深度學習和通用句子嵌入模型
(Deep Learning and Universal Sentence-Embedding Models)

Time: 2020/06/12 (Fri) (9:10 -12:00)
Place: 國立臺北護理健康大學 (台北市明德路365號) G210
Host: 祝國忠 院長 (健康科技學院院長)

Min-Yuh Day
戴敏育
Associate Professor
副教授
Dept. of Information Management, Tamkang University
淡江大學 資訊管理學系

http://mail.tku.edu.tw/myday/
2020-06-12
Topics

1. Core Technologies of Natural Language Processing and Text Mining
2. Artificial Intelligence for Text Analytics: Foundations and Applications
3. Feature Engineering for Text Representation
4. Semantic Analysis and Named Entity Recognition; NER
5. Deep Learning and Universal Sentence-Embedding Models
6. Question Answering and Dialogue Systems
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.

Source: https://tfhub.dev/google/universal-sentence-encoder/4
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?”
“What is your age?”
“My phone is good.”

Embed

[0.3, 0.2, …]

[0.2, 0.1, …]

[0.9, 0.6, …]

Question Classifier

Confidence is a question
(96%) "How old are you?"
(98%) "What is your age?"
(7%) "My phone is good."

Source: https://tfhub.dev/google/universal-sentence-encoder/4
import tensorflow_hub as hub

embed = hub.Module("https://tfhub.dev/google/
    "universal-sentence-encoder/1")

embedding = embed(["The quick brown fox jumps over the lazy dog."])
import tensorflow_hub as hub

module = hub.Module("https://tfhub.dev/google/universal-sentence-encoder-multilingual/1")

multilingual_embeddings = module(["Hola Mundo!", "Bonjour le monde!", "Ciao mondo!", "Hello World!", "Hallo Welt!", "Hallo Wereld!", "你好世界!", "Привет, мир!", "مرحبا بالعالم!"])
NLP

Classical NLP

Deep Learning-based NLP

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

Documents → Language Detection

EN: Language Detection → Preprocessing → Modeling → Task / Output

ZH: Language Detection → Preprocessing → Preprocessing → Modeling → Task / Output

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

- 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

Source: Nitin Hardeniya (2015), NLTK Essentials, Packt Publishing; Florian Leitner (2015), Text mining - from Bayes rule to dependency parsing
**Python in Google Colab (Python101)**

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

---

Deep Learning and Universal Sentence-Embedding Models

Universal Sentence Encoder (USE)


```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 %s loaded" % module_url)
14 def embed(input):
15   return model(input)

[ ]
1 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/imtkupython101
Python in Google Colab (Python101)

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

https://tinyurl.com/imtkupython101
One-hot encoding

'The mouse ran up the clock' =

<table>
<thead>
<tr>
<th>Word</th>
<th>Index</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]
Word embeddings

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

The mouse ran up the clock
- the: 1
- mouse: 2
- ran: 3
- up: 4
- clock: 5

The mouse ran down
- down: 6

The words are embedded into a vector space:

- "the": [1, 2, 3, 4, 1, 5]
- "mouse": [1, 2, 3, 4, 1, 5]
- "ran": [1, 2, 3, 4, 1, 5]
- "up": [1, 2, 3, 4, 1, 5]
- "clock": [1, 2, 3, 4, 1, 5]
- "down": [1, 2, 3, 4, 1, 5]

The embedding layer (output dim = 4):
- "the": [0.236, -0.141, 0.000, 0.045]
- "mouse": [0.006, 0.652, 0.270, -0.556]
- "ran": [0.305, 0.569, -0.028, 0.496]
- "up": [0.421, 0.195, -0.058, 0.477]
- "clock": [0.236, -0.141, 0.000, 0.045]
- "down": [0.844, -0.001, 0.763, 0.201]
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>$E_B$</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>

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
Transformer
Multi-lingual
Bi-directional LM

MultiFiT
Cross-lingual

Bert
Span prediction
Remove NSP
Longer time
Remove NSP
More data

GPT
Larger model
More data

GPT-2
Defense
Whole Word Masking

VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMER
VL-BERT
UNITER

By Xiaozhi Wang & Zhengyuan Zhang @THUNLP

Source: https://github.com/thunlp/PLMpapers
Turing Natural Language Generation (T-NLG)

Pre-trained Models (PTM)

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], XNLLG [19]

- Language-Specific
  - ERNIE (Baidu) [170], BERT-wwm-Chinese [29], NEZHA [198], ZEN [37]
  - BERTje [33], Camembert [125], FlauBERT [95], RobBERT [35]
  - Image
    - ViLBERT [120], LXMERT [175], VisualBERT [103], B2T2 [4], VL-BERT [163]

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

- Domain-Specific
  - SentiLR [83], BioBERT [98], SciBERT [11], PatentBERT [97]
  - 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](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://summariz.es/">https://summariz.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>CliCR</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.qizhixie.com/data/RACE_leaderboard">http://www.qizhixie.com/data/RACE_leaderboard</a></td>
</tr>
<tr>
<td></td>
<td>SQuAD</td>
<td><a href="https://rajpurkar.github.io/SQuAD_leaderboard">https://rajpurkar.github.io/SQuAD_leaderboard</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/bdvingra/quasar">https://github.com/bdvingra/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>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="http://nlp.stanford.edu/sentiment/index.html">http://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

• Universal Sentence Encoder (USE)

• Universal Sentence Encoder Multilingual (USEM)

• Semantic Similarity
References


• Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Ray Kurzweil (2019). Multilingual Universal Sentence Encoder for Semantic Retrieval.


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

Q & A

深度學習和通用句子嵌入模型
(Deep Learning and Universal Sentence-Embedding Models)

Time: 2020/06/12 (Fri) (9:10 -12:00)
Place: 國立臺北護理健康大學 (台北市明德路365號) G210
Host: 祝國忠 院長 (健康科技學院院長)

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

http://mail.tku.edu.tw/myday/
2020-06-12