## **Analyzing Word Embeddings Learned by the LSTM Based Language Model**

## LSTM Model Embedding Size 300, Hidden Size 300, Dropout 0.3

```
In [1]: import os
        import torch
        import dill as pickle
        import warnings
        import matplotlib.pyplot as plt
        from sklearn.manifold import TSNE
        warnings.filterwarnings('ignore')
In [4]: def cosine similarity(emb matrix, device=torch.device('cpu')):
            compute cosine similarity between tokens
            cosine similarity between vecs a and b = (a \cdot b) / |a||b|
            :param emb_matrix: (torch.tensor) embedding matrix (vocab_size x emsize)
            :param device: device (default: cpu)
            :return: (torch.tensor) cosine similarity matrix (vocab size x vocab size)
            with torch.no grad():
                # transfer emb matrix to device
                emb_matrix = emb_matrix.to(device)
                # compute magnitude of embedding vectors
                magnitudes = emb matrix.pow(2).sum(dim=1).sqrt().unsqueeze(0)
                # compute denominators matrix
                denominators = torch.mm(magnitudes.t(), magnitudes)
                # compute cosine similarity matrix
                 sim matrix = torch.mm(emb matrix, emb matrix.t()) / denominators
            return sim_matrix
```

```
In [5]: # set device
  device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
  print('device:', device)
```

device: cpu

```
In [6]: # Load model
         with open(os.path.join('models', 'lstm_emsize300_dropout0.3_tied.pt'),
                    'rb') as f:
             model = torch.load(f, map location=device)
         print('model loaded')
         model loaded
In [7]: # print model
         print(model)
         RNNModel(
           (drop): Dropout(p=0.3, inplace=False)
           (encoder): Embedding(33278, 300)
           (rnn): LSTM(300, 300, num_layers=2, dropout=0.3)
           (decoder): Linear(in features=300, out features=33278, bias=True)
         )
In [8]:
         cosine sim matrix = cosine similarity(emb matrix=model.encoder.weight, device=
         device)
         print('computed cosine similarity matrix')
         computed cosine similarity matrix
In [9]: | cosine_sim_matrix.shape
Out[9]: torch.Size([33278, 33278])
In [10]: # Load corpus
         with open(os.path.join('data', 'wikitext-2', 'corpus.pkl'), 'rb') as f:
             corpus = pickle.load(f)
In [11]:
         # get dictionary
         dictionary = corpus.dictionary
         assert len(dictionary) == len(cosine_sim_matrix)
```

```
In [12]: words = [
             "while",
             "five",
             "some",
             "its",
             "I",
             "by",
             "after",
             "with",
             "experience",
             "institute",
             "writers",
             "Paris",
             "engineering",
             "Bill",
             "not",
             "most",
             "American",
             "its",
             "four",
             "state",
             "operating",
             "BBC",
             "account",
             "accepted",
             "construction"
         ]
```

```
In [13]: # closest and furthest words
for word in words:
    word_id = dictionary.word2idx[word]
    _, sim_word_ids = torch.topk(cosine_sim_matrix[word_id], 11)
    _, dissim_word_ids = torch.topk(-cosine_sim_matrix[word_id], 10)
    print(f'word: {word}')
    print('closest words: ', end='')
    for wid in sim_word_ids[1:]: # the most closest is always the word itself
        print(f'{dictionary.idx2word[wid]} ', end='')
    print('\nfurthest words: ', end='')
    for wid in dissim_word_ids:
        print(f'{dictionary.idx2word[wid]} ', end='')
    print('\n')
```

word: while

closest words: although whilst whereas before when but after though despite i

f

furthest words: Twentieth 19th- Trials Trans Pozières FROG Rushie ASTRA Welch

Establishment

word: five

closest words: three six four seven ten two eight nine sixteen thirteen

furthest words: Claims Suggesting Listed Text opines Experiments chieftain su

rmised Bhopali translator

word: some

closest words: many several any neither various much multiple numerous none b

oth

furthest words: Lounge Ladies Crab dolmen Pampas Associate Gallimard Megasto

re Angels

word: its

closest words: their his the her an a my our some several

furthest words: Role quarrel However Consequently Nevertheless Except Essenti

ally Unfortunately hitter admirer

word: I

closest words: we We you You Who II never he [ What

furthest words: contaminated dealings taxing embellished booking Oriental Ter

rence detecting Kilburn theologian

word: by

closest words: from via under against when with through at after in

furthest words: IBC Bread Papal Uttam overture Shark Pick Saddle Resurrection

Concession

word: after

closest words: before when during while despite upon where if through under furthest words: Rolls Rushie Elf Fourteenth Concession Enriquillo Hurri Bread

Distribution Kishon

word: with

closest words: for alongside under through by via when featuring against with

out

furthest words: Concession IBC Pick Bread Leningrad Resurrection Fourteenth S

urvivor Gentleman Enriquillo

word: experience

closest words: influence impact change offer appeal effect freedom act challe

nge deal

furthest words: Arba R1 Shan Echave Von Milne Else Crac 1754 `Abdu

word: institute

closest words: observatory municipality constituency synagogue referendum ins

titution Campus broadcaster museum newspaper

furthest words: sprung annoying pretty Vandernoot nice cheating averted someb

ody yourself McCoy

word: writers

closest words: artists actors producers musicians poets commentators actress

author scholars characters

furthest words: Crépon Shuswap Pocantico 0300 Northam VT Allegheny Monash Pug et powdered

word: Paris

closest words: Argentina London Rome Berlin Wiedensahl Vienna Pakistan Brusse ls Minneapolis Switzerland

furthest words: analyzed evaluated emitted compounds ratios atoms variation trioxide formulated manufactured

word: engineering

closest words: licensing fitness physics engineer alternative training health care reconnaissance telecommunications piston

furthest words: Unwilling Towards hath Months Notwithstanding Wheel Attempting exposes Were recounts

word: Bill

closest words: David Mike Matt Harry Chris Alex Ron Josh Bobby Rick

furthest words: densely economically amenable fledged hewn visibly extremes c linically endemic unstable

word: not

closest words: still never already probably almost sometimes hardly perhaps o ccasionally clearly

furthest words: Plata Borowski Harney Consequence Universities 1230s Banadir Economics Prow Statue

word: most

closest words: many best more greatest much some several highest less highly furthest words: Cadw Bartov Boulez leak Paddywack 'honneur Jarrah Megastore i njunction

word: American

closest words: Indian British Australian English African French Canadian Italian German Spanish

furthest words: resorted tends evenly conform strove flocked pretends threate ns alludes conspired

word: its

closest words: their his the her an a my our some several

furthest words: Role quarrel However Consequently Nevertheless Except Essentially Unfortunately hitter admirer

word: four

closest words: three six five two seven eight nine ten twelve thirteen furthest words: Claims Suggesting Text Listed Experiments retorts Lietuva sur mised translator Osório

word: state

closest words: city county region country government town community village n ation U.S.

furthest words: Gurevich Ledden Shum 'Dell Florencio fumbled Jabbar Flatts Zo mbies Kaida

word: operating

closest words: stationed working flying flown deployed firing digging serving mounted available

furthest words: Inti Upset Ladies heaven Cheke Bateman Miranic vanity Urn Val

## entine

word: BBC

closest words: ITV iTunes TLC UPN HBO DVD Steeltown Sony RCA NME

furthest words: distinguishable oxidizes preyed outflanked decimated clumps s

mothered taels dragging greyish

word: account

closest words: act behalf example version view issue display use work conditi

on

furthest words: Vista Henrik Bethel Nuestra Yves Sha Cao dos Uttar Gran

word: accepted

closest words: rejected retained taught assumed discussed initiated addressed

visited avoided studied

furthest words: Brower crustaceans Amtrak Plugge Wilmington Queenstown Rowen

Springfield Rarely Minneapolis

word: construction

closest words: reconstruction completion development restoration ownership pr

oduction approval demolition conservation destruction

furthest words: Hiyorigawa Papa Nd4 Fry 2.Nf3 Sussman Squirtle Cory Kōenji Mo

lina

In [27]: dictionary.word2idx['<unk>']

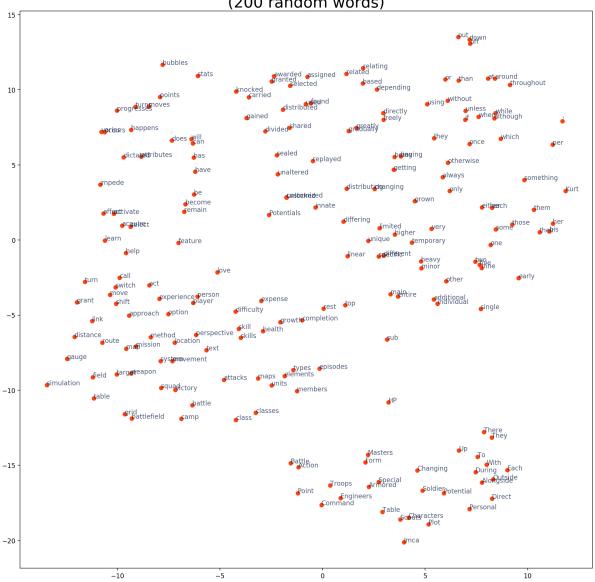
Out[27]: 9

```
In [37]: # TSNE
    emb_tsne = TSNE(metric='cosine').fit_transform(model.encoder.weight.detach().c
    pu().numpy()[200:400, :])

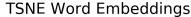
# plot
    plt.figure(figsize=(16, 16), dpi=160)
    for idx in range(200):
        plt.scatter(*emb_tsne[idx, :], color='xkcd:orangered')
        plt.annotate(dictionary.idx2word[200 + idx], (emb_tsne[idx, 0], emb_tsne[idx, 1]), alpha=0.7, color='xkcd:navy')

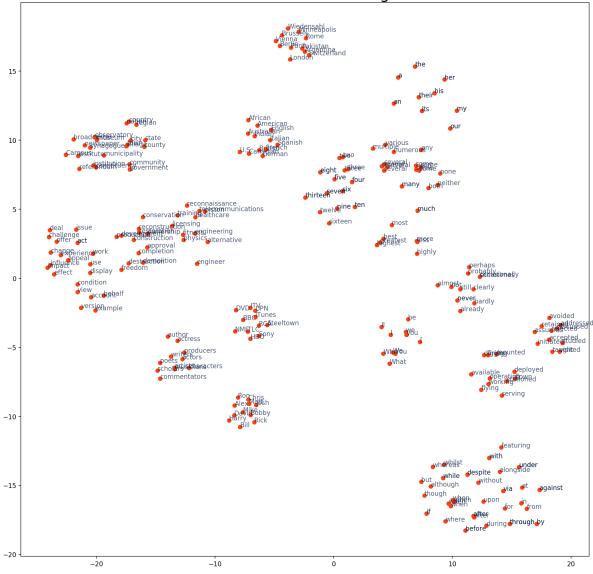
    plt.title('TSNE Word Embeddings\n(200 random words)', fontsize=24)
    plt.show()
```

TSNE Word Embeddings (200 random words)



```
In [38]:
         # selected words
         word ids = []
         for word in words:
             word id = dictionary.word2idx[word]
             _, sim_word_ids = torch.topk(cosine_sim_matrix[word_id], 11)
             word_ids.extend(sim_word_ids.tolist())
         # TSNE
         emb_tsne = TSNE(metric='cosine').fit_transform(model.encoder.weight.detach().c
         pu().numpy()[word_ids, :])
         # plot
         plt.figure(figsize=(16, 16), dpi=160)
         for i, idx in enumerate(word ids):
             plt.scatter(*emb_tsne[i, :], color='xkcd:orangered')
             plt.annotate(dictionary.idx2word[idx], (emb_tsne[i, 0], emb_tsne[i, 1]), a
         lpha=0.7, color='xkcd:navy')
         plt.title('TSNE Word Embeddings', fontsize=24)
         plt.show()
```

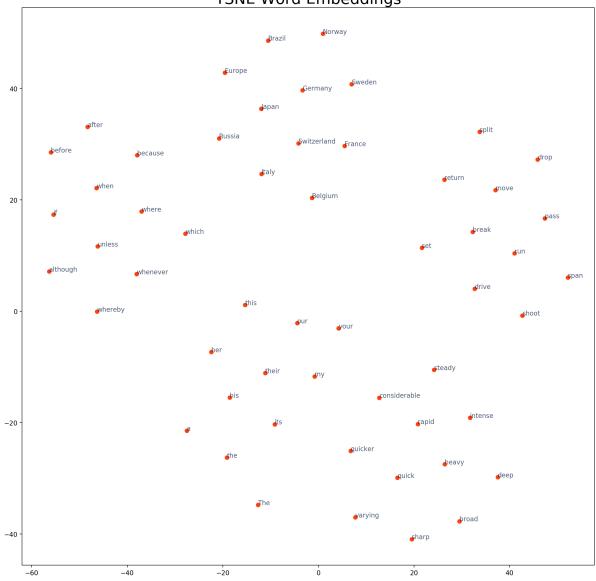




```
In [45]: | words = [
             "the", "run", "Germany", "where", "quick"
In [46]:
         # closest and furthest words
         for word in words:
             word id = dictionary.word2idx[word]
             _, sim_word_ids = torch.topk(cosine_sim_matrix[word_id], 11)
             _, dissim_word_ids = torch.topk(-cosine_sim_matrix[word_id], 10)
             print(f'word: {word}')
             print('closest words: ', end='')
             for wid in sim_word_ids[1:]: # the most closest is always the word itself
                 print(f'{dictionary.idx2word[wid]} ', end='')
             print('\nfurthest words: ', end='')
             for wid in dissim word ids:
                 print(f'{dictionary.idx2word[wid]} ', end='')
             print('\n')
         word: the
         closest words: a their its his this our her The my your
         furthest words: group phenomenon interiors pair programme Battles methods ass
         ortment matter multitude
         word: run
         closest words: pass break move drive drop return set shoot span split
         furthest words: Chang Potts atrox Thutmose Immaculate Zonghan Clancy Porvenir
         Terkel Escorial
         word: Germany
         closest words: Italy France Japan Switzerland Sweden Norway Russia Belgium Eu
         rope Brazil
         furthest words: neckbreaker ferocious tumulus mat spouting joystick coiled ti
         cking button infraction
         word: where
         closest words: when because before unless after although whereby whenever if
         furthest words: Schadla twentieth Rolls 20th micro Gay Welch Twentieth 555th
         fiftieth
         word: quick
         closest words: rapid broad quicker heavy varying steady sharp intense deep co
         furthest words: Campus Bhopali Gallimard Douglass Pliny Pier Factory Ravidass
         Warne Donoghue
```

```
In [47]:
         # selected words
         word ids = []
         for word in words:
             word id = dictionary.word2idx[word]
             _, sim_word_ids = torch.topk(cosine_sim_matrix[word_id], 11)
             word_ids.extend(sim_word_ids.tolist())
         # TSNE
         emb_tsne = TSNE(metric='cosine').fit_transform(model.encoder.weight.detach().c
         pu().numpy()[word_ids, :])
         # plot
         plt.figure(figsize=(16, 16), dpi=160)
         for i, idx in enumerate(word ids):
             plt.scatter(*emb_tsne[i, :], color='xkcd:orangered')
             plt.annotate(dictionary.idx2word[idx], (emb_tsne[i, 0], emb_tsne[i, 1]), a
         lpha=0.7, color='xkcd:navy')
         plt.title('TSNE Word Embeddings', fontsize=24)
         plt.show()
```





In [ ]:			