# 601.465: Homework 3

Tingwen Guo

October 11, 2019

## 1 Question 1

### 1.1 Test with d=50

The most similar words with **Seattle** are:

seahawks
dallas
atlanta
wichita
tacoma
lauderdale
florida
spokane
dulles
chino
philadelphia

The most similar words with **dog** are:

dogs
badger
cat
hound
puppy
dachshund
sighthound
rat
poodle
keeshond
canine

The most similar words with **communist** are:

socialist
bolshevik
communists
comintern

trotskyist
leftist
bolsheviks
stalinist
leninist
communism
cominform

The most similar words with **jpg** are:

png
svg
szczepanek
buteo
gif
pix
image
galleria
regnum
fiav
hopewell

The most similar words with **the** are:

in
its
of
which
entire
within
from
a
second
part
third

The most similar words with **google** are:

webpage
yahoo
web
search
com
faq
blogging
archived
editable
redirection
geocaching

The most similar words with **boeing** are:

airbus
convair
widebody
lockheed
airliner
varig
aerospatiale
aircraft
airlines
mitsubishi
mcdonnell

The most similar words with **hopkins** are:

johns
palmer
grimwood
manley
clairvoyant
stillman
quimby
lightnin
jasper
bront
swanton

'hopkins', 'boeing', 'google', 'jpg' work poorly; 'seattle', 'dog', 'communist', 'the' work relatively good. There are two common traits shared by these poor words:
1. They are proper noun, therefore there are only very few words that are similar. Particularly for 'hopkins', only the output 'johns' makes sense because 'hopins' only refers to our school.
2. They rarely appear in texts. This may lead to lack of samples for word2vec.
On the other hand, the 'good' words appear frequently, and are description of a wide range of objects in the real word. They belongs to a big class of words, for example, **location** or **animal**.

## 1.2 Test with more dimensions

### 1.2.1 word-10.txt

The most similar words with **seattle** are:

indianapolis
atlanta
lakers
dallas
boston
expos
detroit
cleveland
houston
texas
milwaukee

The most similar words with **dog** are:

turnip
ass
pig
eyed
cow
coronets
haired
embroidered
melon
goat
unicorns

The most similar words with **communist** are:

socialist
rightist
communists
fascist
comintern
bolshevik
revolutionary
leftist
instigated
reichswehr
reestablish

The most similar words with **jpg** are:

jpg
hout
maui
storey
lsch
hino
monte
ledger
bahnhof
eau
longship
cru

The most similar words with **the** are:

the
marked
successive
gradually
split
reintroduced

contention
sway
ceased
changed
since
intervening

The most similar words with **google** are:

info
geocaching
downloadable
archiving
web
com
digitized
printing
listings
bitnet
format

### 1.2.2 word-200.txt

The most similar words with **seattle** are:

tacoma
spokane
seahawks
redskins
intelligencer
dulles
bremerton
denzel
dc
dallas
yamasaki

The most similar words with **dog** are:

dogs
hound
inu
komondor
sighthound
mastiff
spaniel
terrier
dachshund
badger
borzoi

The most similar words with **communist** are:

socialist
comintern
bolshevik
communists
pdpa
trotskyist
stalinist
marxist
leninist
leftist
soviet

The most similar words with **jpg** are:

png
gif
svg
modis
buteo
ltspkr
szczepanek
spitting
regnum
skyview
antoninianus

The most similar words with **the** are:

of
its
which
in
a
this
itself
thus
entire
second
and

The most similar words with **google** are:

yahoo
msn
gmail
archived
geocaching
pagerank
mapquest

search
maps
usenet
web

#### 1.2.3 Observation

We can see that the performance of d=10 is very bad; most outputs generated are incorrect. When d==200, the performance also decreases, compared to that of d=50. d=50, as a midpoint, is a local maximum of performance.

### 1.3 Extra Credit

The samples we use are:
1. war - russia + germany
2. food - meat + wheat
3. building - house + skyscraper
4. napkin - dirt + paper
5. kitchen - knife + spoon

1. war - russia + germany:

war
germany
allied
wwi
wwii
ww
occupation
luftwaffe
nazi
rearmament
wehrmacht
hostilities

2. food - meat + wheat

wheat
food
cereals
dryland
agricultural
crops
crop
barley
quinoa
cassava
harvested
sorghum

3. building - house + skyscraper

building
skyscraper
skyscrapers
construction
buildings
constructed
spire
built
tallest
deco
towers
plaza

4. napkin - dirt + paper

paper
papers
entscheidungsproblem essay
periodical
printing
pen
pamphlet
seminar
journals
lecture
anonymously

5. kitchen - knife + spoon

kitchen
breakfast
dining
restaurant
spoon
lunch
guests
breakfasts
cellar
caf
downstairs
oven

We can observe that number 1, 2, 3 works well, but 4, 5 works poorly. Now we can try d=10:
1. war - russia + germany:

war
led
participated
soviet

uprising
ussr
petrograd
falklands
deng
campaign
culminating
campaigns

2. food - meat + wheat

foodstuffs
ecotourism
canneries
lumber
crop
handicrafts
foodcrops
conservancy
crops
earner
fertilizers
bananas

3. building - house + skyscraper

satellite
lighthouses
observatories
construction
kitt
intelsat
shunters
shelf
intersputnik
vectra
shf
mondeo

4. napkin - dirt + paper

endnotes
textbook
mcluhan
review
mit
fundamentals
transcript
symposium
outlining

addison
kluwer
math

5. kitchen - knife + spoon

infirmary
picnic
gras
aberfoyle
glasnevin
hostels
stillwater
hostel
dorms
humberside
lawn
dormitory

With d=10, all the samples work very poorly.
Since all words are represented by vectors, subtraction and minus can be interpreted as removing traits and adding traits. For example, **king - man**, in the vector space, removes the dimensions that 'man' has, and thus pushes the vector as a whole towards the direction opposite from that of 'man'. Similarly, adding in 'woman' is just adding the traits of **king-man** to 'woman', which finally results in a vector that is analogous to 'kings who are women but not men'.

# 2 Question 2

## 2.1 1

$$cross\_entropy\_sample1 = -\frac{1}{k} log_2 P(sample1) = -\frac{1}{1686}(-12121) = 7.189$$
$$perplexity\_sample1 = 2^{7.189} = 145.91$$

$$cross\_entropy\_sample2 = -\frac{1}{k} log_2 P(sample2) = -\frac{1}{978}(-7398.55) = 7.564$$
$$perplexity\_sample2 = 2^{7.564} = 189.23$$

$$cross\_entropy\_sample3 = -\frac{1}{k} log_2 P(sample3) = -\frac{1}{985}(-7477.99) = 7.592$$
$$perplexity\_sample3 = 2^{7.592} = 192.93$$

If we use larger switchboard corpus, the log 2 probability decreases, cross-entropy, and perplexity increases. Since $log(P(sample)) = \sum_k log(P(w_k))$, larger corpus means larger k, and thus will introduce more terms into the summation, make the total log probability smaller, and thus cross entropy and perplexity increases.

### 2.2 3

#### 2.2.1 a

Among all the /dev/gen files, 59 were classified as gen and 121 were classified as spam, so the error rate is 0.672.
Among all the /dev/spam files, 12 files were classified as gen, and 78 were classified as spam, the error rate is 0.067.
The error rate in total is 0.369.

#### 2.2.2 b

$10^{-72}$

#### 2.2.3 c

**Gen** The total minimum cross-entropy is 3.627, achieved at $\lambda = 3.627$. There are 100838 tokens, so the cross-entropy per token is $3.607 \times e^{-5}$.

**Spam** The total minimum cross-entropy is 7.109, achieved at $\lambda = 0.01$. There are 22994 tokens, the cross-entropy per token is 0.000322.

#### 2.2.4 d

The total minimum entropy per token is 0.00008694, achieved at $\lambda = 0.01$.

#### 2.2.5 e

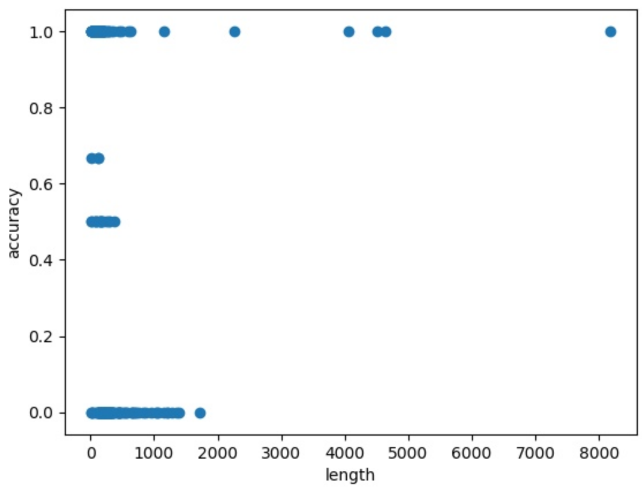


Figure 1: test file length vs accuracy

We can observe that the accuracy increases while the length of test files increases.

#### 2.2.6 f

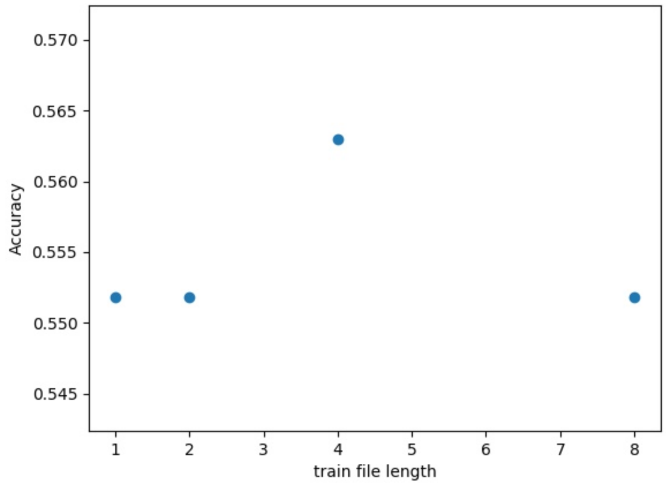


Figure 2: Training file length vs accuracy

We can observe that the accuracy doesn't changes with the length of training files.

### 2.3 4

#### 2.3.1 a

The sum of all $p(z \mid xy)$ should be one. If we take $V = 19999$ then $\sum_x \sum_y p(z \mid xy) > 1 \implies p(z|xy)$ is no longer a probability distribution.
For UNIFORM, the change from $\frac{1}{20000}^n$ to $\frac{1}{19999}^n$ will make the log probability of each file larger than it should be.
For ADDL, the change from $\frac{1}{20000}^n$ to $\frac{1}{19999}^n$ will make the denominator larger, thus will make the log probability of each file smaller.

#### 2.3.2 b

Naive historical estimate will result in over-fitting. For example, if $c(x_0y_0z_0) = 0 \implies p(z_0 \mid y_0x_0)$ for some $z_0, x_0, y_0$ in gen, then the naive historical estimate will conclude that any email with a token $x_0y_0z_0$ in it is impossible to be gen, which can lead to bad performance in the test set.

### 2.3.3 c

$$
\begin{aligned}
c(xyz) &= c(xyz') = 0 \\
\implies p(z \mid xy) = \frac{\lambda p(z \mid y)}{c(xy) + \lambda V}, p(z' \mid xy) &= \frac{\lambda p(z' \mid y)}{c(xy) + \lambda V} \\
\implies \text{if } p(z \mid y) \neq p(z' \mid y), \text{then } p(z \mid xy) &\neq p(z' \mid xy)
\end{aligned}
$$

$$
\begin{aligned}
c(xyz) &= c(xyz') = 1 \\
\implies p(z \mid xy) = \frac{1 + \lambda p(z \mid y)}{c(xy) + \lambda V}, p(z' \mid xy) &= \frac{1 + \lambda p(z' \mid y)}{c(xy) + \lambda V} \\
\implies \text{if } p(z \mid y) \neq p(z' \mid y), \text{then } p(z \mid xy) &\neq p(z' \mid xy)
\end{aligned}
$$

We can see that when $c(xyz) = c(xyz') = 1 \textit{ or } 0$, the trigram probability becomes bigram probability.

### 2.3.4 d

Analogous to the conclusion we have for (c), when $\lambda$ becomes large, the first term, $c(xyz)$, becomes negligible relative to the scale of $\lambda p(z \mid y)$. Thus $p(z \mid xy)$ becomes a bigram probability. It's value converges to $\frac{p(z'|y)}{V}$.

## 2.4 6

### 2.4.1

The log probabilities when C = 1 are from -112 to -52, when trained with chars-10.txt and en.1k. The cross-entropy thus ranges from 11.2 to 5.2. I also tested with C = 0.5, 0.4, 0.02, 0.01, and the best I found is 0.5, which gives cross entropy from 6.9 to 4.1. There is a significant improvement in croos-entropy comparing to $add - \lambda$ model.

### 2.4.2

I did the unigram log-probability. The improvement is significant. The unigram log-probability model gives an average cross-entropy of 5.2.