Star Space

September 24, 2021

0.1 Adventure with StarSpace: A Neural Embedding Approach

```
-> StarSpace is a general-purpose neural model for efficient learning of entity embeddings for

    -> Learning word, sentence or document level embeddings.
    -> Information retrieval: ranking of sets of entities/documents or objects, e.g. ranking we
    -> Text classification, or any other labeling task.
    -> Metric/similarity learning, e.g. learning sentence or document similarity.
    -> Content-based or Collaborative filtering-based Recommendation, e.g. recommending music o
    -> Embedding graphs, e.g. multi-relational graphs such as Freebase.
    -> Image classification, ranking or retrieval (e.g. by using existing ResNet features).
```

0.2 Advanced solution: StarSpace embeddings

Now you are ready to train your own word embeddings! In particular, you need to train embeddings specially for our task of duplicates detection. Unfortunately, StarSpace cannot be run on Windows and we recommend to use provided docker container or other alternatives. Don't delete results of this task because you will need it in the final project.

0.2.1 How it works and what's the main difference with word2vec?

The main point in this section is that StarSpace can be trained specifically for some tasks. In contrast to word2vec model, which tries to train similar embeddings for words in similar contexts, StarSpace uses embeddings for the whole sentence (just as a sum of embeddings of words and phrases). Despite the fact that in both cases we get word embeddings as a result of the training, StarSpace embeddings are trained using some supervised data, e.g. a set of similar sentence pairs, and thus they can better suit the task.

[36]:
```python
import pandas as pd #Analysis
import matplotlib.pyplot as plt #Visulization
import seaborn as sns #Visulization
import numpy as np #Analysis
from scipy.stats import norm #Analysis
from sklearn.preprocessing import StandardScaler #Analysis
from scipy import stats #Analysis
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
import gc
import os
import string
color = sns.color_palette()

%matplotlib inline

from plotly import tools
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.express as px
```

[150]:
```
df_train = pd.read_csv('data/drugsComTrain_raw.csv')
df_test = pd.read_csv('data/drugsComTest_raw.csv')
```

[151]:
```
df_train.head()
```

[151]:
```
   uniqueID                  drugName                     condition  \
0    206461                 Valsartan  Left Ventricular Dysfunction
1     95260                Guanfacine                          ADHD
2     92703                    Lybrel                 Birth Control
3    138000                Ortho Evra                 Birth Control
4     35696  Buprenorphine / naloxone             Opiate Dependence

                                              review  rating       date  \
0  "It has no side effect, I take it in combinati…       9  20-May-12
1  "My son is halfway through his fourth week of …       8  27-Apr-10
2  "I used to take another oral contraceptive, wh…       5  14-Dec-09
3  "This is my first time using any form of birth…       8   3-Nov-15
4  "Suboxone has completely turned my life around…       9  27-Nov-16

   usefulCount
0           27
1          192
2           17
3           10
4           37
```

[152]:
```
df_test.head()
```

[152]:
```
   uniqueID         drugName                     condition  \
0    163740      Mirtazapine                    Depression
1    206473       Mesalamine  Crohn's Disease, Maintenance
2    159672          Bactrim       Urinary Tract Infection
3     39293         Contrave                   Weight Loss
4     97768  Cyclafem 1 / 35                 Birth Control
```

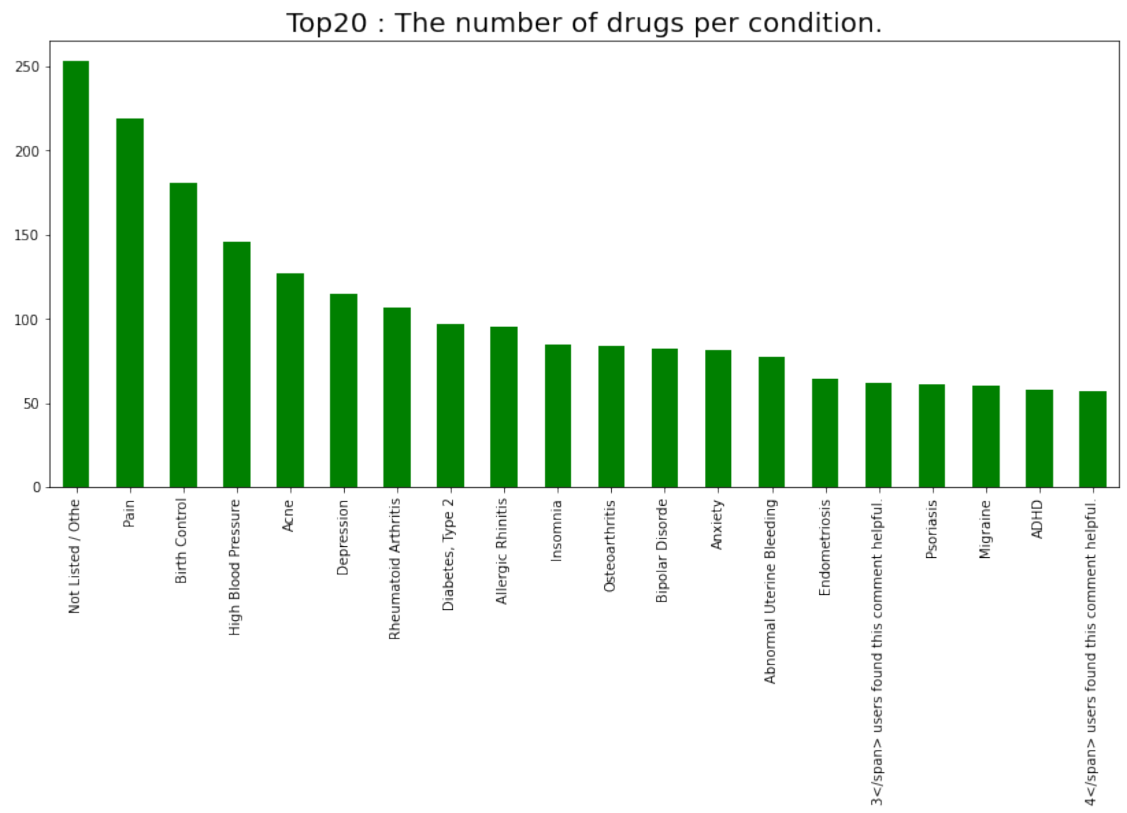
```
                                              review  rating       date  \
0  "I&#039;ve tried a few antidepressants over th…      10  28-Feb-12
1  "My son has Crohn&#039;s disease and has done …       8  17-May-09
2                      "Quick reduction of symptoms"       9  29-Sep-17
3  "Contrave combines drugs that were used for al…       9   5-Mar-17
4  "I have been on this birth control for one cyc…       9  22-Oct-15

   usefulCount
0           22
1           17
2            3
3           35
4            4
```

[153]:
```
df_all = pd.concat([df_train,df_test]).reset_index()
del df_all['index']
```

[154]:
```
df_all = df_all.dropna(axis=0)
df_all = df_all.drop_duplicates(subset=['uniqueID','review'])
```

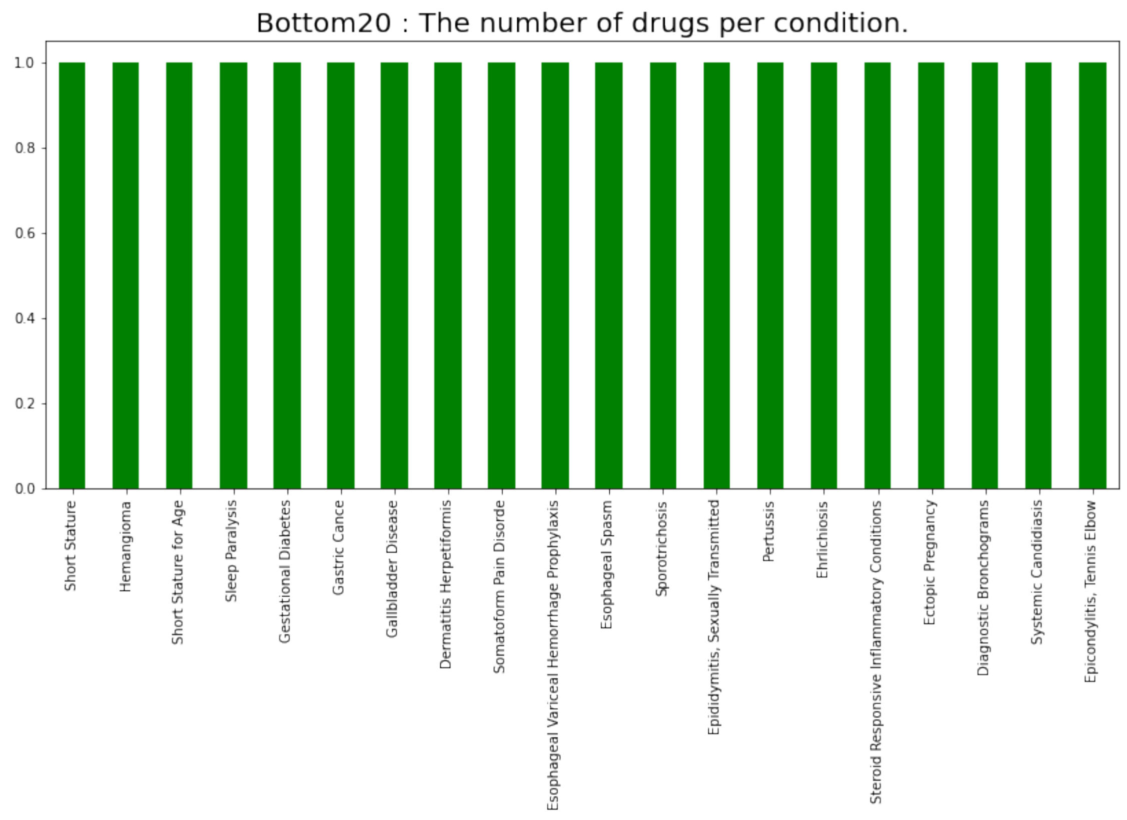
[155]:
```
condition_dn = df_all.groupby(['condition'])['drugName'].nunique().
 ↪sort_values(ascending=False)
condition_dn[0:20].plot(kind="bar", figsize = (14,6), fontsize =␣
 ↪10,color="green")
plt.xlabel("", fontsize = 20)
plt.ylabel("", fontsize = 20)
plt.title("Top20 : The number of drugs per condition.", fontsize = 20)
```

[155]:
```
Text(0.5, 1.0, 'Top20 : The number of drugs per condition.')
```



```
[156]: condition_dn = df_all.groupby(['condition'])['drugName'].nunique().
        ↪sort_values(ascending=False)

       condition_dn[condition_dn.shape[0]-20:condition_dn.shape[0]].plot(kind="bar",␣
        ↪figsize = (14,6), fontsize = 10,color="green")
       plt.xlabel("", fontsize = 20)
       plt.ylabel("", fontsize = 20)
       plt.title("Bottom20 : The number of drugs per condition.", fontsize = 20)
```

```
[156]: Text(0.5, 1.0, 'Bottom20 : The number of drugs per condition.')
```



[157]:
```
count_df = df_all[['condition','review']].groupby('condition').
↪aggregate({'review':'count'}).reset_index().
↪sort_values('review',ascending=False)
count_df.head()
```

[157]:
```
         condition  review
175  Birth Control   38436
273     Depression   12164
613           Pain    8245
133        Anxiety    7812
87            Acne    7435
```

[158]:
```
target_conditions = count_df[count_df['review']>3000]['condition'].values
```

[159]:
```
def condition_parser(x):
    if x in target_conditions:
        return x
    else:
        return "OTHER"

df_all['condition'] = df_all['condition'].apply(lambda x: condition_parser(x))
```

[160]:
```
df_all = df_all[df_all['condition']!='OTHER']
```

[161]:
```
px.bar(count_df[count_df['review']>3000],x='condition',y='review')
```

[162]:
```
import contractions
```

[163]:
```
import re

def clean_text(x):
    pattern = r'[^a-zA-z0-9\s]'
    text = re.sub(pattern, '', x)
    return x

def clean_numbers(x):
    if bool(re.search(r'\d', x)):
        x = re.sub('[0-9]{5,}', '#####', x)
        x = re.sub('[0-9]{4}', '####', x)
        x = re.sub('[0-9]{3}', '###', x)
        x = re.sub('[0-9]{2}', '##', x)
    return x
```

[164]:
```
# lower the text
df_all["review"] = df_all["review"].apply(lambda x: x.lower())

# Clean the text
df_all["review"] = df_all["review"].apply(lambda x: clean_text(x))

# Clean numbers
df_all["review"] = df_all["review"].apply(lambda x: clean_numbers(x))

# Clean Contractions
df_all["review"] = df_all["review"].apply(lambda x:  " ".join([contractions.
 ↪fix(word) for word in x.split()]))
```

[165]:
```
df_all.head()
```

[165]:
```
   uniqueID        drugName                  condition  \
1     95260      Guanfacine                       ADHD
2     92703          Lybrel              Birth Control
3    138000      Ortho Evra              Birth Control
6    165907  Levonorgestrel  Emergency Contraception
7    102654    Aripiprazole           Bipolar Disorde

                                              review  rating       date  \
1  "my son is halfway through his fourth week of …       8  27-Apr-10
2  "i used to take another oral contraceptive, wh…       5  14-Dec-09
3  "this is my first time using any form of birth…       8   3-Nov-15
```

```
6  "he pulled out, but he cummed a bit in me. i t…       1   7-Mar-17
7  "abilify changed my life. there is hope. i was…      10  14-Mar-15

   usefulCount
1          192
2           17
3           10
6            5
7           32
```

[196]:
```
f = open('data.txt','a')
for row in df_all.itertuples():
    f.write(row.review)
    f.write("\n")
```

[197]:
```
import sentencepiece as spm
```

[198]:
```
spm.SentencePieceTrainer.train(input='data.txt', model_prefix='m',
                               vocab_size=10000)
```

[199]:
```
sp = spm.SentencePieceProcessor(model_file='m.model')
```

0.3 Classification: Can you predict the patient's condition based on the review?

[200]:
```
df_all.condition.unique()
```

[200]:
```
array(['ADHD', 'Birth Control', 'Emergency Contraception',
       'Bipolar Disorde', 'Depression', 'Obesity', 'Insomnia',
       'Vaginal Yeast Infection', 'Pain', 'Diabetes, Type 2', 'Anxiety',
       'Acne', 'High Blood Pressure', 'Weight Loss'], dtype=object)
```

[201]:
```
LABELS = ['ADHD', 'Birth Control', 'Emergency Contraception',
       'Bipolar Disorde', 'Depression', 'Obesity', 'Insomnia',
       'Vaginal Yeast Infection', 'Pain', 'Diabetes, Type 2', 'Anxiety',
       'Acne', 'High Blood Pressure', 'Weight Loss']

EMPTY_ID = len(LABELS)

def create_labeled_string(row):
    parts = sp.encode(row["review"], out_type = 'str')
    parts.append("__label__{}".format(row['condition']))
    return " ".join(parts)
```

[202]:
```
from sklearn.model_selection import train_test_split
train, test =␣
 ↪train_test_split(df_all[['review','condition']],stratify=df_all['condition'],
                                                    test_size=0.25)
```

[203]:
```
train_lines = train.apply(create_labeled_string, 1)
with open("cache/train.txt", "w") as f:
    f.write("\n".join(train_lines))
val_lines = test.apply(create_labeled_string, axis=1)
with open("cache/val.txt", "w") as f:
    f.write("\n".join(val_lines))
```

[214]:
```
! /Users/subir/Starspace/./starspace train  -thread 4 -trainFile cache/train.
 ↪txt -model cache/starspace.modela
```

```
Arguments:
lr: 0.01
dim: 100
epoch: 5
maxTrainTime: 8640000
validationPatience: 10
saveEveryEpoch: 0
loss: hinge
margin: 0.05
similarity: cosine
maxNegSamples: 10
negSearchLimit: 50
batchSize: 5
thread: 4
minCount: 1
minCountLabel: 1
label: __label__
label: __label__
ngrams: 1
bucket: 2000000
adagrad: 1
trainMode: 0
fileFormat: fastText
normalizeText: 0
dropoutLHS: 0
dropoutRHS: 0
useWeight: 0
weightSep: :
Start to initialize starspace model.
Build dict from input file : cache/train.txt
Read 9M words
Number of words in dictionary:  10085
Number of labels in dictionary: 14
Loading data from file : cache/train.txt
Total number of examples loaded : 83673
```

```
Training epoch 0: 0.01 0.002
Epoch: 100.0%  lr: 0.008000  loss: 0.009490  eta: 0h2m  tot: 0h0m41s
(20.0%)tot: 0h0m1s  (0.7%) (6.8%)  tot: 0h0m19s  (8.8%)63.8%  lr: 0.008602
loss: 0.011834  eta: 0h3m  tot: 0h0m27s  (12.8%)72.8%  lr: 0.008506  loss:
0.011035  eta: 0h3m  tot: 0h0m30s  (14.6%)0h0m31s  (14.8%)32s  (15.1%)15.4%)s
(15.8%)94.7%  lr: 0.008121  loss: 0.009712  eta: 0h2m  tot: 0h0m39s  (18.9%)m39s
(19.0%)
 ---+++                Epoch    0 Train error : 0.00939365 +++---
Training epoch 1: 0.008 0.002
Epoch: 100.0%  lr: 0.006024  loss: 0.003038  eta: 0h1m  tot: 0h1m21s  (40.0%)8%
lr: 0.007855  loss: 0.002965  eta: 0h2m  tot: 0h0m44s  (21.4%)0m44s
(21.4%)0h0m45s  (21.9%)49s  (23.9%)33.7%  lr: 0.007446  loss: 0.003067  eta:
0h2m  tot: 0h0m55s  (26.7%)0m55s  (26.9%)50.9%  lr: 0.007060  loss: 0.003032
eta: 0h2m  tot: 0h1m1s  (30.2%)54.1%  lr: 0.006988  loss: 0.003112  eta: 0h2m
tot: 0h1m2s  (30.8%)57.4%  lr: 0.006964  loss: 0.003138  eta: 0h2m  tot: 0h1m4s
(31.5%)s  (31.9%)m21s  (39.9%)
 ---+++                Epoch    1 Train error : 0.00302130 +++---
Training epoch 2: 0.006 0.002
Epoch: 100.0%  lr: 0.004000  loss: 0.001989  eta: 0h1m  tot: 0h2m0s
(60.0%)1m23s  (40.9%)41.2%)m26s  (42.4%)1m31s  (45.1%)45.8%) (46.4%)33.3%  lr:
0.005349  loss: 0.002033  eta: 0h1m  tot: 0h1m34s  (46.7%)s  (46.7%)m38s
(48.5%)%)44s  (51.7%)m  tot: 0h1m45s  (52.5%)72.1%  lr: 0.004578  loss: 0.002022
eta: 0h1m  tot: 0h1m49s  (54.4%)73.1%  lr: 0.004554  loss: 0.002013  eta: 0h1m
tot: 0h1m49s  (54.6%)87.8%  lr: 0.004241  loss: 0.002022  eta: 0h1m  tot:
0h1m55s  (57.6%)59s  (59.7%)
 ---+++                Epoch    2 Train error : 0.00206770 +++---
Training epoch 3: 0.004 0.002
Epoch: 100.0%  lr: 0.002000  loss: 0.001655  eta: <1min   tot: 0h2m38s
(80.0%)%)25.8%  lr: 0.003398  loss: 0.001564  eta: 0h1m  tot: 0h2m10s
(65.2%)11s  (65.9%)0h2m19s  (70.0%) (70.4%)71.0%  lr: 0.002386  loss: 0.001687
eta: <1min   tot: 0h2m27s  (74.2%)74.5%)76.2%) (77.7%) (78.7%)97.2%  lr:
0.002048  loss: 0.001636  eta: <1min   tot: 0h2m37s  (79.4%) (79.9%)
 ---+++                Epoch    3 Train error : 0.00159319 +++---
Training epoch 4: 0.002 0.002
Epoch: 100.0%  lr: 0.000024  loss: 0.001269  eta: <1min   tot: 0h3m18s
(100.0%)54s  (88.1%)4s  (93.2%)76.4%  lr: 0.000458  loss: 0.001296  eta: <1min
tot: 0h3m8s  (95.3%)85.7%  lr: 0.000217  loss: 0.001288  eta: <1min   tot:
0h3m12s  (97.1%) (97.6%)
 ---+++                Epoch    4 Train error : 0.00131867 +++---
Saving model to file : cache/starspace.model
Saving model in tsv format : cache/starspace.model.tsv
```

[217]:
```
! /Users/subir/Starspace/./starspace test  -thread 4 -testFile cache/val.txt␣
↪-model cache/starspace.model -predictionFile cache/starspace.pred
```

```
Arguments:
lr: 0.01
dim: 100
```

```
epoch: 5
maxTrainTime: 8640000
validationPatience: 10
saveEveryEpoch: 0
loss: hinge
margin: 0.05
similarity: cosine
maxNegSamples: 10
negSearchLimit: 50
batchSize: 5
thread: 4
minCount: 1
minCountLabel: 1
label: __label__
label: __label__
ngrams: 1
bucket: 2000000
adagrad: 1
trainMode: 0
fileFormat: fastText
normalizeText: 0
dropoutLHS: 0
dropoutRHS: 0
useWeight: 0
weightSep: :
Start to load a trained starspace model.
STARSPACE-2018-2
Model loaded.
Loading data from file : cache/val.txt
Total number of examples loaded : 27891
Predictions use 14 known labels.
------Loaded model args:
Arguments:
lr: 0.01
dim: 100
epoch: 5
maxTrainTime: 8640000
validationPatience: 10
saveEveryEpoch: 0
loss: hinge
margin: 0.05
similarity: cosine
maxNegSamples: 10
negSearchLimit: 50
batchSize: 5
thread: 4
minCount: 1
minCountLabel: 1
```

```
label: __label__
label: __label__
ngrams: 1
bucket: 2000000
adagrad: 1
trainMode: 0
fileFormat: fastText
normalizeText: 0
dropoutLHS: 0
dropoutRHS: 0
useWeight: 0
weightSep: :
Predictions use 14 known labels.
Evaluation Metrics :
hit@1: 0.473988 hit@10: 0.998315 hit@20: 1 hit@50: 1 mean ranks : 1.66581 Total
examples : 27891
```

```
[ ]:
```