Deep Learning (IST, 2021-22)

Practical 11: Word Embeddings and Large Pretrained Models

Taisiya Glushkova, Rita Ramos, André Martins, Ricardo Rei

Question 1

In this question you are going to solve some analogy questions using static word embeddings.

1. Install the `torchtext` package. Download pre-trained GloVe vectors:

```
import torch
from torchtext.vocab import GloVe
glove = GloVe(name='6B', dim=50)
```

2. Compute the following word analogies using vector arithmetic. Provide top-5 closest vectors to each analogy:

   analogy('man', 'actor', 'woman')

   analogy('cat', 'kitten', 'dog')

   analogy('dog', 'puppy', 'cat')

   analogy('russia', 'moscow', 'france')

   analogy('obama', 'president', 'trump')

   analogy('rich', 'mansion', 'poor')

   analogy('elvis', 'rock', 'eminem')

   analogy('paper', 'newspaper', 'screen')

   analogy('monet', 'paint', 'michelangelo')

   analogy('beer', 'barley', 'wine')

   analogy('earth', 'moon', 'sun')

   analogy('house', 'roof', 'castle')

   analogy('building', 'architect', 'software')

   analogy('boston', 'bruins', 'phoenix')

   analogy('good', 'heaven', 'bad')

   analogy('jordan', 'basketball', 'woods')

   Example: analogy('king', 'man', 'queen')

   Output: [king - man + queen = ?]

   (2.8391) woman

(3.3545) girl

(3.9518) boy

(4.0233) her

(4.0554) herself

Solution:

[man - actor + woman = ?]
(2.0527) actress
(3.6065) starred
(3.8781) comedian
(3.9407) starring
(3.9920) entertainer

[cat - kitten + dog = ?]
(3.0314) puppy
(3.2785) rottweiler
(3.5163) spunky
(3.5478) toddler
(3.5482) mannequin

[dog - puppy + cat = ?]
(3.0314) kitten
(3.0836) puppies
(3.2215) pug
(3.2300) frisky
(3.2628) tarantula

[russia - moscow + france = ?]
(2.5632) paris
(3.5555) strasbourg
(3.8609) brussels
(3.9079) lyon
(3.9367) marseille

[obama - president + trump = ?]
(5.1069) debartolo
(5.1298) bally
(5.1754) ebbers
(5.1826) harrah
(5.2083) petronas

[rich - mansion + poor = ?]
(4.4530) bungalow

(4.7109) apartment
(4.7145) residence
(4.7241) dormitory
(4.7605) dilapidated

[elvis - rock + eminem = ?]
(4.5673) rap
(5.1407) hip-hop
(5.1510) rappers
(5.2317) hop
(5.2441) rapper

[paper - newspaper + screen = ?]
(3.4250) tv
(3.5702) television
(4.0667) broadcast
(4.1467) radio
(4.2523) audience

[monet - paint + michelangelo = ?]
(4.7947) molded
(4.8189) microscope
(4.9944) stained
(4.9970) handwriting
(5.0162) plaster

[beer - barley + wine = ?]
(4.1063) grape
(4.4254) legumes
(4.4577) grapes
(4.4731) varieties
(4.5731) beans

[earth - moon + sun = ?]
(4.9071) chung
(4.9905) chan
(4.9941) myung
(4.9970) ho
(5.0008) kim

[house - roof + castle = ?]
(4.7628) moat

(4.9241) fortress
(5.0980) tower
(5.1121) stonework
(5.1523) battlements

[building - architect + software = ?]
(4.4894) programmer
(4.7926) inventor
(5.2666) explorer
(5.2762) innovator
(5.3507) pioneered

[boston - bruins + phoenix = ?]
(2.5751) celtics
(2.6327) mavericks
(2.6589) mavs
(2.6967) suns
(2.7843) lakers

[good - heaven + bad = ?]
(3.2037) hell
(3.6382) curse
(3.7827) eternity
(3.8168) ghosts
(3.8482) madness

[jordan - basketball + woods = ?]
(5.2863) golf
(5.5034) gators
(5.7383) championship
(5.8291) pga
(5.8761) nicklaus

Question 2

In this question you are going to experiment with large pretrained models using the Huggingface's `transformers` library.