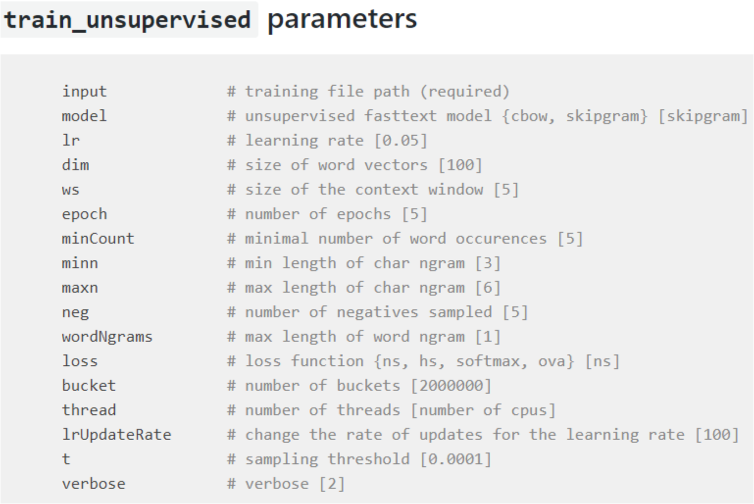
NetID: jl11046

Experiment setup:

Using the FastText train_unsupervised API, and the default parameters. This model reached a score of 0.6487. The default parameters are as follows:

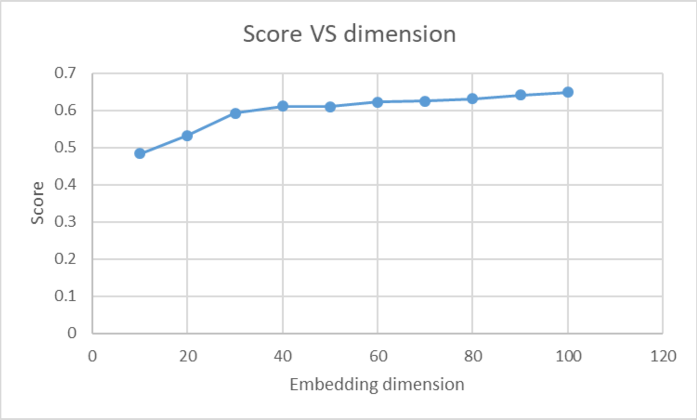


`train_unsupervised` parameters

```
input             # training file path (required)
model             # unsupervised fasttext model {cbow, skipgram} [skipgram]
lr                # learning rate [0.05]
dim               # size of word vectors [100]
ws                # size of the context window [5]
epoch             # number of epochs [5]
minCount          # minimal number of word occurences [5]
minn              # min length of char ngram [3]
maxn              # max length of char ngram [6]
neg               # number of negatives sampled [5]
wordNgrams        # max length of word ngram [1]
loss              # loss function {ns, hs, softmax, ova} [ns]
bucket            # number of buckets [2000000]
thread            # number of threads [number of cpus]
lrUpdateRate      # change the rate of updates for the learning rate [100]
t                 # sampling threshold [0.0001]
verbose           # verbose [2]
```

1. Change embedding dimensions:

dimension	score
10	0.4834
20	0.5326
30	0.5928
40	0.6114
50	0.6094
60	0.6224
70	0.6248
80	0.6314
90	0.6413
100	0.6487

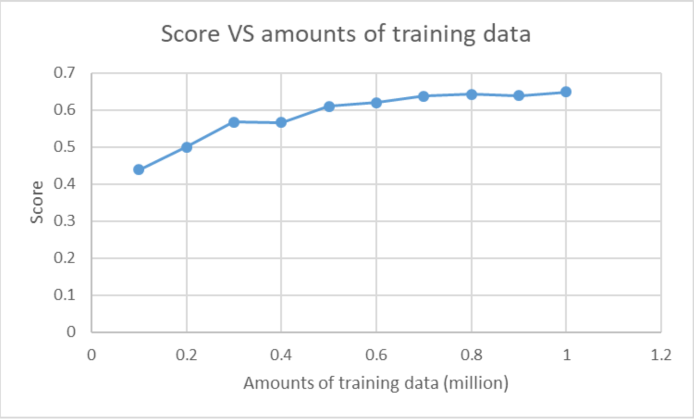


As the embedding dimension increases from 10 to 100, the score increases correspondingly.

The increase is rapid with lower dimensions and gets slower when the dimension gets large.

2. Change amount of training data:

data (million)	score	embeddings
0.1	0.4381	24231
0.2	0.5002	36386
0.3	0.5672	45565
0.4	0.5661	53420
0.5	0.6102	60364
0.6	0.6193	66550
0.7	0.6368	72223
0.8	0.6425	77428
0.9	0.6384	82331
1	0.6487	86965



As the amount of training data increases from 0.1 million to 1 million, the score basically increases correspondingly. However, at 0.4 million and 0.9 million, the score decreases a little.

The increase is rapid with small amount of training data and gets slower when the amount of data gets large.

We can also observe that the number of embeddings (i.e., the number of words) increases with the amount of the training data.