CS 224n Assignment #5 Written Part

1.

(a)

Because the total number of the characters is typically smaller than the total number of the words. Additionally, a single word is likely to have a couple of characters, whose vectors are combined to be enough to hold the information of the word.

(b)

Character-based embedding model:

$$e_{char} \times V_{char} + k \times e_{char} \times e_{word} + e_{word} + 2 \times e_{word} \times e_{word} + 2 \times e_{word}$$

$$= e_{char} \times V_{char} + k \times e_{char} \times e_{word} + 2 \times e_{word} \times e_{word} + 3 \times e_{word}$$

Word-based embedding model:

$$e_{word} \times V_{word}$$

Word-based embedding model has more parameters, by as 637 times many.

(c)

When a 1D convnet computes features for a given window of the input, it learns pieces of sentences. By contrast, an RNN, though maybe bi-directional, can only learn the sentence in an integrated way. Therefore, unlike a RNN, a convnet can learn the meanings of pieces of sentences such as phrases better.

(d)

Max-pooling only needs to work out the maximum of inputs, while average-pooling needs to handle all inputs, which renders max-pooling a little bit faster and more convenient. What's more, when inputs are sparse, max-pooling can provide the most significant information while average-pooling may give a rather small value.

Average-pooling can represent the meaning of all inputs in a synthetic way while max-pooling only provides the information of the maximum. Therefore, when the inputs are similar, average-pooling is more accurate.

2.

```
$ sh run.sh test
Decoding: 100%|████████████████████| 8064/8064 [13:13<00:00, 10.17it/s]load test source sentences from [./en_es_data/test.es]
load test target sentences from [./en_es_data/test.en]
load model from model.bin
Corpus BLEU: 24.29222811461412
```

BLEU: 24.29222811461412

3.

(a)

```
Administrator@win-10 MINGW64 /d/Kortez/NLP/cs224n_assignment/CS224-n/a5 (master)
$ cat vocab.json | grep traducir
    "traducirlo": 31948,
    "traducir": 5120,
    "traducir.": 41954,
(pytorch)
Administrator@win-10 MINGW64 /d/Kortez/NLP/cs224n_assignment/CS224-n/a5 (master)
$ cat vocab.json | grep traduzco
(pytorch)
Administrator@win-10 MINGW64 /d/Kortez/NLP/cs224n_assignment/CS224-n/a5 (master)
$ cat vocab.json | grep traduces
(pytorch)
Administrator@win-10 MINGW64 /d/Kortez/NLP/cs224n_assignment/CS224-n/a5 (master)
$ cat vocab.json | grep traduce
    "traducen": 23774,
    "traduce": 8570,
(pytorch)
Administrator@win-10 MINGW64 /d/Kortez/NLP/cs224n_assignment/CS224-n/a5 (master)
$ cat vocab.json | grep traduzca
    "traduzcan": 29908,
(pytorch)
Administrator@win-10 MINGW64 /d/Kortez/NLP/cs224n_assignment/CS224-n/a5 (master)
$ cat vocab.json | grep traduzcas
(pytorch)
Administrator@win-10 MINGW64 /d/Kortez/NLP/cs224n_assignment/CS224-n/a5 (master)
```

As is shown in the picture, *traducir* and *traduce* occur while others don't.

This is bad for word-based NMT because it views them as totally different words and cannot understand their relationship.

In our new character-aware NMT model, the embedding of a word is up to its the embeddings of its characters. Thus, our model can recognize the similarity of these words and can learn better.

(b)

(i)


Search
financial
.*
by
word
neighbors
100
distance
COSINE
EUCLIDEAN
Nearest points in the original space:
economic 0.463
business 0.484
markets 0.516
banking 0.534
finance 0.557
investment 0.558
monetary 0.562
corporate 0.589
market 0.594
money 0.596
economy 0.604
companies 0.606
businesses 0.608
employment 0.609
legal 0.609
assistance 0.611
accounting 0.613
commercial 0.613
technical 0.617
corporations 0.618

Search

neuron .*

by

word

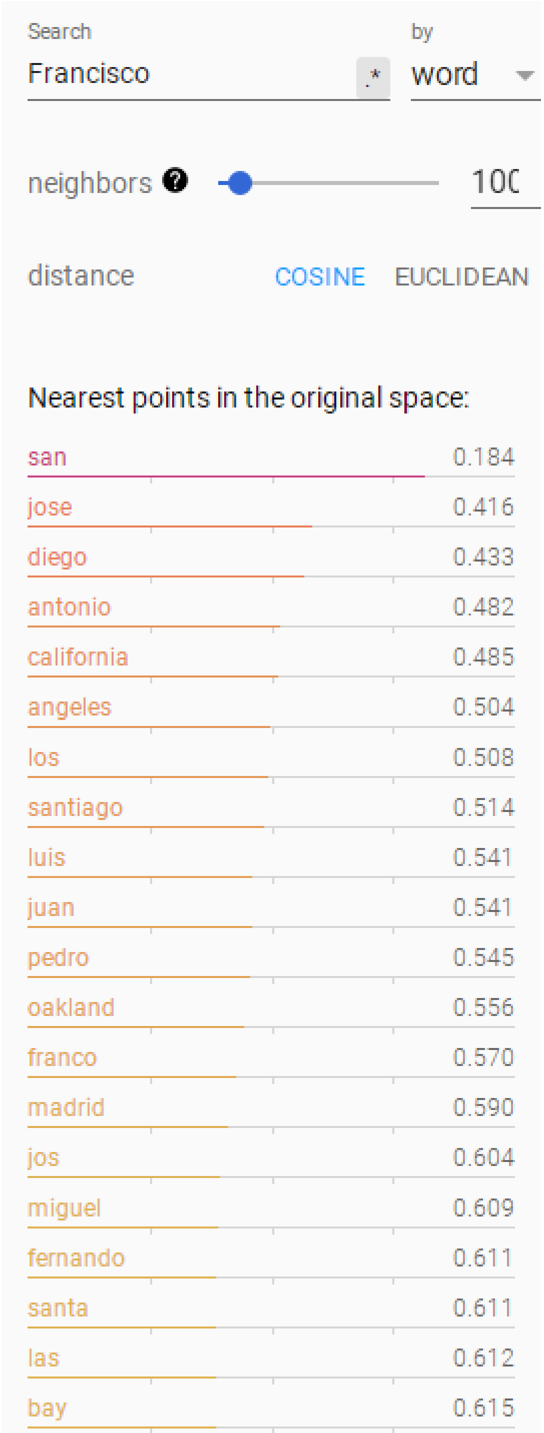
neighbors 100

distance COSINE EUCLIDEAN

Nearest points in the original space:

nerve	0.559
neural	0.586
cells	0.601
brain	0.607
nervous	0.615
receptors	0.621
tissue	0.633
muscle	0.638
tissues	0.640
motor	0.648
membranes	0.656
sensory	0.663
muscles	0.671
spinal	0.673
lungs	0.678
patients	0.688
cell	0.691
nuclei	0.694
molecules	0.709
ethanol	0.711

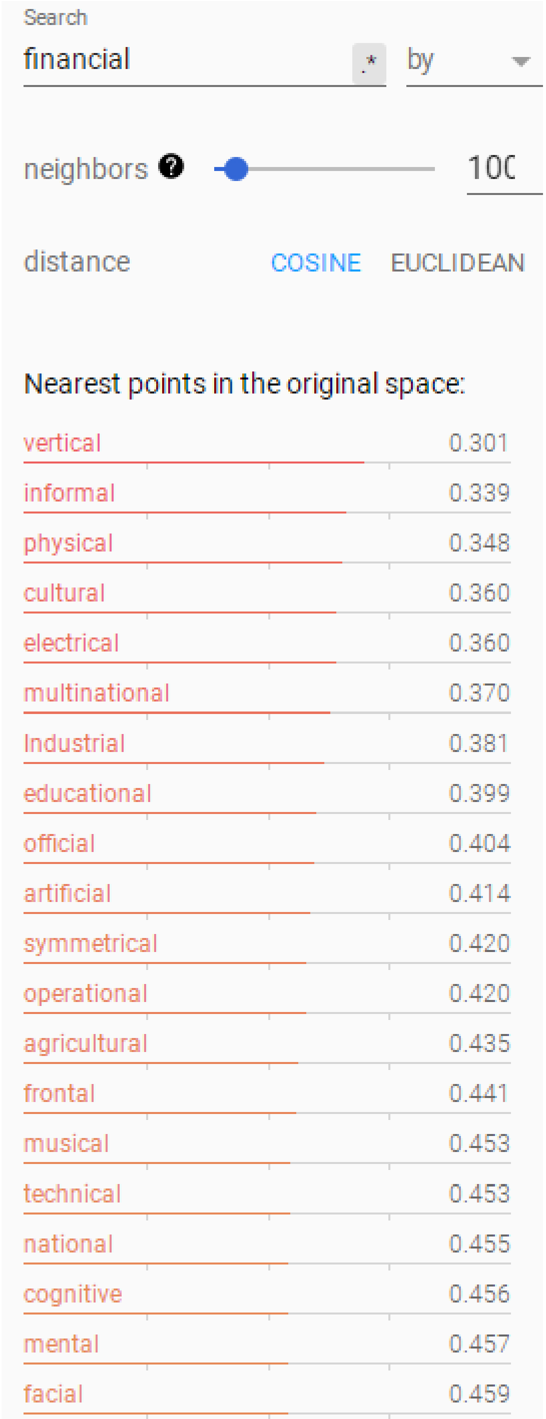

Search
expectation
by
word
neighbors
100
distance
COSINE
EUCLIDEAN
Nearest points in the original space:
norms 0.627
assumptions 0.662
policies 0.683
inflation 0.689
confidence 0.693
concerns 0.693
unemployment 0.700
rational 0.702
buying 0.706
acceptance 0.711
ideas 0.712
prices 0.717
spending 0.718
desire 0.724
price 0.726
pressures 0.729
odds 0.731
welfare 0.733
demand 0.733
dickens 0.735



Search
Francisco
by
word
neighbors
100
distance
COSINE
EUCLIDEAN
Nearest points in the original space:
san 0.184
jose 0.416
diego 0.433
antonio 0.482
california 0.485
angeles 0.504
los 0.508
santiago 0.514
luis 0.541
juan 0.541
pedro 0.545
oakland 0.556
franco 0.570
madrid 0.590
jos 0.604
miguel 0.609
fernando 0.611
santa 0.611
las 0.612
bay 0.615


Search
naturally
by
word
neighbors
100
distance
COSINE
EUCLIDEAN
Nearest points in the original space:
occurring 0.545
readily 0.614
humans 0.618
arise 0.621
easily 0.629
natural 0.630
stable 0.650
occurrence 0.657
synthetic 0.665
slowly 0.666
primitive 0.667
compounds 0.668
biological 0.673
ions 0.675
wherever 0.678
properly 0.679
inherently 0.686
rare 0.687
selective 0.688
mankind 0.689

(ii)


Search
financial
.*
by
neighbors
100
distance
COSINE
EUCLIDEAN
Nearest points in the original space:
vertical 0.301
informal 0.339
physical 0.348
cultural 0.360
electrical 0.360
multinational 0.370
Industrial 0.381
educational 0.399
official 0.404
artificial 0.414
symmetrical 0.420
operational 0.420
agricultural 0.435
frontal 0.441
musical 0.453
technical 0.453
national 0.455
cognitive 0.456
mental 0.457
facial 0.459

Search

neuron .* by

neighbors 100

distance COSINE EUCLIDEAN

Nearest points in the original space:

Newton	0.354
George	0.383
NBA	0.404
Delhi	0.415
golden	0.421
person	0.421
Google	0.427
Virgin	0.428
folk	0.430
garden	0.440
monkeys	0.447
Florida	0.450
carbon	0.451
neurons	0.453
gun	0.455
Martin	0.456
silicon	0.457
monster	0.472
robot	0.473
environments	0.475

Search

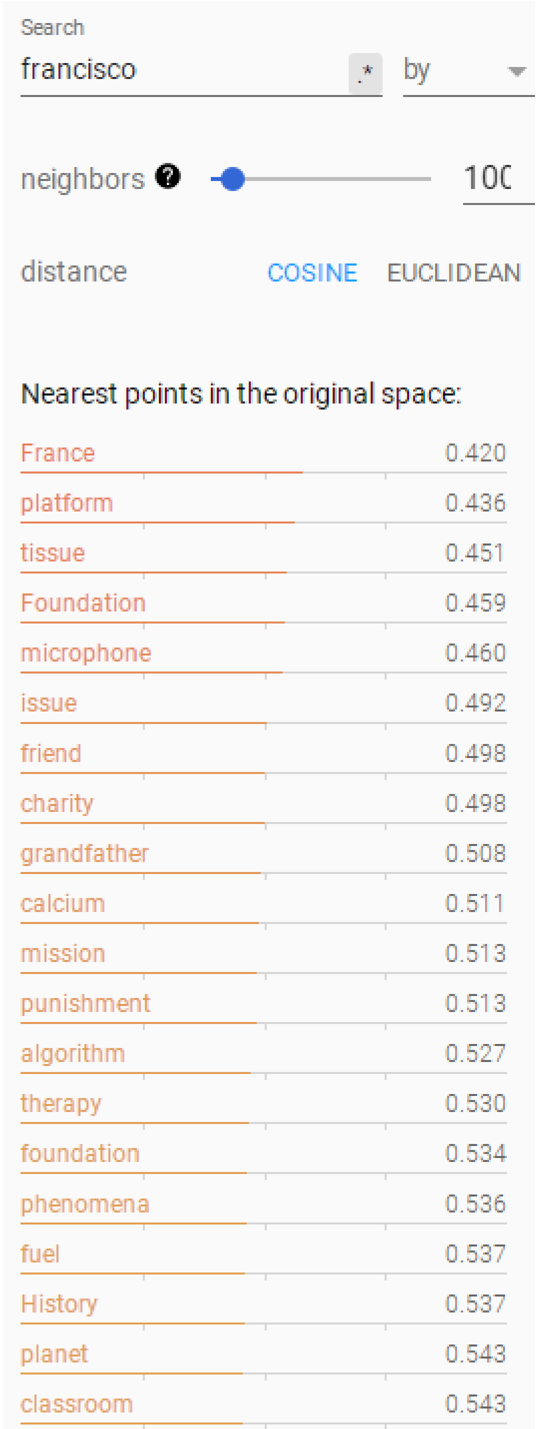
expectation .* by

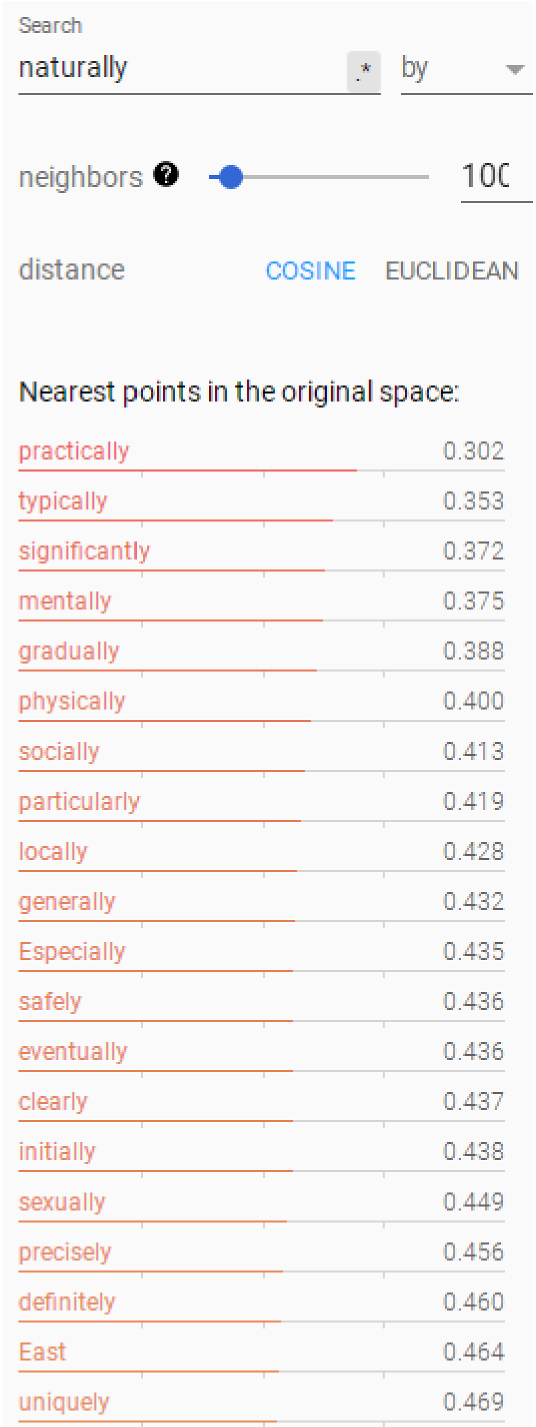
neighbors 100

distance COSINE EUCLIDEAN

Nearest points in the original space:

exception	0.389
indication	0.405
integration	0.405
separation	0.429
expected	0.473
definition	0.499
expectations	0.505
expertise	0.506
expedition	0.508
expectancy	0.508
demonstration	0.512
exercise	0.515
inspiration	0.517
animation	0.523
generation	0.524
infrastructure	0.530
recognition	0.532
imagination	0.541
tragedy	0.551
concentration	0.558


Search
francisco
.*
by
neighbors
100
distance
COSINE
EUCLIDEAN
Nearest points in the original space:
France 0.420
platform 0.436
tissue 0.451
Foundation 0.459
microphone 0.460
issue 0.492
friend 0.498
charity 0.498
grandfather 0.508
calcium 0.511
mission 0.513
punishment 0.513
algorithm 0.527
therapy 0.530
foundation 0.534
phenomena 0.536
fuel 0.537
History 0.537
planet 0.543
classroom 0.543



(iii)

In Word2Vec, words those show up in similar contexts will be close to each other, while in CharCNN, spelling similarity is modeled.

For instance, 'neuron' is near 'Newton' in CharCNN. They are nothing but similar in spelling. In Word2Vec it's near 'nerve', which makes sense and is more acceptable.

(c)

(acceptable example)

Source sentence: En uno de mis roles, trabajo con la historia de la anatoma.

Reference translation: As the one hat, I do history of anatomy.

A4 translation: In one of my **<unk>** I work with the history of anatomy.

A5 translation: In one of my **roles** -- I work with the history of the anatomy.

Comment: This is acceptable because the word 'role' in Spanish just means role in English.

Possible Explanation: The character-based model finds that the word 'roles' itself can fit well in this context and just leave it unchanged.

(incorrect example)

Source sentence: Producimos andrgenos y respondemos a los andrgenos.

Reference translation: We're making androgen, and we're responding to androgens.

A4 translation: We produce **<unk>** and we respond to the **<unk>**.

A5 translation: We produce **angers** and respond to **angers**.

Comment: This is incorrect because the word 'andrgenos' in Spanish means androgen in English.

Possible Explanation: The character-based model finds that the word 'anger' is similar to ''andrgenos', and that quite fits the sentences given the context indeed.