```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import pairwise
from sklearn.manifold import TSNE


defining_model = dict()
f = open('/content/sample_data/vectors.txt', encoding="utf8")

for line in f:
    values = line.split()
    word = values[0]
    coefficient = np.asarray(values[1:], dtype='float32')
    defining_model[word] = coefficient
f.close()
print('Loaded %s word vectors.' % (len(defining_model)))
```

```
      Loaded 400001 word vectors.
```

```
def similar_words(defining_model, word,target_list , num) :
    cosine_dict ={}
    word_list = []
    if(word in defining_model.keys()):
      a = defining_model[word]
    else:
      print('No embeddings found for %s' % word)
      return

    for i in target_list :
        if i != word :
            a = defining_model [i]
            cos_sim = pairwise.cosine_similarity(a.reshape(1, -1),a.reshape(1, -1))
            cosine_dict[i] = cos_sim
    dist_sort=sorted(cosine_dict.items(), key=lambda dist: dist[1],reverse = True)
    for i in dist_sort:
        word_list.append((i[0], i[1]))
    return word_list[0:num]


similar_words(defining_model,'life',list(defining_model.keys()),20)
```

```
[('difference', array([[1.0000004]], dtype=float32)),
 ('forgotten', array([[1.0000004]], dtype=float32)),
 ('chester', array([[1.0000004]], dtype=float32)),
 ('protagonist', array([[1.0000004]], dtype=float32)),
 ('amounting', array([[1.0000004]], dtype=float32)),
 ('laude', array([[1.0000004]], dtype=float32)),
 ('congressionally', array([[1.0000004]], dtype=float32)),
 ('twinkling', array([[1.0000004]], dtype=float32)),
 ('dianna', array([[1.0000004]], dtype=float32)),
 ('replaceable', array([[1.0000004]], dtype=float32)),
 ('cnn.com', array([[1.0000004]], dtype=float32)),
 ('acupuncturist', array([[1.0000004]], dtype=float32)),
 ('lembo', array([[1.0000004]], dtype=float32)),
 ('35-billion', array([[1.0000004]], dtype=float32)),
 ('thf', array([[1.0000004]], dtype=float32)),
 ('najd', array([[1.0000004]], dtype=float32)),
 ('meilan', array([[1.0000004]], dtype=float32)),
 ('silverleaf', array([[1.0000004]], dtype=float32)),
 ('featherbed', array([[1.0000004]], dtype=float32)),
 ('18-26', array([[1.0000004]], dtype=float32))]
```

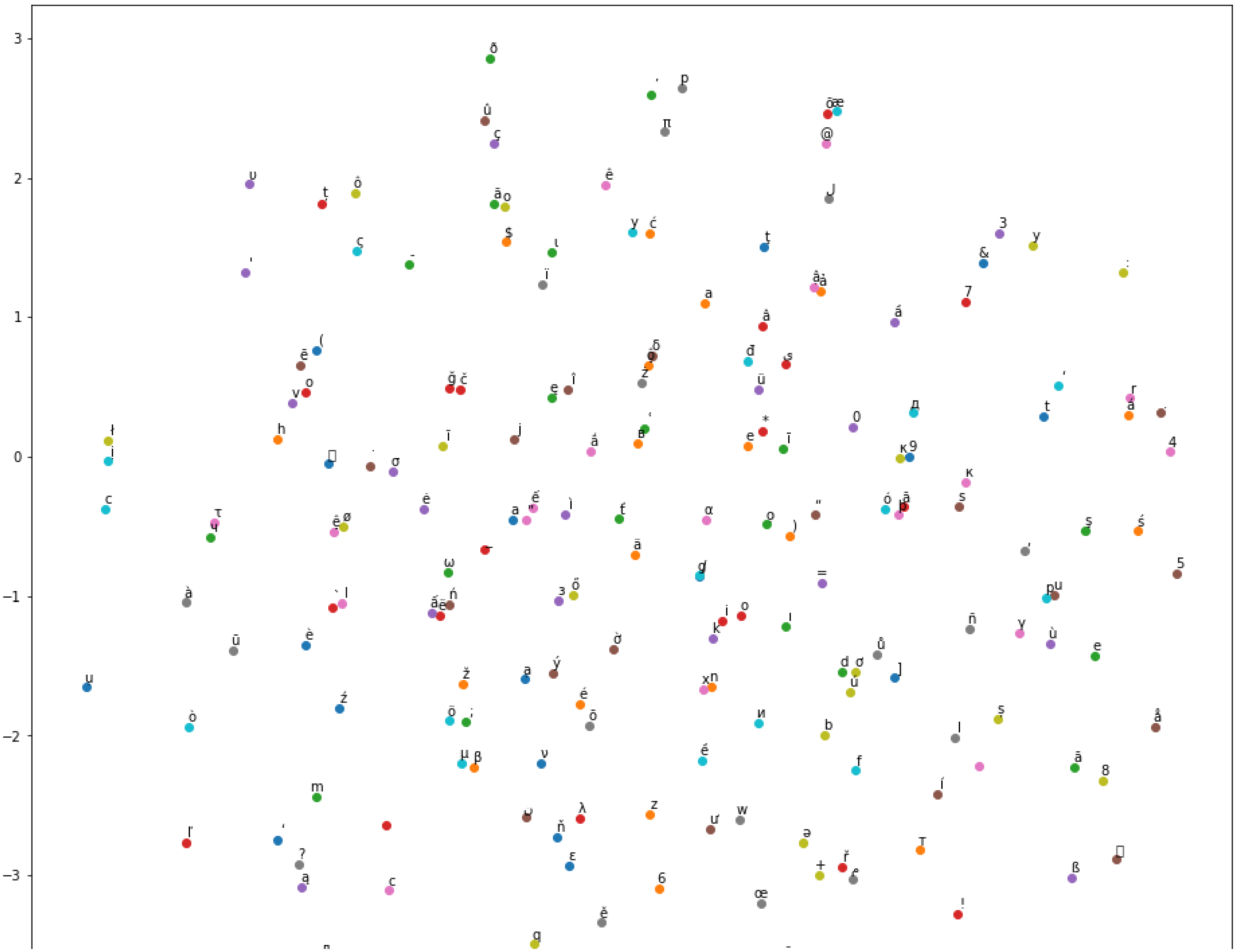
```
similar_words(defining_model,'market',list(defining_model.keys()),20)
```

```
[('difference', array([[1.0000004]], dtype=float32)),
 ('forgotten', array([[1.0000004]], dtype=float32)),
 ('chester', array([[1.0000004]], dtype=float32)),
 ('protagonist', array([[1.0000004]], dtype=float32)),
 ('amounting', array([[1.0000004]], dtype=float32)),
 ('laude', array([[1.0000004]], dtype=float32)),
 ('congressionally', array([[1.0000004]], dtype=float32)),
 ('twinkling', array([[1.0000004]], dtype=float32)),
 ('dianna', array([[1.0000004]], dtype=float32)),
 ('replaceable', array([[1.0000004]], dtype=float32)),
 ('cnn.com', array([[1.0000004]], dtype=float32)),
 ('acupuncturist', array([[1.0000004]], dtype=float32)),
 ('lembo', array([[1.0000004]], dtype=float32)),
 ('35-billion', array([[1.0000004]], dtype=float32)),
 ('thf', array([[1.0000004]], dtype=float32)),
 ('najd', array([[1.0000004]], dtype=float32)),
 ('meilan', array([[1.0000004]], dtype=float32)),
 ('silverleaf', array([[1.0000004]], dtype=float32)),
 ('featherbed', array([[1.0000004]], dtype=float32)),
 ('18-26', array([[1.0000004]], dtype=float32))]
```

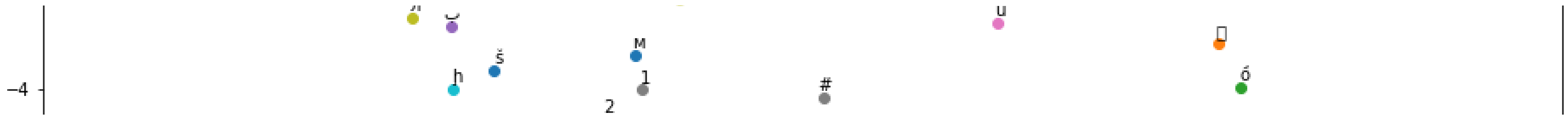
```
similar_words(defining_model,'stanford',list(defining_model.keys()),20)
```

```
[('difference', array([[1.0000004]], dtype=float32)),
 ('forgotten', array([[1.0000004]], dtype=float32)),
 ('chester', array([[1.0000004]], dtype=float32)),
 ('protagonist', array([[1.0000004]], dtype=float32)),
 ('amounting', array([[1.0000004]], dtype=float32)),
 ('laude', array([[1.0000004]], dtype=float32)),
 ('congressionally', array([[1.0000004]], dtype=float32)),
 ('twinkling', array([[1.0000004]], dtype=float32)),
 ('dianna', array([[1.0000004]], dtype=float32)),
 ('replaceable', array([[1.0000004]], dtype=float32)),
 ('cnn.com', array([[1.0000004]], dtype=float32)),
 ('acupuncturist', array([[1.0000004]], dtype=float32)),
 ('lembo', array([[1.0000004]], dtype=float32)),
 ('35-billion', array([[1.0000004]], dtype=float32)),
 ('thf', array([[1.0000004]], dtype=float32)),
 ('najd', array([[1.0000004]], dtype=float32)),
 ('meilan', array([[1.0000004]], dtype=float32)),
 ('silverleaf', array([[1.0000004]], dtype=float32)),
 ('featherbed', array([[1.0000004]], dtype=float32)),
 ('18-26', array([[1.0000004]], dtype=float32))]
```

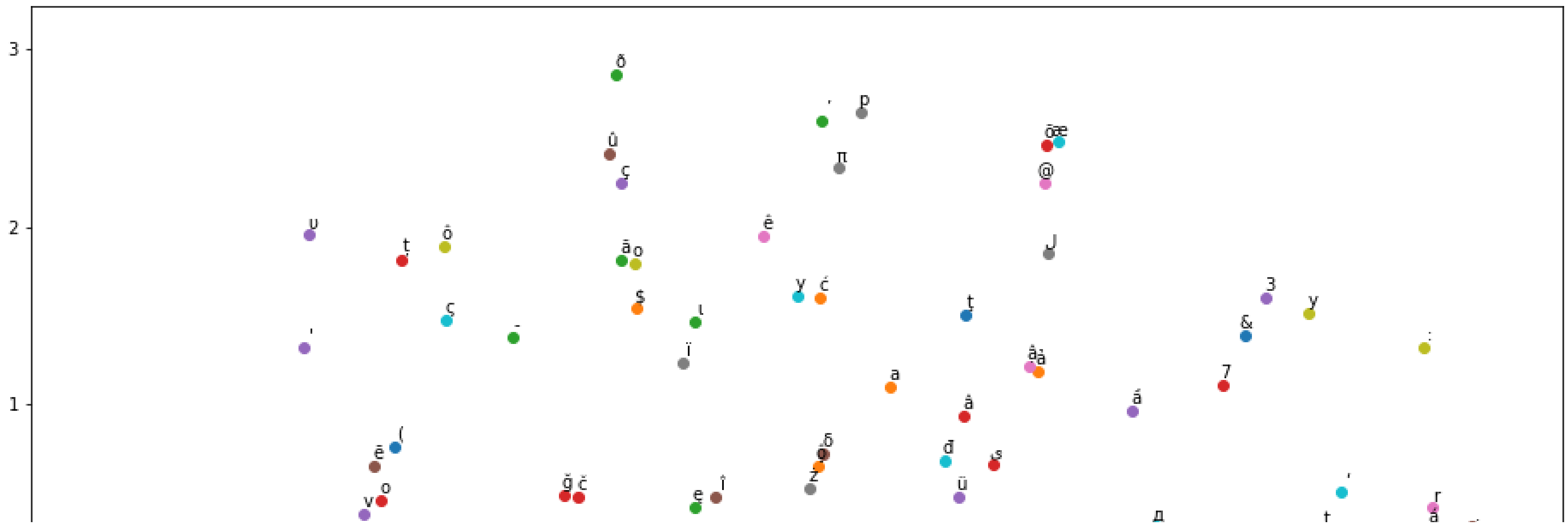
```
from nltk.tokenize import sent_tokenize, word_tokenize
import warnings

warnings.filterwarnings(action = 'ignore')

import gensim
from gensim.models import Word2Vec

def tsne_plot(model, word):
    "Creates and TSNE model and plots it"
    labels = []
    tokens = []
    if(word in defining_model.keys()):
        for word in model.wv.vocab:
            tokens.append(model[word])
            labels.append(word)
        tsne_model = TSNE(perplexity=40, n_components=2, init='pca', n_iter=2500, random_state=23)
        new_values = tsne_model.fit_transform(tokens)

        x = []
        y = []
        for value in new_values:
            x.append(value[0])
            y.append(value[1])

        plt.figure(figsize=(16, 16))
        for i in range(len(x)):
            plt.scatter(x[i],y[i])
            plt.annotate(labels[i],
                        xy=(x[i], y[i]),
                        xytext=(5, 2),
                        textcoords='offset points',
                        ha='right',
                        va='bottom')
        plt.show()


f = open('/content/sample_data/vectors.txt', encoding="utf8")
model = gensim.models.Word2Vec(f)
for line in f:
    values = line.split()
    word = values[0]
    coefficient = np.asarray(values[1:], dtype='float32')
    model[word] = coefficient
f.close()


tsne_plot(model, 'life')
```



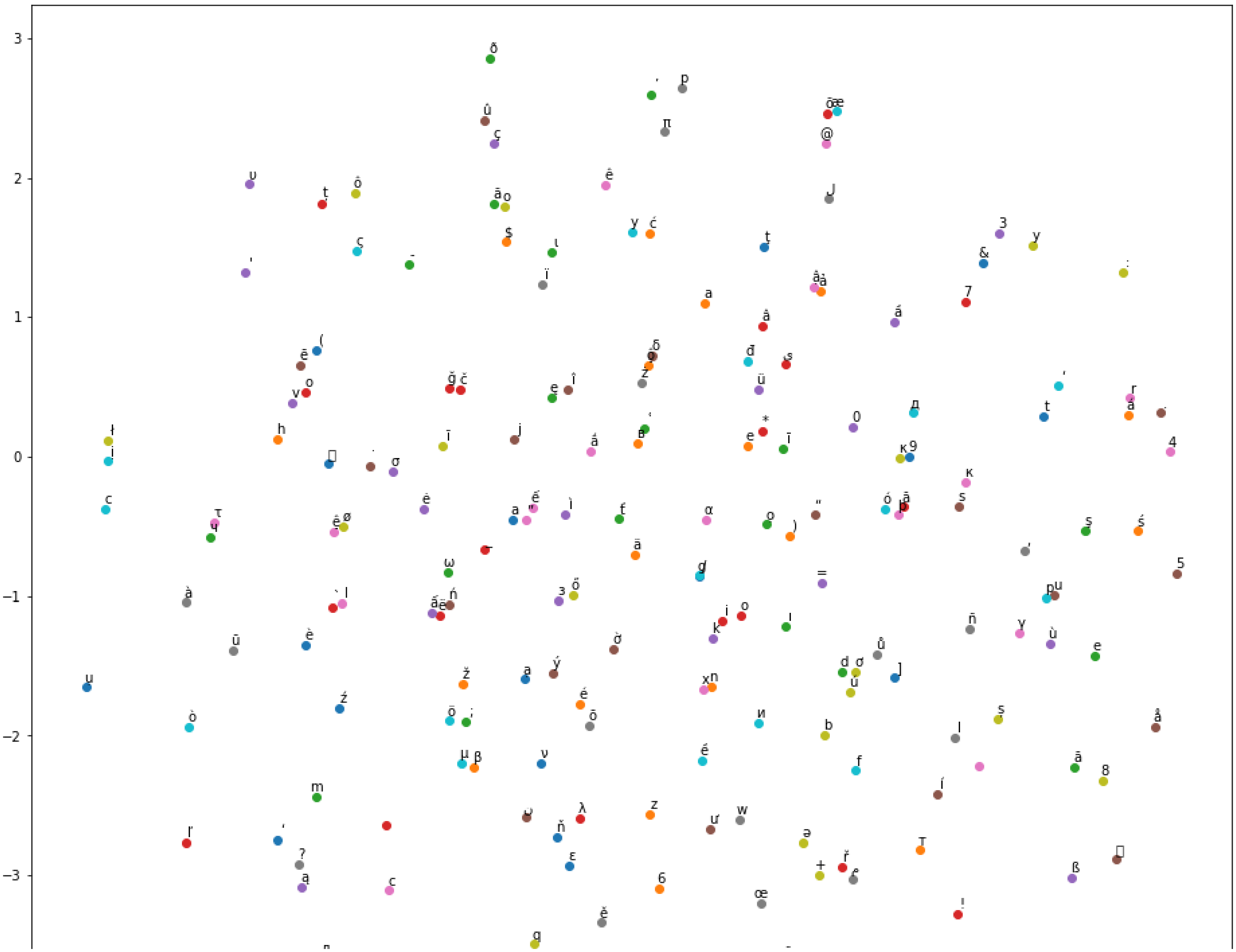
3
2
1
0
−1
−2
−3
ő
ˌ
p
û
ç
π
œ
ǣ
@
υ
ê
ʈ
ô
ā
o
ɹ
ç
$
y
ć
ι
ˈ
-
ï
t
3
&
y
:
a
â
ȁ
ạ
ả
7
(
ē
δ
g
đ
ʂ
v
o
ğ
č
e
î
ž
ü
ˈ
r
t
ä
д
ł
h
ī
j
ǻ
ˈ
в
*
e
ī
0
κ
9
4
i
.
σ
κ
ė
a
ǘ
ì
ŧ
α
o
"
ó
ā
þ
s
c
τ
ч
ê
ø
ṣ
ś
)
ʻ
ä
−
ω
q
=
5
à
`
l
ǻ
ë
ń
з
ö
i
o
l
p
u
ū
è
k
ở
ñ
γ
ù
e
ŭ
ŷ
u
ž
a
d
σ
ű
]
ǿ
x
n
ó
ż
é
ō
;
õ
и
b
l
s
å
µ
β
v
é
f
ā
8
m
í
ʋ
λ
z
ư
w
ř
ə
ʈ
ľ
ˌ
?
+
ř
ɇ
ß
ą
c
ε
6
œ
ě
!
q



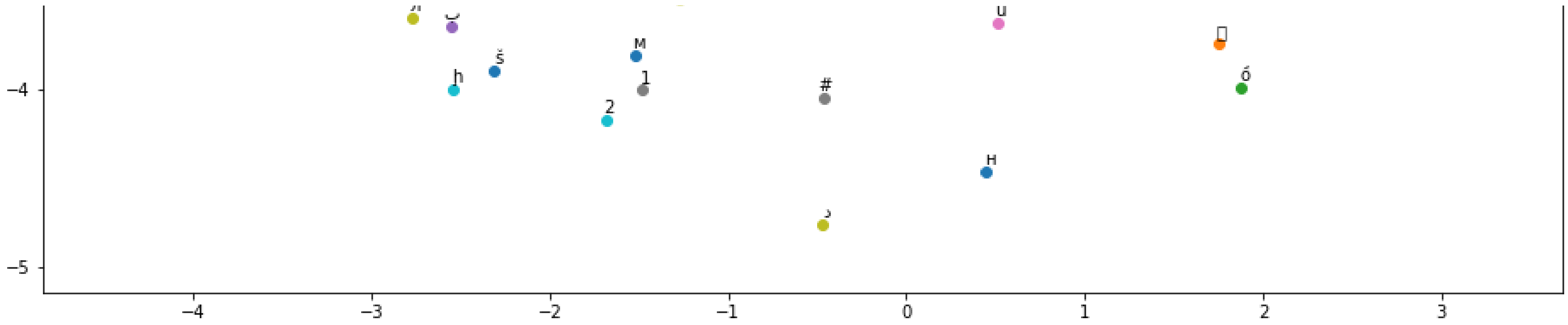
```
tsne_plot(model, 'market')
```



```
tsne_plot(model, 'stanford')
```



3
2
1
0
−1
−2
−3
ő
p
û
π
ç
ōæ
@
ʋ
ô
ê
ṭ
ā o
ȷ
y ć
$
ç
ι
t
3
&
y
-
ï
:
a
ȧȧ
ȁ
ȃ
7
(
ē
δ
g
đ
ɕ
ğ č
ẑ
ü
v o
e î
'
r
д
t
ā
ɫ
h
ī
j
ȁ
в
*
e
ī
0
ĸ 9
4
i
ʘ
σ
κ
c
ė
a ɛ́
ì
ť
α
ő
ȁ
s
τ
ч
ė ø
o
)
ṣ
ś
ä
ω
g
=
5
à
` l
ắē ń
з ö
pu
i
o
k
l
ñ
γ
ū
è
ơ
ù
e
û
u
ž
a
ŷ
d σ
ů
]
ẋn
é
ò
ź
ō ;
õ
и
b
ṡ
l
ȧ
μ β
ν
é
f
ā
8
m
í
ʊ
λ
z
ư
w
ľ
ň
ə
ᴛ
?
ε
6
+
ř
ʑ
ß
ą
c
œ
!
ĕ
q



u
M
š
ḫ
1
#
ó
2
H
−4
−5
−4
−3
−2
−1
0
1
2
3