Negative_Sampling_Exercise

April 2, 2020

1 Skip-gram Word2Vec

In this notebook, I'll lead you through using PyTorch to implement the Word2Vec algorithm using the skip-gram architecture. By implementing this, you'll learn about embedding words for use in natural language processing. This will come in handy when dealing with things like machine translation.

1.1 Readings

Here are the resources I used to build this notebook. I suggest reading these either beforehand or while you're working on this material.

- A really good conceptual overview of Word2Vec from Chris McCormick
- First Word2Vec paper from Mikolov et al.
- Neural Information Processing Systems, paper with improvements for Word2Vec also from Mikolov et al.

1.2 Word embeddings

When you're dealing with words in text, you end up with tens of thousands of word classes to analyze; one for each word in a vocabulary. Trying to one-hot encode these words is massively inefficient because most values in a one-hot vector will be set to zero. So, the matrix multiplication that happens in between a one-hot input vector and a first, hidden layer will result in mostly zero-valued hidden outputs.

To solve this problem and greatly increase the efficiency of our networks, we use what are called **embeddings**. Embeddings are just a fully connected layer like you've seen before. We call this layer the embedding layer and the weights are embedding weights. We skip the multiplication into the embedding layer by instead directly grabbing the hidden layer values from the weight matrix. We can do this because the multiplication of a one-hot encoded vector with a matrix returns the row of the matrix corresponding the index of the "on" input unit.

Instead of doing the matrix multiplication, we use the weight matrix as a lookup table. We encode the words as integers, for example "heart" is encoded as 958, "mind" as 18094. Then to get hidden layer values for "heart", you just take the 958th row of the embedding matrix. This process is called an **embedding lookup** and the number of hidden units is the **embedding dimension**.

There is nothing magical going on here. The embedding lookup table is just a weight matrix. The embedding layer is just a hidden layer. The lookup is just a shortcut for the matrix multiplication. The lookup table is trained just like any weight matrix.

Embeddings aren't only used for words of course. You can use them for any model where you have a massive number of classes. A particular type of model called **Word2Vec** uses the embedding layer to find vector representations of words that contain semantic meaning.

1.3 Word2Vec

The Word2Vec algorithm finds much more efficient representations by finding vectors that represent the words. These vectors also contain semantic information about the words.

Words that show up in similar **contexts**, such as "coffee", "tea", and "water" will have vectors near each other. Different words will be further away from one another, and relationships can be represented by distance in vector space.

There are two architectures for implementing Word2Vec: >* CBOW (Continuous Bag-Of-Words) and * Skip-gram

In this implementation, we'll be using the **skip-gram architecture** with **negative sampling** because it performs better than CBOW and trains faster with negative sampling. Here, we pass in a word and try to predict the words surrounding it in the text. In this way, we can train the network to learn representations for words that show up in similar contexts.

1.4 Loading Data

Next, we'll ask you to load in data and place it in the `data` directory

1. Load the text8 dataset; a file of cleaned up *Wikipedia article text* from Matt Mahoney.
2. Place that data in the `data` folder in the home directory.
3. Then you can extract it and delete the archive, zip file to save storage space.

After following these steps, you should have one file in your data directory: `data/text8`.

```
In [1]: # read in the extracted text file
        with open('data/text8') as f:
            text = f.read()

        # print out the first 100 characters
        print(text[:100])
```

```
 anarchism originated as a term of abuse first used against early working class radicals includi
```

1.5 Pre-processing

Here I'm fixing up the text to make training easier. This comes from the `utils.py` file. The `preprocess` function does a few things: >* It converts any punctuation into tokens, so a period is changed to `<PERIOD>`. In this data set, there aren't any periods, but it will help in other NLP problems. * It removes all words that show up five or *fewer* times in the dataset. This will greatly reduce issues due to noise in the data and improve the quality of the vector representations. * It returns a list of words in the text.

This may take a few seconds to run, since our text file is quite large. If you want to write your own functions for this stuff, go for it!

```
In [2]: import utils

        # get list of words
        words = utils.preprocess(text)
        print(words[:30])
```

```
['anarchism', 'originated', 'as', 'a', 'term', 'of', 'abuse', 'first', 'used', 'against', 'early
```

```
In [3]: # print some stats about this word data
        print("Total words in text: {}".format(len(words)))
        print("Unique words: {}".format(len(set(words)))) # `set` removes any duplicate words
```

```
Total words in text: 16680599
Unique words: 63641
```

1.5.1 Dictionaries

Next, I'm creating two dictionaries to convert words to integers and back again (integers to words). This is again done with a function in the `utils.py` file. `create_lookup_tables` takes in a list of words in a text and returns two dictionaries. >* The integers are assigned in descending frequency order, so the most frequent word ("the") is given the integer 0 and the next most frequent is 1, and so on.

Once we have our dictionaries, the words are converted to integers and stored in the list `int_words`.

```
In [4]: vocab_to_int, int_to_vocab = utils.create_lookup_tables(words)
        int_words = [vocab_to_int[word] for word in words]


        print(len(int_to_vocab))
        print(len(int_words))
        print(int_words[:30])
```

```
63641
16680599
[5233, 3080, 11, 5, 194, 1, 3133, 45, 58, 155, 127, 741, 476, 10571, 133, 0, 27349, 1, 0, 102, 8
```

1.6 Subsampling

Words that show up often such as "the", "of", and "for" don't provide much context to the nearby words. If we discard some of them, we can remove some of the noise from our data and in return get faster training and better representations. This process is called subsampling by Mikolov. For each word w_i in the training set, we'll discard it with probability given by

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

where t is a threshold parameter and $f(w_i)$ is the frequency of word w_i in the total dataset.

> Implement subsampling for the words in `int_words`. That is, go through `int_words` and discard each word given the probablility $P(w_i)$ shown above. Note that $P(w_i)$ is the probability that a word is discarded. Assign the subsampled data to `train_words`.

In [5]:
```
from collections import Counter
import random
import numpy as np

threshold = 1e-5
word_counts = Counter(int_words)
print(len(word_counts))
print(list(word_counts.items())[0])  # dictionary of int_words, how many times they appe

total_count = len(int_words)
freqs = {word: count/total_count for word, count in word_counts.items()}
p_drop = {word: 1 - np.sqrt(threshold/freqs[word]) for word in word_counts}
# discard some frequent words, according to the subsampling equation
# create a new list of words for training
train_words = [word for word in int_words if random.random() < (1 - p_drop[word])]

print(train_words[:30])

print('\n\n\n')
print(len(train_words))
```

```
63641
(5233, 303)
[5233, 3133, 741, 10571, 27349, 15067, 58112, 150, 190, 10712, 1324, 2731, 3672, 708, 371, 40, 3




4628080
```

1.7 Making batches

Now that our data is in good shape, we need to get it into the proper form to pass it into our network. With the skip-gram architecture, for each word in the text, we want to define a surrounding *context* and grab all the words in a window around that word, with size C.

From Mikolov et al.:

"Since the more distant words are usually less related to the current word than those close to it, we give less weight to the distant words by sampling less from those words in our training examples... If we choose $C = 5$, for each training word we will select randomly a number R in range $[1 : C]$, and then use R words from history and R words from the future of the current word as correct labels."

> **Exercise:** Implement a function `get_target` that receives a list of words, an index, and a window size, then returns a list of words in the window around the index. Make sure to use the algorithm described above, where you chose a random number of words to from the window.

Say, we have an input and we're interested in the idx=2 token, `741`:

```
[5233, 58, 741, 10571, 27349, 0, 15067, 58112, 3580, 58, 10712]
```

For `R=2`, `get_target` should return a list of four values:

```
[5233, 58, 10571, 27349]
```

```
In [6]: def get_target(words, idx, window_size=5):
            ''' Get a list of words in a window around an index. '''

            R = np.random.randint(1, window_size+1)
            start = idx - R if (idx - R) > 0 else 0
            stop = idx + R
            target_words = words[start:idx] + words[idx+1:stop+1]

            return list(target_words)
```

```
In [7]: # test your code!

        # run this cell multiple times to check for random window selection
        int_text = [i for i in range(10)]
        print('Input: ', int_text)
        idx=5 # word index of interest

        target = get_target(int_text, idx=idx, window_size=5)
        print('Target: ', target)  # you should get some indices around the idx
```

```
Input:  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Target:  [4, 6]
```

1.7.1 Generating Batches

Here's a generator function that returns batches of input and target data for our model, using the `get_target` function from above. The idea is that it grabs `batch_size` words from a words list. Then for each of those batches, it gets the target words in a window.

```
In [8]: def get_batches(words, batch_size, window_size=5):
            ''' Create a generator of word batches as a tuple (inputs, targets) '''

            n_batches = len(words)//batch_size

            # only full batches
            words = words[:n_batches*batch_size]

            for idx in range(0, len(words), batch_size):
                x, y = [], []
                batch = words[idx:idx+batch_size]
                for ii in range(len(batch)):
                    batch_x = batch[ii]
                    batch_y = get_target(batch, ii, window_size)
                    y.extend(batch_y)
                    x.extend([batch_x]*len(batch_y))
                yield x, y
```

```
In [9]: int_text = [i for i in range(20)]
        x,y = next(get_batches(int_text, batch_size=4, window_size=5))

        print('x\n', x)
        print('y\n', y)
```

```
x
 [0, 0, 0, 1, 1, 1, 2, 2, 2, 3]
y
 [1, 2, 3, 0, 2, 3, 0, 1, 3, 2]
```

1.8 Validation

Here, I'm creating a function that will help us observe our model as it learns. We're going to choose a few common words and few uncommon words. Then, we'll print out the closest words to them using the cosine similarity:

$$\text{similarity} = \cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}$$

We can encode the validation words as vectors $\vec{a}$ using the embedding table, then calculate the similarity with each word vector $\vec{b}$ in the embedding table. With the similarities, we can print out the validation words and words in our embedding table semantically similar to those words. It's a nice way to check that our embedding table is grouping together words with similar semantic meanings.

```
In [10]: def cosine_similarity(embedding, valid_size=16, valid_window=100, device='cpu'):
             """ Returns the cosine similarity of validation words with words in the embedding m
                 Here, embedding should be a PyTorch embedding module.
             """

             # Here we're calculating the cosine similarity between some random words and
             # our embedding vectors. With the similarities, we can look at what words are
             # close to our random words.

             # sim = (a . b) / |a||b|

             embed_vectors = embedding.weight

             # magnitude of embedding vectors, |b|
             magnitudes = embed_vectors.pow(2).sum(dim=1).sqrt().unsqueeze(0)

             # pick N words from our ranges (0,window) and (1000,1000+window). lower id implies
             valid_examples = np.array(random.sample(range(valid_window), valid_size//2))
             valid_examples = np.append(valid_examples,
                                        random.sample(range(1000,1000+valid_window), valid_size/
             valid_examples = torch.LongTensor(valid_examples).to(device)

             valid_vectors = embedding(valid_examples)
             similarities = torch.mm(valid_vectors, embed_vectors.t())/magnitudes

             return valid_examples, similarities
```

---

2 SkipGram model

Define and train the SkipGram model. > You'll need to define an embedding layer and a final, softmax output layer.

An Embedding layer takes in a number of inputs, importantly: * **num_embeddings** – the size of the dictionary of embeddings, or how many rows you'll want in the embedding weight matrix * **embedding_dim** – the size of each embedding vector; the embedding dimension

Below is an approximate diagram of the general structure of our network.

- The input words are passed in as batches of input word tokens.
- This will go into a hidden layer of linear units (our embedding layer).
- Then, finally into a softmax output layer.

We'll use the softmax layer to make a prediction about the context words by sampling, as usual.

2.1 Negative Sampling

For every example we give the network, we train it using the output from the softmax layer. That means for each input, we're making very small changes to millions of weights even though we only have one true example. This makes training the network very inefficient. We can approximate the loss from the softmax layer by only updating a small subset of all the weights at once. We'll update the weights for the correct example, but only a small number of incorrect, or noise, examples. This is called "negative sampling".

There are two modifications we need to make. First, since we're not taking the softmax output over all the words, we're really only concerned with one output word at a time. Similar to how we use an embedding table to map the input word to the hidden layer, we can now use another embedding table to map the hidden layer to the output word. Now we have two embedding layers, one for input words and one for output words. Secondly, we use a modified loss function where we only care about the true example and a small subset of noise examples.

$$-\log \sigma \left(u_{w_O}{}^{\top} v_{w_I} \right) - \sum_i^N \mathbb{E}_{w_i \sim P_n(w)} \log \sigma \left(-u_{w_i}{}^{\top} v_{w_I} \right)$$

This is a little complicated so I'll go through it bit by bit. $u_{w_O}{}^{\top}$ is the embedding vector for our "output" target word (transposed, that's the $^{\top}$ symbol) and v_{w_I} is the embedding vector for the "input" word. Then the first term

$$\log \sigma \left(u_{w_O}{}^{\top} v_{w_I} \right)$$

says we take the log-sigmoid of the inner product of the output word vector and the input word vector. Now the second term, let's first look at

$$\sum_i^N \mathbb{E}_{w_i \sim P_n(w)}$$

This means we're going to take a sum over words w_i drawn from a noise distribution $w_i \sim P_n(w)$. The noise distribution is basically our vocabulary of words that aren't in the context of our input word. In effect, we can randomly sample words from our vocabulary to get these words. $P_n(w)$ is an arbitrary probability distribution though, which means we get to decide how to weight the words that we're sampling. This could be a uniform distribution, where we sample all words with equal probability. Or it could be according to the frequency that each word shows up in our text corpus, the unigram distribution $U(w)$. The authors found the best distribution to be $U(w)^{3/4}$, empirically.

Finally, in

$$\log \sigma \left(-u_{w_i}{}^{\top} v_{w_I} \right),$$

we take the log-sigmoid of the negated inner product of a noise vector with the input vector.

To give you an intuition for what we're doing here, remember that the sigmoid function returns a probability between 0 and 1. The first term in the loss pushes the probability that our network will predict the correct word w_O towards 1. In the second term, since we are negating the sigmoid input, we're pushing the probabilities of the noise words towards 0.

In [11]:
```
import torch
from torch import nn
import torch.optim as optim
```

In [12]:
```
class SkipGramNeg(nn.Module):
    def __init__(self, n_vocab, n_embed, noise_dist=None):
        super().__init__()

        self.n_vocab = n_vocab
        self.n_embed = n_embed
        self.noise_dist = noise_dist

        # define embedding layers for input and output words
        #Mapping from our vocab to our embedding dimension
        self.in_embed = nn.Embedding(n_vocab, n_embed)
        self.out_embed = nn.Embedding(n_vocab, n_embed)

        # Initialize both embedding tables with uniform distribution

        self.in_embed.weight.data.uniform_(-1, 1)
        self.out_embed.weight.data.uniform_(-1, 1)

    def forward_input(self, input_words):
        input_vectors = self.in_embed(input_words)
        # return input vector embeddings

        return input_vectors

    def forward_output(self, output_words):
        # return output vector embeddings
        output_vectors = self.out_embed(output_words)
        return output_vectors

    def forward_noise(self, batch_size, n_samples):
        """ Generate noise vectors with shape (batch_size, n_samples, n_embed)"""
        if self.noise_dist is None:
            # Sample words uniformly
            noise_dist = torch.ones(self.n_vocab)
        else:
            noise_dist = self.noise_dist

        # Sample words from our noise distribution
        noise_words = torch.multinomial(noise_dist,
                                        batch_size * n_samples,
                                        replacement=True)

        device = "cuda" if model.out_embed.weight.is_cuda else "cpu"
        noise_words = noise_words.to(device)

        ## TODO: get the noise embeddings
        # reshape the embeddings so that they have dims (batch_size, n_samples, n_embed
        noise_vectors = self.out_embed(noise_words).view(batch_size, n_samples, self.n_
```

```
            return noise_vectors
```

```
In [13]: class NegativeSamplingLoss(nn.Module):
             def __init__(self):
                 super().__init__()

             def forward(self, input_vectors, output_vectors, noise_vectors):

                 batch_size, embed_size = input_vectors.shape

                 # Input vectors should be a batch of column vectors
                 input_vectors = input_vectors.view(batch_size, embed_size, 1)

                 # Output vectors should be a batch of row vectors
                 output_vectors = output_vectors.view(batch_size, 1, embed_size)

                 # bmm = batch matrix multiplication
                 # correct log-sigmoid loss
                 out_loss = torch.bmm(output_vectors, input_vectors).sigmoid().log()
                 out_loss = out_loss.squeeze()

                 # incorrect log-sigmoid loss
                 noise_loss = torch.bmm(noise_vectors.neg(), input_vectors).sigmoid().log()
                 noise_loss = noise_loss.squeeze().sum(1)  # sum the losses over the sample of n

                 # negate and sum correct and noisy log-sigmoid losses
                 # return average batch loss
                 return -(out_loss + noise_loss).mean()
```

2.1.1 Training

Below is our training loop, and I recommend that you train on GPU, if available.

```
In [14]: device = 'cuda' if torch.cuda.is_available() else 'cpu'

         # Get our noise distribution
         # Using word frequencies calculated earlier in the notebook
         word_freqs = np.array(sorted(freqs.values(), reverse=True))
         unigram_dist = word_freqs/word_freqs.sum()
         noise_dist = torch.from_numpy(unigram_dist**(0.75)/np.sum(unigram_dist**(0.75)))

         # instantiating the model
         embedding_dim = 300
         model = SkipGramNeg(len(vocab_to_int), embedding_dim, noise_dist=noise_dist).to(device)

         # using the loss that we defined
         criterion = NegativeSamplingLoss()
```

```
optimizer = optim.Adam(model.parameters(), lr=0.003)

print_every = 1500
steps = 0
epochs = 5

# train for some number of epochs
for e in range(epochs):

    # get our input, target batches
    for input_words, target_words in get_batches(train_words, 512):
        steps += 1
        inputs, targets = torch.LongTensor(input_words), torch.LongTensor(target_words)
        inputs, targets = inputs.to(device), targets.to(device)

        # input, outpt, and noise vectors
        input_vectors = model.forward_input(inputs)
        output_vectors = model.forward_output(targets)
        noise_vectors = model.forward_noise(inputs.shape[0], 5)

        # negative sampling loss
        loss = criterion(input_vectors, output_vectors, noise_vectors)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        # loss stats
        if steps % print_every == 0:
            print("Epoch: {}/{}".format(e+1, epochs))
            print("Loss: ", loss.item()) # avg batch loss at this point in training
            valid_examples, valid_similarities = cosine_similarity(model.in_embed, devi
            _, closest_idxs = valid_similarities.topk(6)

            valid_examples, closest_idxs = valid_examples.to('cpu'), closest_idxs.to('c
            for ii, valid_idx in enumerate(valid_examples):
                closest_words = [int_to_vocab[idx.item()] for idx in closest_idxs[ii]][
                print(int_to_vocab[valid_idx.item()] + " | " + ', '.join(closest_words)
            print("...\n")
```

```
Epoch: 1/5
Loss:  6.880287170410156
he | bike, closing, telecommunications, young, unguided
at | teutonic, brooks, empire, still, generators
they | the, sumitomo, fresco, electors, deaths
two | of, the, in, a, plan
united | eca, merchants, speed, ricks, heraclea
who | imperium, splits, pinto, grasse, eating
```

```
into | the, fame, hybrids, bolsheviks, spreading
where | slovene, builder, inflammatory, e, present
universe | priest, dish, zero, augustinian, heavily
engine | vh, given, readily, perceptible, carbonate
mathematics | fatwa, banners, cranston, solid, caroli
scale | clean, trustworthy, untrue, dichotomy, orla
square | rolling, pedals, export, dirks, followed
units | insects, were, model, pavlov, shalt
consists | gospels, enzyme, coalition, repeat, protest
assembly | graze, lavey, contraction, ofdm, comorian
...

Epoch: 1/5
Loss:  5.129039287567139
would | as, the, been, on, ground
on | in, and, one, of, the
a | the, of, is, in, and
he | was, to, zero, the, as
were | on, in, nine, one, the
over | of, third, assist, larger, known
six | zero, the, one, two, s
had | to, the, and, he, or
existence | for, boadicea, melting, exposition, trial
bible | coefficient, walls, profiled, of, historical
channel | vanderbilt, industrialization, maneuver, had, posturing
nobel | thunderbirds, paradise, lengthy, attractive, guarantor
issue | into, grail, the, department, arf
numerous | films, worse, alliaceae, imaginative, commemorated
placed | wheels, criollos, extremely, moist, shoe
writers | such, spanning, sinai, steganography, spins
...

Epoch: 1/5
Loss:  3.9502527713775635
th | four, of, m, on, one
many | which, their, of, to, and
if | is, are, n, following, can
war | six, five, and, the, one
in | of, the, a, to, which
up | which, the, be, or, an
however | with, not, to, and, has
s | one, two, the, was, five
engineering | can, pathophysiology, adapts, blackjack, piece
pope | that, lost, to, lifetime, vc
active | friend, conscientious, guidance, phytoplankton, evolutionist
mean | emmy, trento, for, opry, or
powers | timorese, monarch, firsthand, talmud, bucaram
gold | later, terrorists, exhibition, corbett, behind
```

```
placed | wheels, deposits, extremely, over, criollos
accepted | since, favoured, deprecating, priority, called
...

Epoch: 1/5
Loss:  3.5027823448181152
known | of, the, in, by, one
it | to, the, is, which, this
th | king, century, one, later, eight
but | the, on, are, to, this
they | to, that, the, have, of
during | of, the, war, was, and
used | such, is, can, or, be
if | can, any, be, we, x
pre | brereton, empire, the, shouted, odp
accepted | deprecating, most, and, jacob, closely
resources | staves, importantly, bluesy, line, holst
existence | it, prisoners, mere, have, come
additional | system, should, emissary, term, metabolism
writers | known, one, writing, spanning, theologians
cost | stems, system, can, shadowing, these
something | programmer, some, notated, redrawn, ausonius
...

Epoch: 1/5
Loss:  3.2015912532806396
an | and, a, was, to, the
one | nine, six, four, seven, eight
than | are, is, these, it, the
history | one, four, and, nine, of
where | a, was, and, at, of
nine | one, eight, seven, six, four
had | was, he, one, his, nine
d | b, seven, american, eight, nine
cost | resulting, systems, stems, less, to
know | you, that, her, might, do
smith | st, william, brother, york, thomas
taking | he, that, put, serious, but
articles | admired, published, written, guts, pythagoreans
bbc | day, zero, october, nine, april
question | that, bring, idea, beliefs, god
marriage | he, sister, emperor, deadline, viii
...

Epoch: 1/5
Loss:  3.2606048583984375
there | because, which, not, if, being
if | we, can, function, not, functions
```

```
people | for, of, against, who, to
can | if, used, all, does, not
also | and, a, the, for, as
history | and, links, author, main, in
they | to, their, but, however, other
have | for, other, however, a, than
rise | known, and, of, period, soviet
brother | father, friends, american, wife, him
troops | war, soviet, army, forces, police
paris | de, london, british, st, born
construction | in, place, or, low, built
frac | cdot, function, vector, mathbf, n
account | be, when, other, any, have
experience | art, often, consciousness, study, not
...

Epoch: 2/5
Loss:  2.8497090339660645
american | actor, actress, singer, nine, musician
six | eight, three, one, seven, five
world | in, nine, eight, united, war
from | of, it, in, is, an
th | st, nd, eight, roman, century
with | in, the, a, to, and
after | was, returned, became, president, in
seven | one, six, eight, five, three
hold | opposition, cabinet, party, after, not
consists | are, or, controls, type, between
freedom | society, claim, laws, reform, religion
discovered | discovery, earth, found, called, star
ocean | atlantic, sea, km, lies, west
behind | on, made, face, at, but
prince | emperor, son, john, king, ii
gold | was, industry, iron, from, banks
...

Epoch: 2/5
Loss:  2.839489221572876
use | used, source, different, like, multiple
between | both, the, is, of, these
five | eight, six, one, three, four
people | other, live, sexual, among, who
its | is, and, from, other, the
into | the, and, it, by, as
been | the, has, by, to, they
can | or, if, be, cannot, not
engineering | graduate, management, engineers, research, institute
http | www, com, org, html, external
```

```
pre | in, the, ages, important, speaking
magazine | tv, appeared, bbc, steve, com
operations | operation, force, planned, military, war
bill | william, jim, robert, actor, campbell
animals | animal, species, plants, eat, fish
troops | war, army, forces, armies, fighting
...

Epoch: 2/5
Loss:  2.6920864582061768
he | his, her, wife, had, father
will | must, action, we, may, therefore
th | rd, centuries, st, century, nd
up | with, down, or, called, the
two | three, one, zero, five, eight
by | the, and, had, of, into
has | its, the, and, through, it
seven | eight, one, four, five, nine
universe | thought, quantum, theories, existence, cosmological
frac | x, f, n, mathbf, cos
scale | quantities, range, waves, measure, measurements
award | awards, best, winning, won, winners
except | are, or, is, therefore, whereas
http | www, org, com, html, links
dr | d, actor, chris, roger, robert
troops | war, armies, forces, fighting, infantry
...

Epoch: 2/5
Loss:  2.6274213790893555
united | states, canada, nations, zealand, member
s | and, one, nine, by, was
there | is, be, than, of, roughly
no | insubstantial, import, and, info, done
nine | one, zero, five, four, seven
only | is, which, or, not, be
and | the, in, of, an, to
in | and, the, of, from, became
egypt | syria, arab, egyptian, israel, conquered
account | accounts, that, centuries, life, according
lived | his, death, family, ancient, brother
construction | industrial, buildings, oil, cement, built
road | traffic, city, routes, airport, roads
report | cia, international, department, committee, agency
know | you, give, want, him, think
numerous | mostly, western, region, more, from
...
```

```
Epoch: 2/5
Loss:  2.6998088359832764
four | three, two, five, eight, one
be | will, do, that, thus, or
at | he, his, when, in, a
nine | one, eight, seven, six, zero
over | was, and, between, there, the
so | can, but, it, if, this
for | and, with, a, as, in
in | of, and, the, a, on
creation | role, evil, outstanding, authority, christian
additional | stop, provide, multiple, separate, mandatory
quite | are, seem, much, thus, might
mathematics | theory, mathematical, philosophy, mathematicians, euclid
question | questions, we, beliefs, truth, therefore
smith | john, press, mormon, d, baptist
gold | silver, copper, precious, ore, iron
file | user, files, unix, microsoft, software
...

Epoch: 2/5
Loss:  2.444589614868164
all | a, and, separate, are, other
about | which, of, and, an, how
also | some, and, include, a, as
up | they, small, than, heavy, relatively
other | common, used, most, for, these
united | states, january, commission, canada, national
three | two, one, five, six, four
six | three, five, one, two, zero
instance | variables, vector, perfectly, operator, if
notes | note, text, instruments, tone, instrument
http | www, org, external, edu, html
paris | de, des, ne, leipzig, vienna
frac | equation, sqrt, cdot, mathbf, cos
engine | engines, powered, fuel, motors, vehicles
alternative | lyrics, minor, label, see, music
ice | hockey, water, frozen, temperatures, winter
...

Epoch: 3/5
Loss:  2.497276782989502
all | only, number, to, are, which
up | out, a, so, at, stand
not | if, they, be, that, difference
state | states, southern, government, parliamentary, district
his | he, him, wife, himself, had
can | normal, be, may, typically, than
```

```
he | his, him, himself, brother, mother
zero | three, two, four, five, six
pope | emperor, john, king, bishop, pius
award | best, awards, winners, won, academy
hold | just, himself, before, second, judges
scale | prices, scales, large, range, industry
hit | hits, batter, score, ball, lineup
prince | king, emperor, eldest, duke, succeeded
bible | hebrew, biblical, testament, books, torah
woman | she, her, man, husband, men
...

Epoch: 3/5
Loss:  2.707700490951538
seven | four, six, one, five, three
history | historical, links, article, external, list
the | to, a, and, of, from
zero | four, one, two, three, nine
states | united, state, america, american, national
between | a, is, the, as, or
their | they, the, have, as, to
about | a, is, do, are, for
orthodox | catholic, churches, church, catholics, christianity
existence | universe, theories, concept, argument, theory
award | awards, winners, won, awarded, winner
square | located, metres, area, west, defined
construction | constructed, buildings, built, building, downtown
pope | emperor, rome, pius, archbishop, bishop
versions | version, windows, pc, standard, microsoft
issue | has, act, despite, government, corruption
...

Epoch: 3/5
Loss:  2.4874892234802246
this | that, as, the, and, been
up | about, it, so, down, then
to | the, in, a, and, that
years | seven, male, five, age, births
the | of, in, and, a, to
but | the, not, to, in, some
used | are, commonly, sometimes, use, which
there | be, are, some, those, not
magazine | news, weekly, published, interview, titled
woman | her, female, children, male, married
hit | hits, album, songs, singles, billboard
san | francisco, california, los, diego, jose
pressure | temperature, liquid, gases, heat, pressures
channel | stations, channels, radio, broadcast, broadcasting
```

```
paris | de, des, french, le, leipzig
resources | resource, trade, topics, agricultural, provide
...

Epoch: 3/5
Loss:  2.6476051807403564
were | was, the, after, by, in
new | york, later, member, was, of
states | united, canada, republic, america, state
about | from, many, history, zero, four
eight | one, four, five, six, three
by | the, was, of, and, s
zero | five, one, two, nine, four
use | used, some, simple, are, be
creation | universe, belief, cosmology, evil, genesis
active | weak, combines, aspect, phosphorylation, stabilizers
troops | allied, forces, army, war, soldiers
primarily | its, has, particularly, due, these
heavy | heavier, metal, air, damage, pollution
test | tests, requires, match, ability, failure
mainly | native, settled, ethnic, east, today
universe | existence, cosmic, theories, cosmological, alien
...

Epoch: 3/5
Loss:  2.5929524898529053
can | or, cannot, is, does, have
are | is, example, or, different, there
state | states, legislature, federal, representatives, government
of | and, the, in, a, is
american | nine, eight, actress, actor, singer
the | and, of, by, a, to
into | and, the, by, to, of
at | the, to, after, of, was
institute | university, graduate, sciences, science, universities
scale | atmospheric, industrial, scales, natural, higher
dr | went, friend, roger, ian, johnson
grand | awarded, de, prestigious, france, french
arts | martial, art, school, styles, disciplines
mainly | most, as, well, notably, various
marriage | wives, married, daughter, divorce, marry
versions | version, windows, default, dos, unix
...

Epoch: 3/5
Loss:  2.2919394969940186
often | typically, or, generally, used, than
into | and, of, the, are, other
```

```
for | a, and, to, as, the
may | in, some, five, is, all
would | to, that, could, it, enough
most | are, is, some, and, other
zero | two, six, four, one, five
is | are, and, or, of, a
brother | son, daughter, wife, father, younger
cost | costs, required, market, demand, price
discovered | discoveries, discovery, unknown, scientists, evidence
issue | government, controversy, political, criticism, seemed
pressure | measures, gas, cooling, increases, liquid
versions | version, windows, microsoft, operating, desktop
stage | film, films, musical, comedy, performances
animals | animal, humans, mammals, species, insects
...

Epoch: 4/5
Loss:  2.223264694213867
that | it, be, they, some, had
can | typically, be, usually, are, or
a | an, for, the, is, with
on | a, and, in, for, first
into | the, from, region, are, through
six | five, four, zero, one, eight
some | these, still, those, even, that
of | in, the, and, with, to
http | www, org, htm, com, html
recorded | album, albums, record, song, music
numerous | around, including, near, been, extensive
behind | side, inside, straight, face, center
bible | hebrew, testament, biblical, torah, tanakh
additional | in, made, with, for, also
resources | agricultural, economic, natural, arable, soils
heavy | heavier, metal, air, fire, lighter
...

Epoch: 4/5
Loss:  2.0220956802368164
during | after, in, early, period, was
for | the, in, to, be, a
at | the, where, a, one, near
if | not, let, can, we, must
however | the, more, has, have, only
where | or, at, the, as, are
these | many, tend, often, more, such
an | in, as, with, is, of
assembly | legislative, parliament, council, appointed, judicial
consists | are, consist, composed, consisting, both
```

```
experience | knowledge, anything, mind, mental, that
units | unit, si, metric, soldiers, measured
older | age, household, median, years, families
smith | john, joseph, paul, jesus, david
report | news, committee, review, external, articles
freedom | rights, liberty, economic, liberalism, liberal
...

Epoch: 4/5
Loss:  2.3671505451202393
a | the, is, and, for, an
on | the, s, by, a, and
has | are, been, and, as, is
which | the, is, a, as, of
th | rd, nd, century, early, st
where | is, the, hence, or, at
d | b, one, j, laureate, c
that | the, this, an, not, is
square | m, metres, attractions, center, downtown
institute | university, college, technology, sciences, research
cost | costs, increase, expensive, supply, amount
recorded | album, song, record, songs, records
centre | situated, halifax, trains, street, center
test | tests, testing, evaluation, launch, problems
woman | she, married, her, wife, daughter
numerous | many, mostly, these, grouped, large
...

Epoch: 4/5
Loss:  2.361933469772339
some | many, these, most, often, other
he | his, she, him, attended, met
its | as, a, has, however, is
b | d, composer, p, laureate, f
are | is, such, other, which, can
also | a, to, as, are, the
had | was, returned, after, remained, later
six | seven, five, four, three, eight
brother | wife, son, daughter, throne, heir
hold | be, will, all, any, while
animals | animal, species, mammals, insects, breeding
shown | is, example, symbol, denoted, silent
heavy | heavier, lighter, affected, resulting, in
powers | exercised, minister, cabinet, constitutionally, territory
placed | height, thus, with, when, removed
applications | techniques, integrated, application, processing, systems
...
```

```
Epoch: 4/5
Loss:  2.2718966007232666
an | a, as, in, s, and
first | in, was, he, s, the
war | forces, troops, army, battle, ii
b | d, composer, poet, politician, writer
th | century, centuries, rd, nd, of
united | states, state, u, nations, presidents
than | have, it, few, as, their
some | there, many, such, is, well
ice | hockey, glacial, glaciers, rocks, glacier
file | files, formats, ftp, user, unix
bill | president, senator, jackson, alan, michael
derived | word, used, meaning, common, derives
centre | located, situated, railway, metropolitan, gardens
mean | word, equivalent, or, meaning, is
running | run, mike, coach, runs, linebacker
troops | forces, war, army, captured, battle
...

Epoch: 4/5
Loss:  2.39384126663208
over | total, of, than, in, the
no | that, if, so, stated, previous
he | his, him, himself, friend, she
into | the, which, from, and, of
and | in, of, s, the, to
seven | one, eight, four, six, nine
years | year, age, female, male, zero
use | used, these, or, commonly, devices
animals | animal, humans, mammals, species, human
primarily | mostly, most, other, main, many
behind | front, back, tied, side, door
bbc | listing, news, report, day, june
file | files, formats, software, user, unix
test | testing, tests, nuclear, matches, match
institute | university, technology, science, sciences, college
defense | personnel, police, against, tactics, weapons
...

Epoch: 5/5
Loss:  2.5499191284179688
where | which, called, for, it, same
have | been, of, few, are, not
of | the, in, and, a, is
often | others, sometimes, common, or, as
when | he, time, him, then, his
united | states, countries, the, kingdom, nations
```

```
war | forces, casualties, army, armed, troops
time | when, it, have, however, that
hold | any, god, then, theology, church
grand | prix, duchy, title, won, victories
operating | software, os, linux, microsoft, desktop
applied | refers, definition, which, terminology, or
applications | application, software, user, networking, allows
egypt | egyptian, syria, jerusalem, cairo, tunisia
creation | settlement, development, refer, ultimate, established
http | www, org, htm, com, edu
...

Epoch: 5/5
Loss:  2.19671368598938
not | be, but, can, to, it
when | the, be, to, that, must
often | some, such, many, or, are
five | four, eight, one, three, six
would | that, could, even, not, have
i | t, you, p, e, we
into | which, the, it, came, their
from | in, of, the, and, it
prince | succeeded, diplomat, duke, empress, lieutenant
lived | married, isolated, travelled, moved, native
numerous | throughout, thousands, large, including, several
scale | scales, measurements, produce, notes, harmonic
test | tests, cricket, match, testing, tested
know | we, you, think, understand, don
articles | links, wiki, documents, online, topics
file | files, user, software, executable, virus
...

Epoch: 5/5
Loss:  2.2722795009613037
zero | two, three, five, one, four
eight | one, five, six, nine, seven
after | returned, january, was, his, during
was | had, his, s, in, the
state | states, system, is, energy, between
b | d, p, k, y, g
have | are, it, been, some, that
who | him, father, were, she, his
animals | animal, mammals, species, prey, insects
joseph | thomas, john, smith, james, jr
account | books, bible, because, biblical, prophecy
something | think, you, really, we, ways
hit | hits, songs, singles, billboard, album
freedom | freedoms, liberty, rights, welfare, hayek
```

```
magazine | interview, weekly, published, magazines, awards
except | are, certain, have, called, only
...

Epoch: 5/5
Loss:  2.356029748916626
on | the, of, by, at, to
s | one, in, by, four, nine
or | are, not, the, is, they
eight | six, one, four, five, three
while | the, of, and, is, with
american | actress, americans, births, singer, association
as | is, the, of, in, and
six | eight, five, one, seven, four
rise | the, great, fall, growth, peasants
brother | wife, throne, daughter, son, uncle
articles | information, wikipedia, online, org, wiki
pressure | gases, liquid, vapor, heat, cooling
applied | widely, philosophy, refers, individual, usage
centre | railway, centres, buildings, seat, brunswick
engine | engines, powered, turbine, diesel, fuel
placed | large, side, specially, spot, placing
...

Epoch: 5/5
Loss:  2.3188295364379883
his | he, him, was, career, father
other | and, many, or, called, as
from | a, the, of, in, s
many | other, been, have, related, groups
time | before, his, in, at, the
b | d, politician, six, composer, poet
no | info, yet, major, though, well
see | list, article, external, links, references
alternative | minor, suites, more, major, groups
marriage | divorce, married, marry, marriages, sister
running | run, runs, yards, nfl, quarterback
assembly | parliament, council, legislative, parliamentary, constitution
powers | mutant, abilities, power, marvel, phoenix
behind | on, the, side, through, may
versions | version, text, windows, window, variant
stage | performing, film, comedy, broadway, few
...

Epoch: 5/5
Loss:  2.380748987197876
b | d, politician, actor, writer, seven
one | eight, five, three, six, seven
```

```
who | his, him, people, whom, their
d | b, writer, politician, composer, physicist
five | one, four, three, six, two
after | was, in, s, the, to
during | after, in, period, been, had
american | actor, b, musician, actress, singer
mean | value, arithmetic, denote, calculated, variance
san | francisco, diego, jose, los, santa
animals | animal, humans, species, human, mammals
applications | systems, software, integrated, application, operating
cost | costs, market, decrease, dollars, prices
bill | senator, bills, jackson, dave, drummer
proposed | proposal, future, demonstrated, discoveries, accepted
nobel | prize, laureate, physicist, recipient, physiology
...
```

2.2 Visualizing the word vectors

Below we'll use T-SNE to visualize how our high-dimensional word vectors cluster together. T-SNE is used to project these vectors into two dimensions while preserving local stucture. Check out this post from Christopher Olah to learn more about T-SNE and other ways to visualize high-dimensional data.

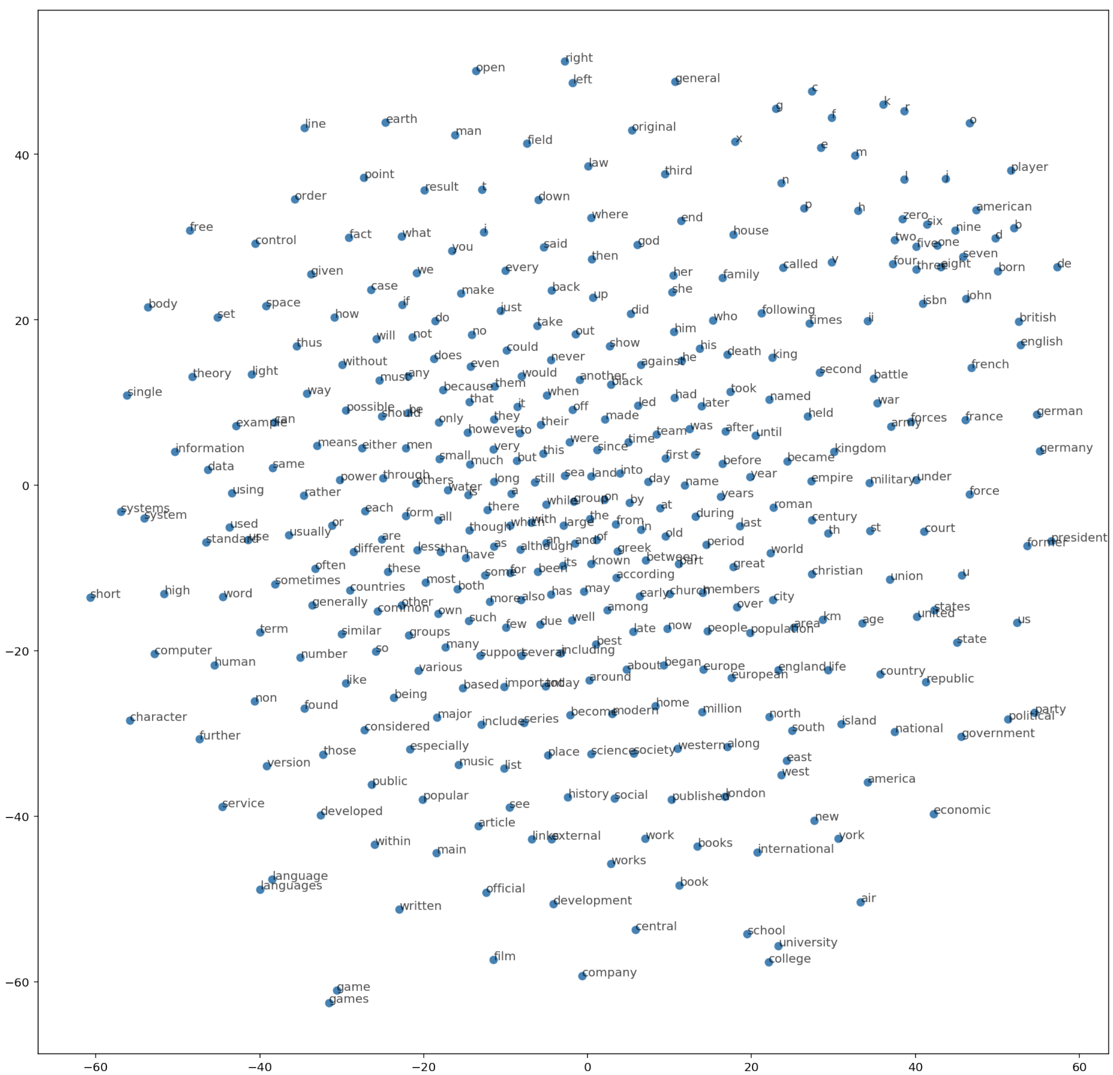
In [15]:
```
%matplotlib inline
%config InlineBackend.figure_format = 'retina'

import matplotlib.pyplot as plt
from sklearn.manifold import TSNE
```

In [16]:
```
# getting embeddings from the embedding layer of our model, by name
embeddings = model.in_embed.weight.to('cpu').data.numpy()
```

In [17]:
```
viz_words = 380
tsne = TSNE()
embed_tsne = tsne.fit_transform(embeddings[:viz_words, :])
```

In [18]:
```
fig, ax = plt.subplots(figsize=(16, 16))
for idx in range(viz_words):
    plt.scatter(*embed_tsne[idx, :], color='steelblue')
    plt.annotate(int_to_vocab[idx], (embed_tsne[idx, 0], embed_tsne[idx, 1]), alpha=0.7
```



In [ ]: