB) (332647 word 300 vec)

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
- 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
Word Embeddings in LSTM RNN

Time Expanded LSTM Network

LSTM Internal States

Word Embeddings

Input Question
Is this person dancing?

Fixed length question vector encoded by the LSTM

Source: https://avisingh599.github.io/deeplearning/visual-qa/
Sequence to Sequence (Seq2Seq)

Source: https://google.github.io/seq2seq/
Transformer (Attention is All You Need)
(Vaswani et al., 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

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

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

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

BERT (Bidirectional Encoder Representations from Transformers)

**BERT input representation**

<table>
<thead>
<tr>
<th>Input</th>
<th>[CLS]</th>
<th>my</th>
<th>dog</th>
<th>is</th>
<th>cute</th>
<th>[SEP]</th>
<th>he</th>
<th>likes</th>
<th>play</th>
<th>#ing</th>
<th>[SEP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token Embeddings</td>
<td>$E_{[CLS]}$</td>
<td>$E_{my}$</td>
<td>$E_{dog}$</td>
<td>$E_{is}$</td>
<td>$E_{cute}$</td>
<td>$E_{[SEP]}$</td>
<td>$E_{he}$</td>
<td>$E_{likes}$</td>
<td>$E_{play}$</td>
<td>$E_{#ing}$</td>
<td>$E_{[SEP]}$</td>
</tr>
<tr>
<td>Segment Embeddings</td>
<td>$E_A$</td>
<td>$E_A$</td>
<td>$E_A$</td>
<td>$E_A$</td>
<td>$E_A$</td>
<td>$E_B$</td>
<td>$E_B$</td>
<td>$E_B$</td>
<td>$E_B$</td>
<td>$E_B$</td>
<td>$E_B$</td>
</tr>
<tr>
<td>Position Embeddings</td>
<td>$E_0$</td>
<td>$E_1$</td>
<td>$E_2$</td>
<td>$E_3$</td>
<td>$E_4$</td>
<td>$E_5$</td>
<td>$E_6$</td>
<td>$E_7$</td>
<td>$E_8$</td>
<td>$E_9$</td>
<td>$E_{10}$</td>
</tr>
</tbody>
</table>

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

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

BERT Sequence-level tasks

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

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

BERT Token-level tasks

(c) Question Answering Tasks: SQuAD v1.1

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

## General Language Understanding Evaluation (GLUE) benchmark

### GLUE Test results

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

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

Pre-trained Language Model (PLM)

Source: https://github.com/thunlp/PLMpapers
Transformers Pre-trained Language Model

90+ Models

Scaling Transformers

The diagram shows a log-log plot of the number of parameters vs. release date for various transformer models. The x-axis represents the release date, with marks at 2017-07, 2018-01, 2018-07, 2019-01, 2019-07, 2020-01, 2020-07, and 2021-01. The y-axis represents the number of parameters in units of 10^8 to 10^13. The models are plotted as points, with names such as Transformer, GPT, BERT, GPT-2, Megatron, T5, GShard, GPT-3, Turing-NLG, and Switch-C. The data shows an increase in the number of parameters over time for most models.
Pre-trained Models (PTM)

- **Contextual?**
  - **Non-Contextual**
    - CBOW, Skip-Gram [129]
    - GloVe [133]
  - **Contextual**
    - ELMo [135], GPT [142], BERT [36]

- **Architectures**
  - LSTM
    - LM-LSTM [30], Shared LSTM[109], ELMo [135], CoVe [126]
  - Transformer Enc.
    - BERT [36], SpanBERT [117], XLNet [209], RoBERTa [117]
  - Transformer Dec.
    - GPT [142], GPT-2 [143]
  - Transformer
    - MASS [160], BART [100]
    - XNLG [19], mBART [118]

- **Task Types**
  - Supervised
    - MT
      - CoVe [126]
    - LM
      - ELMo [135], GPT [142], GPT-2 [143], UniLM [39]
      - BERT [36], SpanBERT [117], RoBERTa [117], XLM-R [28]
    - MLM
      - TLM
        - XLM [27]
      - Seq2Seq MLM
        - MASS [160], T5 [144]
    - PLM
      - XLNet [209]
    - DAE
      - BART [100]

- **PTMs**
  - Unsupervised/Self-Supervised
    - RTD
      - CBOW-NS [129], ELECTRA [24]
    - CTL
      - NSP
        - BERT [36], UniLM [39]
    - SOP
      - ALBERT [93], StructBERT [193]

Pre-trained Models (PTM)

Transformer • 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
The AI community building the future.

Build, train and deploy state of the art models powered by the reference open source in machine learning.

https://huggingface.co/
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 be 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.

https://huggingface.co/docs/transformers/index
Hugging Face Tasks
Natural Language Processing

Text Classification: 3345 models
Token Classification: 1492 models
Question Answering: 1140 models
Translation: 1467 models
Summarization: 323 models
Text Generation: 3959 models
Fill-Mask: 2453 models
Sentence Similarity: 352 models

https://huggingface.co/tasks
Text Analytics with Python
NLP Libraries and Tools
spaCy:
Natural Language Processing

Industrial-Strength Natural Language Processing

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

Blazing fast
spaCy excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython. Independent research in 2015 found spaCy to be the fastest in the world. If your application needs to process entire web dumps, spaCy is the library you want to be using.

Deep learning
spaCy is the best way to prepare text for deep learning. It interoperates seamlessly with TensorFlow, PyTorch, scikit-learn, Gensim and the rest of Python's awesome AI ecosystem. With spaCy, you can easily construct linguistically sophisticated statistical models for a variety of NLP problems.

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

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

Bibliography
Term Index

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

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

Gensim is a FREE Python library

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

https://radimrehurek.com/gensim/
TextBlob

TextBlob: Simplified Text Processing

Release v0.12.0. (Changelog)

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, classification, translation, and more.

```python
from textblob import TextBlob

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

blob = TextBlob(text)

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

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

for sentence in blob.sentences:
    print(sentence.sentiment.polarity)
    # 0.050
```

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

polyglot

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

- Free software: GPLv3 license
- Documentation: http://polyglot.readthedocs.org

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/
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/1FEG6DnGvwfUbeo4z1zTunjMqf2RkCrT

---

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)

---

Python for Natural Language Processing

spaCy

• spaCy: Industrial-Strength Natural Language Processing in Python
• Source: https://spacy.io/usage/spacy-101

```python
1: python -m spacy download en_core_web_sm

3:
1 import spacy
2 nlp = spacy.load("en_core_web_sm")
3 doc = nlp("Apple is looking at buying U.K. startup for $1 billion")
4 for token in doc:
5     print(token.text, token.pos_, token.dep_)
```

---

Apple PROPN nsubj
is AUX aux
looking VERB ROOT
at ADP prep
buying VERB pcomp
U.K. PROPN compound
startup NOUN dobj
for ADP prep
$ SYM quantmod
1 NOUN compound
billion NUM pobj

https://tinyurl.com/aintpuppython101
Python in Google Colab (Python101)

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

```
1 import spacy
2 nlp = spacy.load('en_core_web_sm')
3 doc = nlp('Apple is looking at buying U.K. startup for $1 billion')
4 import pandas as pd
5 cols = ['text', 'lemma', 'POS', 'explain', 'stopword']
6 rows = []
7 for t in doc:
8    row = {t.text, t.lemma_, t.pos_, spacy.explain(t.pos_)[is_stop]
9    rows.append(row)
10 df = pd.DataFrame(rows, columns=cols)
11 df
```

<table>
<thead>
<tr>
<th>text</th>
<th>lemma</th>
<th>POS</th>
<th>explain</th>
<th>stopword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>Apple</td>
<td>PROP</td>
<td>proper noun</td>
<td>False</td>
</tr>
<tr>
<td>is</td>
<td>be</td>
<td>VERB</td>
<td>verb</td>
<td>True</td>
</tr>
<tr>
<td>looking</td>
<td>look</td>
<td>VERB</td>
<td>verb</td>
<td>False</td>
</tr>
<tr>
<td>at</td>
<td>at</td>
<td>ADP</td>
<td>adposition</td>
<td>True</td>
</tr>
<tr>
<td>buying</td>
<td>buy</td>
<td>VERB</td>
<td>verb</td>
<td>False</td>
</tr>
<tr>
<td>U.K.</td>
<td>U.K.</td>
<td>PROP</td>
<td>proper noun</td>
<td>False</td>
</tr>
<tr>
<td>startup</td>
<td>startup</td>
<td>NOUN</td>
<td>noun</td>
<td>False</td>
</tr>
<tr>
<td>for</td>
<td>for</td>
<td>ADP</td>
<td>adposition</td>
<td>True</td>
</tr>
<tr>
<td>$</td>
<td>$</td>
<td>SYM</td>
<td>symbol</td>
<td>False</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>NUM</td>
<td>numeral</td>
<td>False</td>
</tr>
<tr>
<td>billion</td>
<td>billion</td>
<td>NUM</td>
<td>numeral</td>
<td>False</td>
</tr>
</tbody>
</table>
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

```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.
from spacy import disqaly

text = "Stanford University is located in California. It is a great university."

doc = nlp(text)
disqaly.render(doc, style="ent", jupyter=True)
disqaly.render(doc, style="dep", jupyter=True)
Python in Google Colab (Python101)

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

https://tinyurl.com/aintpuppython101
NLP with Transformers Github

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

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Colab</th>
<th>Kaggle</th>
<th>Gradient</th>
<th>Studio Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Text Classification</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Transformer Anatomy</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Multilingual Named Entity Recognition</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Text Generation</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Summarization</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Question Answering</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Making Transformers Efficient in Production</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Dealing with Few to No Labels</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Training Transformers from Scratch</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
<tr>
<td>Future Directions</td>
<td>![Open in Colab]</td>
<td>![Open in Kaggle]</td>
<td>![Run on Gradient]</td>
<td>![Open in Studio Lab]</td>
</tr>
</tbody>
</table>

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)
<table>
<thead>
<tr>
<th>Task</th>
<th>Dataset</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WMT 2014 EN-FR</td>
<td></td>
</tr>
<tr>
<td>Text Summarization</td>
<td>CNN/DM</td>
<td><a href="https://cs.nyu.edu/~kcho/DMQA/">https://cs.nyu.edu/~kcho/DMQA/</a></td>
</tr>
<tr>
<td></td>
<td>Newsroom</td>
<td><a href="https://summar.es/">https://summar.es/</a></td>
</tr>
<tr>
<td></td>
<td>Gigaword</td>
<td><a href="https://catalog.ldc.upenn.edu/LDC2012T21">https://catalog.ldc.upenn.edu/LDC2012T21</a></td>
</tr>
<tr>
<td>Reading Comprehension</td>
<td>ARC</td>
<td><a href="http://data.allenai.org/arc/">http://data.allenai.org/arc/</a></td>
</tr>
<tr>
<td>Question Answering</td>
<td>CICR</td>
<td><a href="http://aclweb.org/anthology/N18-1140">http://aclweb.org/anthology/N18-1140</a></td>
</tr>
<tr>
<td>Question Generation</td>
<td>CNN/DM</td>
<td><a href="https://cs.nyu.edu/~kcho/DMQA/">https://cs.nyu.edu/~kcho/DMQA/</a></td>
</tr>
<tr>
<td></td>
<td>NewsQA</td>
<td><a href="https://datasets.maluuba.com/NewsQA">https://datasets.maluuba.com/NewsQA</a></td>
</tr>
<tr>
<td></td>
<td>RACE</td>
<td><a href="http://www.qizhexie.com/data/RACE_leaderboard">http://www.qizhexie.com/data/RACE_leaderboard</a></td>
</tr>
<tr>
<td></td>
<td>SQuAD</td>
<td><a href="https://rajpurkar.github.io/SQuAD-explorer/">https://rajpurkar.github.io/SQuAD-explorer/</a></td>
</tr>
<tr>
<td></td>
<td>NarrativeQA</td>
<td><a href="https://github.com/deepmind/narrativeqa">https://github.com/deepmind/narrativeqa</a></td>
</tr>
<tr>
<td></td>
<td>Quasar</td>
<td><a href="https://github.com/bdhiingra/quasar">https://github.com/bdhiingra/quasar</a></td>
</tr>
<tr>
<td></td>
<td>SearchQA</td>
<td><a href="https://github.com/nyu-dl/SearchQA">https://github.com/nyu-dl/SearchQA</a></td>
</tr>
<tr>
<td>Semantic Parsing</td>
<td>AMR Parsing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATIS (SQL Parsing)</td>
<td><a href="https://github.com/jkkummerfeld/text2sql-data/tree/master/data">https://github.com/jkkummerfeld/text2sql-data/tree/master/data</a></td>
</tr>
<tr>
<td></td>
<td>WikiSQL (SQL Parsing)</td>
<td><a href="https://github.com/salesforce/wikiSQL">https://github.com/salesforce/wikiSQL</a></td>
</tr>
<tr>
<td>Sentiment Analysis</td>
<td>IMDB Reviews</td>
<td><a href="http://ai.stanford.edu/~amaas/data/sentiment/">http://ai.stanford.edu/~amaas/data/sentiment/</a></td>
</tr>
<tr>
<td></td>
<td>SST</td>
<td><a href="https://nlp.stanford.edu/sentiment/index.html">https://nlp.stanford.edu/sentiment/index.html</a></td>
</tr>
<tr>
<td></td>
<td>Yelp Reviews</td>
<td><a href="https://www.yelp.com/dataset/challenge">https://www.yelp.com/dataset/challenge</a></td>
</tr>
<tr>
<td></td>
<td>Subjectivity Dataset</td>
<td><a href="http://www.cs.cornell.edu/people/pabo/movie-review-data/">http://www.cs.cornell.edu/people/pabo/movie-review-data/</a></td>
</tr>
<tr>
<td></td>
<td>DBpedia</td>
<td><a href="https://wiki.dbpedia.org/Datasets">https://wiki.dbpedia.org/Datasets</a></td>
</tr>
<tr>
<td></td>
<td>TREC</td>
<td><a href="https://trec.nist.gov/data.html">https://trec.nist.gov/data.html</a></td>
</tr>
<tr>
<td>Natural Language Inference</td>
<td>SNLI Corpus</td>
<td><a href="https://nlp.stanford.edu/projects/snli/">https://nlp.stanford.edu/projects/snli/</a></td>
</tr>
<tr>
<td></td>
<td>MultiNLI</td>
<td><a href="https://www.nyu.edu/projects/bowman/multinli/">https://www.nyu.edu/projects/bowman/multinli/</a></td>
</tr>
<tr>
<td></td>
<td>SciTail</td>
<td><a href="http://data.allenai.org/scitail/">http://data.allenai.org/scitail/</a></td>
</tr>
<tr>
<td>Semantic Role Labeling</td>
<td>Proposition Bank</td>
<td><a href="http://propbank.github.io/">http://propbank.github.io/</a></td>
</tr>
<tr>
<td></td>
<td>OneNotes</td>
<td><a href="https://catalog.ldc.upenn.edu/LDC2013T19">https://catalog.ldc.upenn.edu/LDC2013T19</a></td>
</tr>
</tbody>
</table>

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-Chenaghlou (2021), Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques, Packt Publishing.
- 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.
- Christopher D. Manning and Hinrich Schütze (1999), Foundations of Statistical Natural Language Processing, The MIT Press.