Word2Vec 实例

罗瀚

2017.4.17

Word2Vec

- Word to vector 也就是词汇转化成向量
- 将词汇映射至一个 连续的k维向量空间中，向量空间上的相似度可以用来表达文本语义上的相似度
- NLP：聚类、找同义词、词性分析等等

TensorFlow

- 谷歌基于DistBelief进行研发的第二代人工智能学习系统， TensorFlow是将复杂 的数据结构传输至人工智能神经网中进行分析和处理过程的系统。
- 安装：

  Windows:　　window64位

  Python3.5(只能是3.5)

  cuda8.0

  （可以直接安装 Python 3.5 from Anaconda，然后用pip 来安装TensorFlow）

  Linux:

  用pip来安装即可，对python 的版本也很友好

1.Salsalate

- 从NCBI中搜索salsalate并下载了225篇文献的摘要+上次课的salsalate的文本总大小145KB.

```
def read_file(filename):
    file_read = open(filename,'r').read()
    word_=re.sub("[^a-zA-Z]+", " ",file_read).lower()
    words=list(word_.split())
    return words

path='/data1/gzhang/hluo/'
file_='Clean_result2.txt'
filename=path+file_

words = read_file(filename)
```

最后得到词的总数

```
In [14]: len(words)
Out[14]: 20762
```

- 从函数build_dataset()中给words中出现过的单词做频数统计，取vocavulary_size-1的放入字典

```
def build_dataset(words):
  count = [['UNK', -1]]
  count.extend(collections.Counter(words).most_common(vocabulary_size - 1))
  dictionary = dict()
  for word, _ in count:
    dictionary[word] = len(dictionary)
  data = list()
  unk_count = 0
  for word in words:
    if word in dictionary:
      index = dictionary[word]
    else:
      index = 0
      unk_count += 1
    data.append(index)
  count[0][1] = unk_count
  reverse_dictionary = dict(zip(dictionary.values(), dictionary.keys()))
  return data, count, dictionary, reverse_dictionary

data, count, dictionary, reverse_dictionary = build_dataset(words)
```

- 从generate_batch()函数中使用skip-Gram模式来生成Word2vec训练样本

```
def generate_batch(batch_size, num_skips, skip_window):
  global data_index
  assert batch_size % num_skips == 0
  assert num_skips <= 2 * skip_window
  batch = np.ndarray(shape=(batch_size), dtype=np.int32)
  labels = np.ndarray(shape=(batch_size, 1), dtype=np.int32)
  span = 2 * skip_window + 1 # [ skip_window target skip_window ]
  buffer = collections.deque(maxlen=span)
  for _ in range(span):
    buffer.append(data[data_index])
    data_index = (data_index + 1) % len(data)
  for i in range(batch_size // num_skips):
    target = skip_window  # target label at the center of the buffer
    targets_to_avoid = [ skip_window ]
    for j in range(num_skips):
      while target in targets_to_avoid:
        target = random.randint(0, span - 1)
      targets_to_avoid.append(target)
      batch[i * num_skips + j] = buffer[skip_window]
      labels[i * num_skips + j, 0] = buffer[target]
    buffer.append(data[data_index])
    data_index = (data_index + 1) % len(data)
  return batch, labels

batch, labels = generate_batch(batch_size=8, num_skips=2, skip_window=1)
```

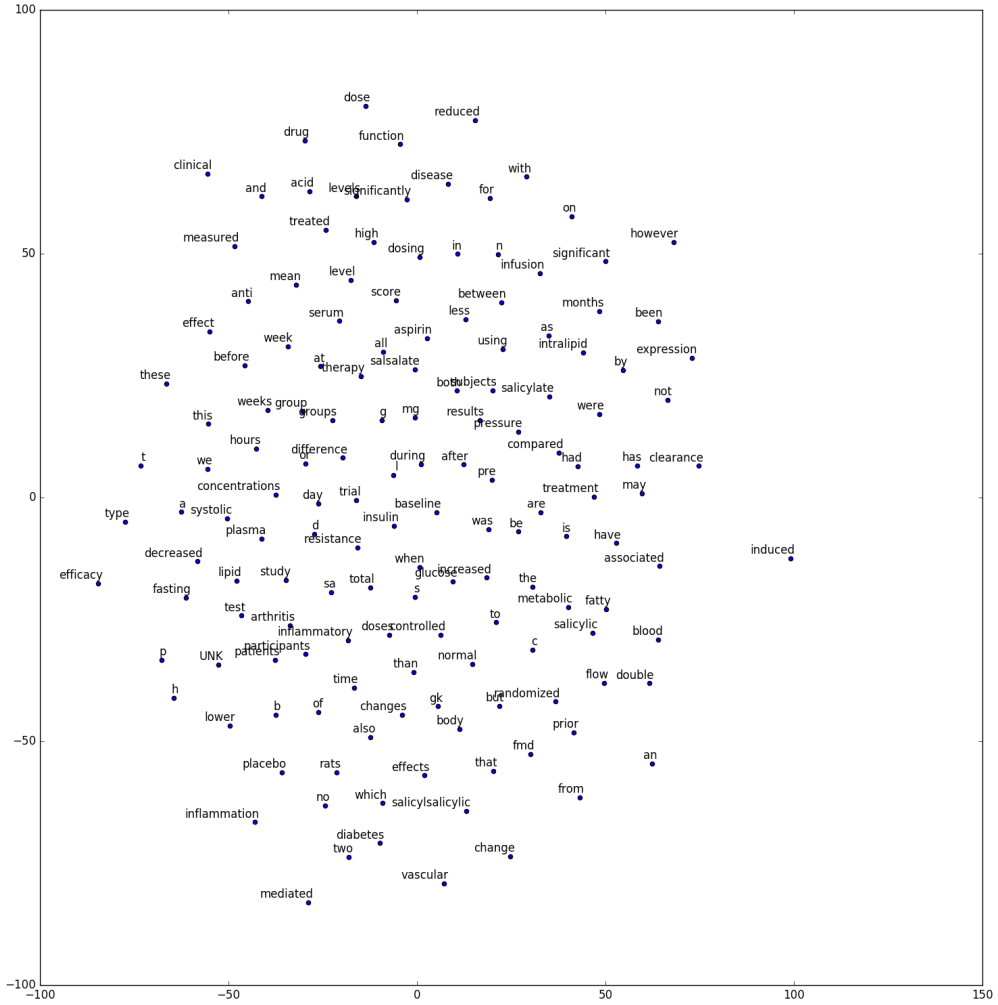
batch_size:每个批次训练的样本数
Num_skips:为每个单词生成多少个样本
skip_window:单词最远可以联系的距离

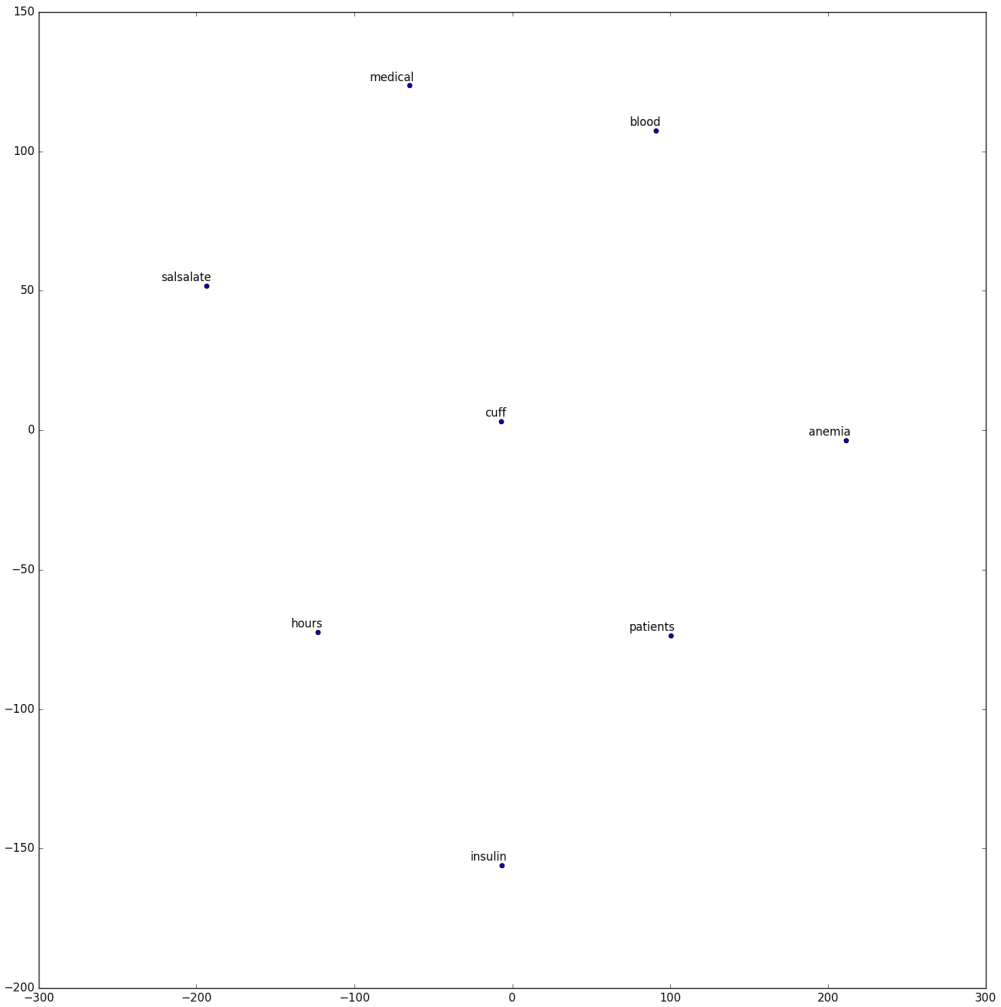
- 然后是训练的过程（代码略）,num_step 由101改为100001最后返回的final_embeddings表示的是词向量的字典

- 可视化。Plot_with_labels()
- Tsne降维：将原始的嵌入向量降到2维

```
from sklearn.manifold import TSNE
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
tsne = TSNE(perplexity=30, n_components=2, init='pca', n_iter=5000)
#plot_only = 250
#low_dim_embs = tsne.fit_transform(final_embeddings[:plot_only, :])
#labels = [reverse_dictionary[i] for i in xrange(plot_only)]
#plot_with_labels(low_dim_embs, labels)
w=['salsalate','cuff','hours','blood','patients','medical','insulin','anemia']
i=0
ws= [[0] * 50] * 8
for j in w:
    #ws[i].append(final_embeddings[dictionary[j]])
    ws[i] = final_embeddings[dictionary[j]]
    i=i+1
low_dim_embs = tsne.fit_transform(ws)
labels=w
plot_with_labels(low_dim_embs, labels)
```

Result:


100
50
0
−50
−100
−100
−50
0
50
100
150
dose
reduced
drug
function
clinical
disease
with
and
acid
levels
significantly
for
on
treated
high
however
measured
dosing
in
n
significant
infusion
mean
level
anti
score
between
months
been
serum
less
effect
aspirin
as
week
all
using
intralipid
expression
before
at
therapy
salsalate
by
these
both
subjects
salicylate
not
weeks
group
groups
g
mg
results
were
this
pressure
hours
difference
compared
t
we
of
during
after
had
has
clearance
l
pre
concentrations
day
trial
treatment
may
a
systolic
baseline
are
type
insulin
was
be
is
plasma
d
have
resistance
induced
decreased
when
associated
efficacy
lipid
study
sa
total
increased
glucose
the
fasting
s
metabolic
fatty
test
arthritis
to
salicylic
inflammatory
doses
controlled
blood
p
UNK
participants
patients
normal
c
than
flow
double
time
h
randomized
gk
b
of
changes
body
lower
prior
also
fmd
placebo
rats
effects
that
an
no
which
from
salicylsalicylic
inflammation
diabetes
two
change
vascular
mediated



Cuff:袖口状白血球聚集

Insulin:胰岛素，文章一般和resistance一起表示抗胰岛性

Anemia:贫血症

哈利波特：

- 文本大小：6.2MB　　Words:1122750

- 来源于：

https://github.com/LouisScorpio/datamining/blob/master/tensorflow-program/nlp/word2vec/code/word2vec_harrypotter.py

- Windows: python 3.5; tensorflow-0.12.1;num_step:1000001