人工智慧文本分析
(Artificial Intelligence for Text Analytics)
深度學習和通用句子嵌入模型
(Deep Learning and Universal Sentence-Embedding Models)

1082AITA11
MBA, IMTKU (M2455) (8410) (Spring 2020)
Wed 8, 9 (15:10-17:00) (B605)

Min-Yuh Day
戴敏育
Associate Professor
副教授
Dept. of Information Management, Tamkang University

http://mail.tku.edu.tw/myday/
2020-06-03
<table>
<thead>
<tr>
<th>過次 (Week)</th>
<th>日期 (Date)</th>
<th>內容 (Subject/Topics)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2020/03/04</td>
<td>人工智慧文本分析課程介紹</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Course Orientation on Artificial Intelligence for Text Analytics)</td>
</tr>
<tr>
<td>2</td>
<td>2020/03/11</td>
<td>文本分析的基礎：自然語言處理</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Foundations of Text Analytics: Natural Language Processing; NLP)</td>
</tr>
<tr>
<td>3</td>
<td>2020/03/18</td>
<td>Python自然語言處理</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Python for Natural Language Processing)</td>
</tr>
<tr>
<td>4</td>
<td>2020/03/25</td>
<td>處理和理解文本</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Processing and Understanding Text)</td>
</tr>
<tr>
<td>5</td>
<td>2020/04/01</td>
<td>文本表達特徵工程</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Feature Engineering for Text Representation)</td>
</tr>
<tr>
<td>6</td>
<td>2020/04/08</td>
<td>人工智慧文本分析個案研究1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Case Study on Artificial Intelligence for Text Analytics I)</td>
</tr>
</tbody>
</table>
課程大綱 (Syllabus)

週次 (Week)  日期 (Date)  內容 (Subject/Topics)
7  2020/04/15  文本分類 (Text Classification)
8  2020/04/22  文本摘要和主題模型
(Text Summarization and Topic Models)
9  2020/04/29  期中報告 (Midterm Project Report)
10  2020/05/06  文本相似度和分群 (Text Similarity and Clustering)
11  2020/05/13  語意分析和命名實體識別
(Semantic Analysis and Named Entity Recognition; NER)
12  2020/05/20  情感分析 (Sentiment Analysis)
<table>
<thead>
<tr>
<th>週次 (Week)</th>
<th>日期 (Date)</th>
<th>內容 (Subject/Topics)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>2020/05/27</td>
<td>人工智慧文本分析個案研究 II (Case Study on Artificial Intelligence for Text Analytics II)</td>
</tr>
<tr>
<td>14</td>
<td>2020/06/03</td>
<td>深度學習和通用句子嵌入模型 (Deep Learning and Universal Sentence-Embedding Models)</td>
</tr>
<tr>
<td>15</td>
<td>2020/06/10</td>
<td>問答系統與對話系統 (Question Answering and Dialogue Systems)</td>
</tr>
<tr>
<td>16</td>
<td>2020/06/17</td>
<td>期末報告 I (Final Project Presentation I)</td>
</tr>
<tr>
<td>17</td>
<td>2020/06/24</td>
<td>期末報告 II (Final Project Presentation II)</td>
</tr>
<tr>
<td>18</td>
<td>2020/07/01</td>
<td>教師彈性補充教學</td>
</tr>
</tbody>
</table>
Deep Learning and Universal Sentence-Embedding Models
Outline

• Universal Sentence Encoder (USE)

• Universal Sentence Encoder Multilingual (USEM)

• Semantic Similarity
Universal Sentence Encoder (USE)

• The Universal Sentence Encoder encodes text into high-dimensional vectors that can be used for text classification, semantic similarity, clustering and other natural language tasks.

• The universal-sentence-encoder model is trained with a deep averaging network (DAN) encoder.
Universal Sentence Encoder (USE) Semantic Similarity

"How old are you?" [0.3, 0.2, ...]
"What is your age?" [0.2, 0.1, ...]
"My phone is good." [0.9, 0.6, ...]

Source: https://tfhub.dev/google/universal-sentence-encoder/4
Universal Sentence Encoder (USE) Classification

"How old are you?"  [0.3, 0.2, ...]  (96%) "How old are you?"
"What is your age?"  [0.2, 0.1, ...]  (98%) "What is your age?"
"My phone is good."  [0.9, 0.6, ...]  (7%) "My phone is good."

Source: https://tfhub.dev/google/universal-sentence-encoder/4
NLP

Classical NLP

Documents -> Language Detection

English -> Pre-processing

Spanish -> Pre-processing

Arabic -> Pre-processing

Pre-processing:

Tokanization (English)

Pos Tagging (English)

Stopword Removal (EN)

...>

Modeling:

Feature Extraction (EN)

Modeling (English)

Inference (English)

Output:

Sentiment

Classification

Entity Extraction

Translation

Topic Modelling

Deep Learning-based NLP

Documents -> Preprocessing

Dense Embeddings

obtained via word2vec, doc2vec, GloVe, etc.

Hidden Layers

Output Units

Output:

Sentiment

Classification

Entity Extraction

Translation

Topic Modelling

...
Modern NLP Pipeline

Modern NLP Pipeline

Documents

Language Detection

Preprocessing

Preprocessing

Preprocessing

Modeling

Modeling

Modeling

Task / Output

Classification

Sentiment Analysis

Entity Extraction

Topic Modeling

Document Similarity

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

Documents → Preprocessing → Dense Word Embeddings → Deep Neural Network

- Pre-generated Lookup OR Generated in 1st level of NeuralNet

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

Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/
Natural Language Processing (NLP) and Text Mining

1. Raw text
2. Sentence Segmentation
3. Tokenization
4. Part-of-Speech (POS)
5. Stop word removal
6. Stemming / Lemmatization
7. Dependency Parser
8. String Metrics & Matching

Example of word’s stem and word’s lemma:
- am → am
- having → hav
- am → be
- having → have

Source: Nitin Hardeniya (2015), NLTK Essentials, Packt Publishing; Florian Leitner (2015), Text mining - from Bayes rule to dependency parsing
Deep Learning and Universal Sentence-Embedding Models

Universal Sentence Encoder (USE)

- Source: Universal Sentence Encoder: https://tfhub.dev/google/universal-sentence-encoder/4

```python
[ ] 1 import tensorflow as tf
2 import tensorflow_hub as hub
3 import numpy as np
4 import pandas as pd
5 import os
6 import re
7 import matplotlib.pyplot as plt
8 import seaborn as sns
9
10 module_url = "https://tfhub.dev/google/universal-sentence-encoder/4"
11 #"https://tfhub.dev/google/universal-sentence-encoder-large/5"
12 model = hub.load(module_url)
13 print("module is loaded" % module_url)
14 def embed(input):
15     return model(input)

[ ] module https://tfhub.dev/google/universal-sentence-encoder/4 loaded

[ ] 1 word = "Elephant"
2 sentence = "I am a sentence for which I would like to get its embedding."
```

https://tinyurl.com/imtkupyteron101
Python in Google Colab (Python101)

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

Semantic Textual Similarity

1. I like my phone
2. My phone is not good.
3. Your cellphone looks great.
4. Will it snow tomorrow?
5. Recently a lot of hurricanes have hit the US
6. Global warming is real
7. An apple a day, keeps the doctors away
8. Eating strawberries is healthy
9. Is paleo better than keto?
10. How old are you?
11. What is your age?
One-hot encoding

'The mouse ran up the clock' =

<table>
<thead>
<tr>
<th>Word</th>
<th>Index</th>
<th>One-hot 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

Male-Female

Verb Tense

Country-Capital

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

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

Pre-training

Fine-Tuning

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>$E_A$</td>
<td>$E_A$</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>

BERT, OpenAI GPT, ELMo

Fine-tuning BERT on Different 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

Pre-trained Language Model (PLM)

Semi-supervised Sequence Learning context2Vec Pre-trained seq2seq

ULMFiT ELMo Multi-lingual Transformer Bidirectional LM

MultiFiT Cross-lingual

GPT Larger model More data

GPT-2 Defense Whole Word Masking

Grover

by Xiaozhi Wang & Zhengyan Zhang @THUNLP

Source: https://github.com/thunlp/PLMpapers
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]
- Unsupervised/ Self-Supervised
  - 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]
  - RTD
    - CBoW-NS [129], ELECTRA [24]
  - CTL
    - NSP
    - BERT [36], UniLM [39]
  - SOP
    - ALBERT [93], StructBERT [193]
Pre-trained Models (PTM)

- Knowledge-Enriched
  - ERNIE (THU) [214], KnowBERT [136], K-BERT [111]
  - SentiLR [83], KEPLER [195], WKLM [202]

- Multilingual
  - XLU
  - mBERT [36], Unicoder [68], XLM [27], XLM-R [28], MultiFit [42]
  - XLG
  - MASS [160], mBART [118], XNLG [19]

- Language-Specific
  - ERNIE (Baidu) [170], BERT-wwm-Chinese [29], NEZHA [198], ZEN [37]
  - BERTje [33], Camembert [125], FlauBERT [95], RobBERT [35]

- Extensions
  - Image
    - ViLBERT [120], LXMERT [175], VisualBERT [103], B2T2 [2], VL-BERT [163]

- Multi-Modal
  - Video
    - VideoBERT [165], CBT [164]
  - Speech
    - SpeechBERT [22]

- Domain-Specific
  - SentiLR [83], BioBERT [98], SciBERT [11], PatentBERT [97]

- Model Compression
  - Model Pruning
  - CompressingBERT [51]
  - Quantization
    - Q-BERT [156], Q8BERT [211]
  - Parameter Sharing
    - ALBERT [93]
  - Distillation
    - DistilBERT [152], TinyBERT [75], MiniLM [194]
  - Module Replacing
    - BERT-of-Theseus [203]

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
# NLP Benchmark Datasets

<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://summari.es/">https://summari.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 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>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/bdhighra/quasar">https://github.com/bdhighra/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><a href="https://amr.isi.edu/index.html">https://amr.isi.edu/index.html</a></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></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>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>OneNotes</td>
<td><a href="https://catalog.ldc.upenn.edu/LDC2013T19">https://catalog.ldc.upenn.edu/LDC2013T19</a></td>
</tr>
</tbody>
</table>

Summary

- Universal Sentence Encoder (USE)

- Universal Sentence Encoder Multilingual (USEM)

- Semantic Similarity
References


• The Super Duper NLP Repo, https://notebooks.quantumstat.com/