1. Character-based convolutional encoder for NMT (36 points)

a) Characters have less meanings than words, so the embedding size of characters is lower than that used for words.

b) character-based embedding model:

$$N_{char} = V_{char} * e_{char} + f * e_{char} * k + f + 2 * (e_{word} * e_{word} + e_{word})$$

$$f = e_{word}$$

$$N_{char} = V_{char} * e_{char} + e_{word} * e_{char} * k + e_{word} + 2 * (e_{word} * e_{word} + e_{word})$$

$$N_{char} = 96 * 50 + 256 * 50 * 5 + 256 + 2 * (256 * 256 + 256) = 200640$$

word-based lookup embedding model:

$$N_{word} = V_{word} * e_{word}$$

$$N_{word} = 50000 * 256 = 12800000$$

12800000/200640=63.8

c) In 1D convnet, learned parameters are $\mathrm{in}_{\mathrm{channel}} * \mathrm{kernel}_{\mathrm{size}} * \mathrm{out}_{\mathrm{channel}} + \mathrm{out}_{\mathrm{channel}} = 50 * 256 * 5 + 256 = 64256$. By contract, learned parameters Bi-LISTM will be: $\left(size_{input} * size_{hidden} + size_{hidden} + size_{hidden} * size_{hidden} + size_{hidden}\right) * 4 = (50 * 256 + 256 + 256 * 256 + 256) * 4 = 315392$, which is about five times bigger than number of parameters in 1D convnet.

d) Average pooling extracts all features, but it weakens the important features. Max pooling emphasizes the most important feature but loses other features.

h)

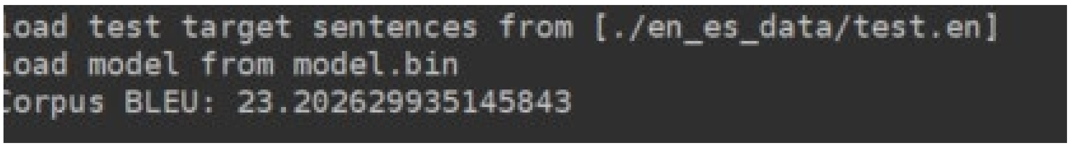
```
model = Highway(7)
x_input = torch.randn(10, 7)
x_highway = model(x_input)
assert x_input.shape == x_highway.shape
```

i)

```python
model = CNN(50, 256)  # char_e=50, word_e=256
input = torch.randn(100, 50, 30)  # N=100, char_e=50, word_len=30
output = model(input)
assert output.shape == torch.Size([100, 256])
```

2. Character-based LSTM decoder for NMT (26 points)

f)



```
Load test target sentences from [./en_es_data/test.en]
Load model from model.bin
Corpus BLEU: 23.202629935145843
```

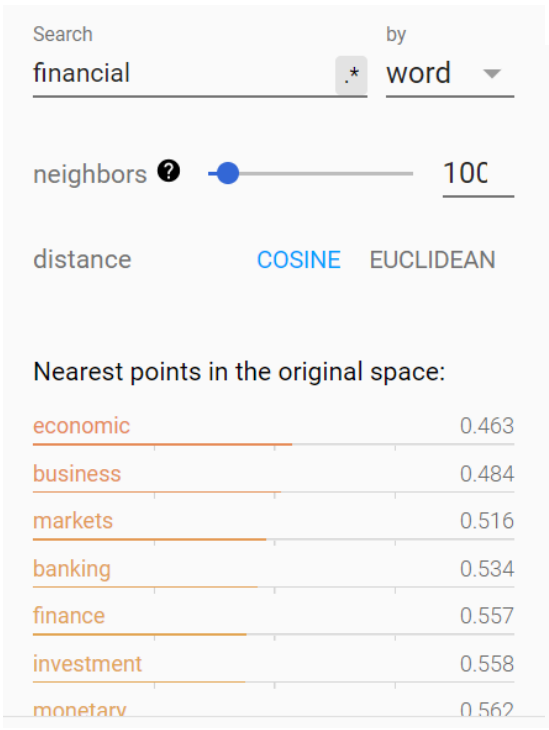
3. Analyzing NMT Systems (8 points)

a) “traducir” and “traduce” are in the word-vocabulary.

In word-based NMT, if a word is not in the word-vocabulary, it will be defined as “unk”, so the embeddings of these words are undifferentiated.

In character-aware NMT, we do character-level embedding for the out-of-vocabulary words. In this way, words not in the vocabulary will be well embedded.

b) 1.



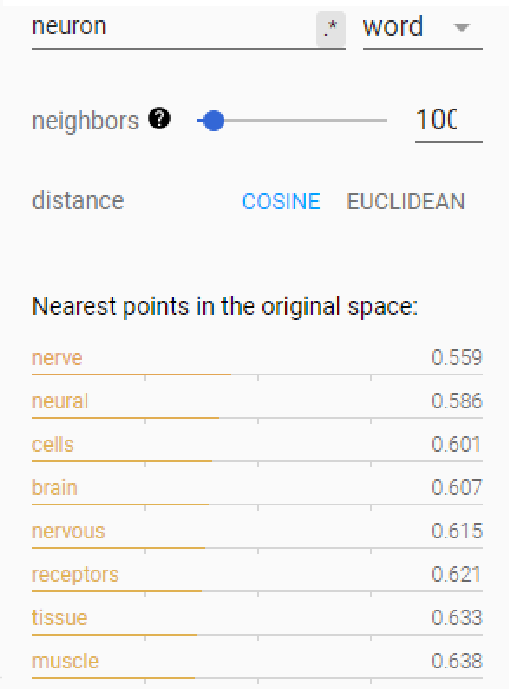
Search: financial .* by: word

neighbors ❓ 10C

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



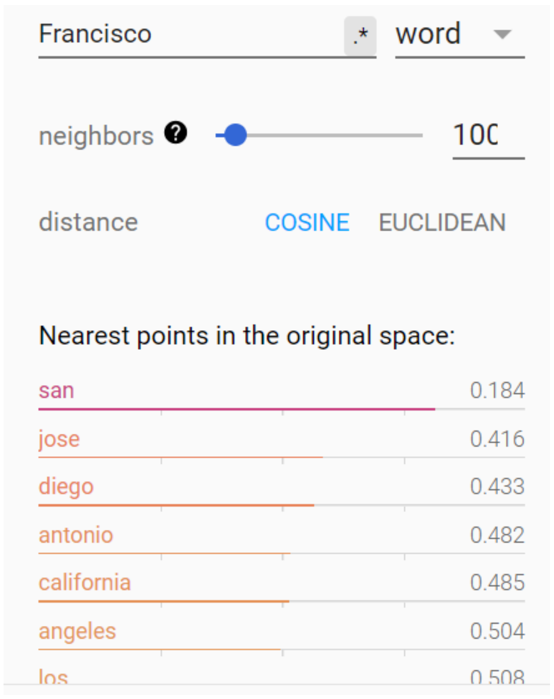
neuron .* word

neighbors ❓ 10C

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



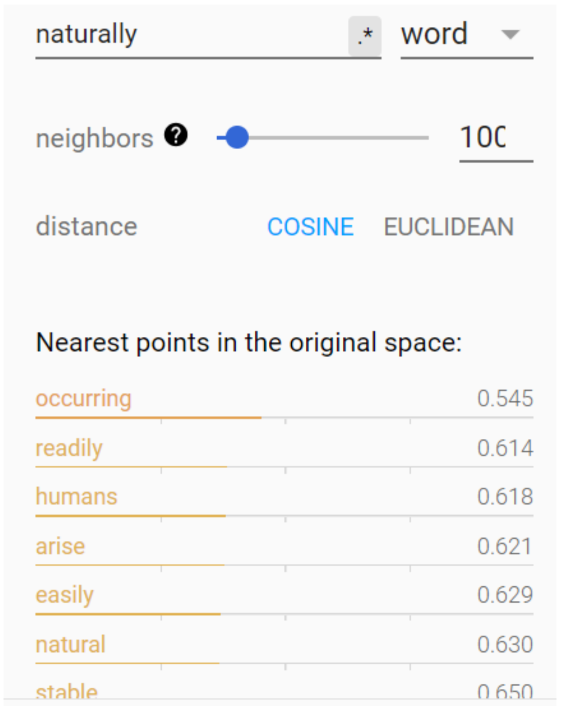
Francisco .* word

neighbors ❓ 10C

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



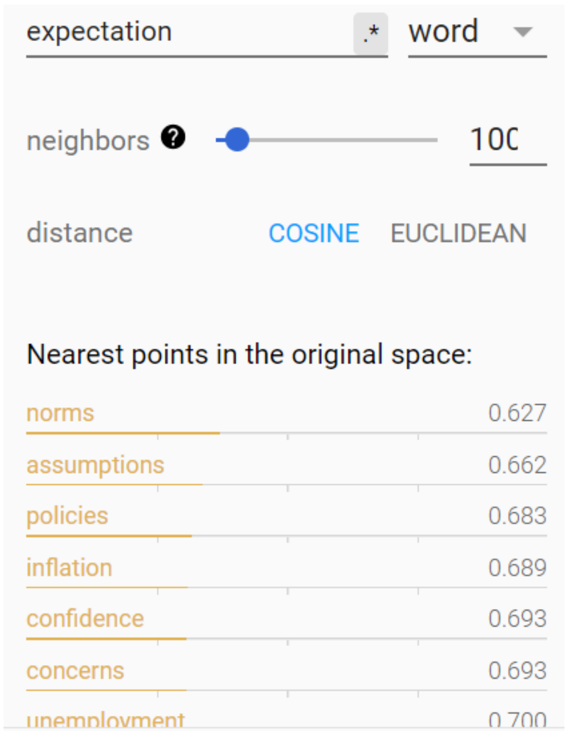
naturally .* word

neighbors ❓ 10C

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



expectation .* word

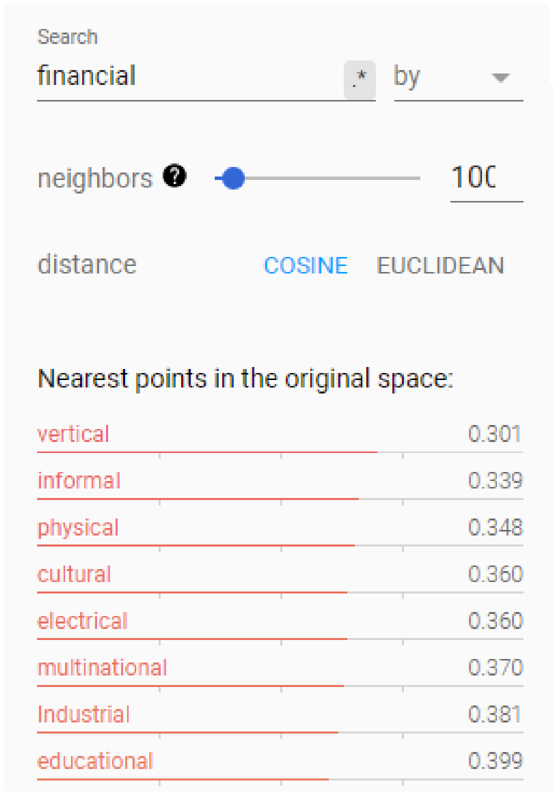
neighbors ❓ 10

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

2)



Search

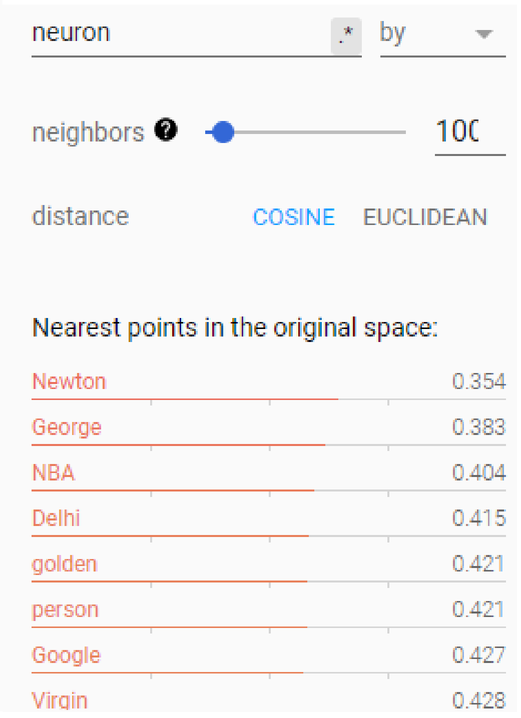
financial .* by

neighbors ❓ 10

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



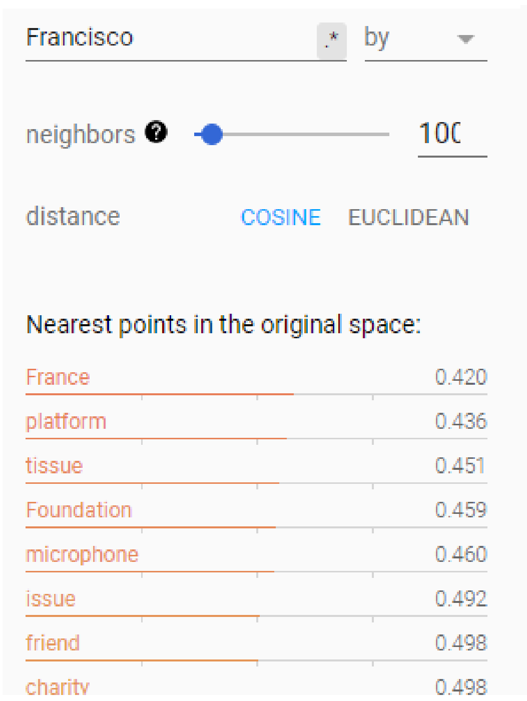
neuron .* by

neighbors ❓ 10

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



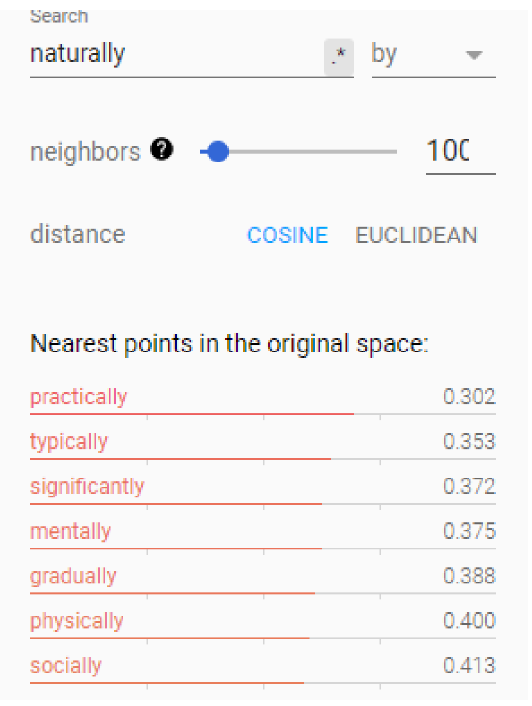
Francisco

neighbors 10(

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



Search

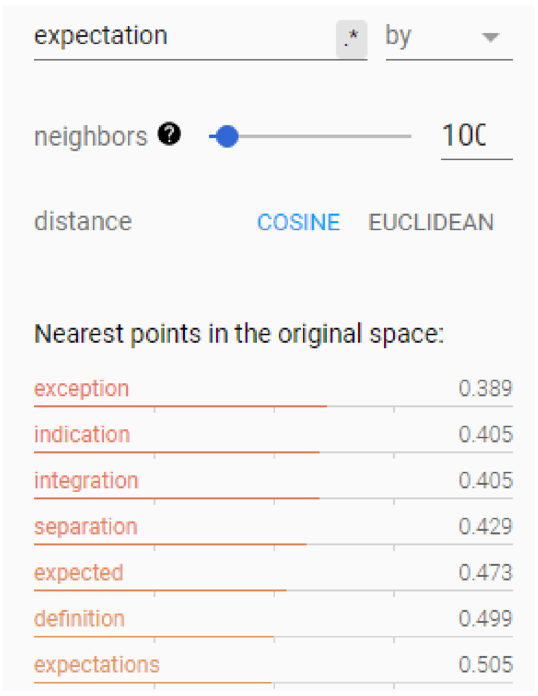
naturally

neighbors 10(

distance COSINE EUCLIDEAN

Nearest points in the original space:

practically	0.302
typically	0.353
significantly	0.372
mentally	0.375
gradually	0.388
physically	0.400
socially	0.413



expectation

neighbors 10(

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

3)

Word2Vec models semantic meanings of words. Words with similar meanings are close in the embedding space.

CharCNN models spelling of words. Words with similar spellings are close in the embedding space.

It is because Word2Vec focuses on the meaning of words, while CharCNN is based on the characters of words, and it focuses less on the semantic information of words.

c)

Spanish source sentence: As, mientras tratamos de crear una unin ms perfecta, pensemos en qu hacemos los unos por los otros.

Reference English translation: So that as we the people try to create a more perfect union, we're thinking about what we do for each other.

Translations from A4: So as we try to create a more perfect <unk> think about what we do for each other.

Character-based translation: So as we try to create a more perfect Union, we think about what we do for each other.

This is an acceptable example. character-based model can generate the word from characters correctly.

Spanish source sentence: Lunes: el color es energtico.

Reference English translation: Monday: Color is powerful.

Translations from A4: <unk> color is power.

Character-based translation: Long, the color is energy.

This is an incorrect example. “long” and “Monday” is similar in the character level, so the model makes a mistake.