# Introduction to Natural Language Processing

## Assignment 4

Ankit Satpute 120825, Gabriel Becker 120770, Ukasha Ali 120798.

July 2020

| Quote (1) | Code | en-word | Quote (2) | Code | en-word | Quote (3) | Code | en-word |
|---|---|---|---|---|---|---|---|---|
| do | Make %2:41:00 | Make %2:41:00 | how | - | - | it | Information_ technology %1:09:00 | - |
| you | - | - | it | Information _technology %1:09:00 | - | is | Be %2:42:03 | Be %2:42:03 |
| train | Train %2:31:01 | Train %2:31:01 | is | Be %2:42:03 | Be %2:42:03 | the | - | - |
| for | - | - | we | - | - | responsibility | Duty %1:04:00 | Duty %1:04:00 |
| passing | Pass %2:38:00 | Pass %2:38:00 | have | Have %2:40:00 | Have %2:40:00 | of | - | - |
| tests | Trial %1:09:00 | Trial %1:09:00 | so | - | So %4:02:02 | intellectuals | Intellectual %1:18:00 | Intellectual %1:18:00 |
| or | - | - | much | Much %3:00:00 | Much %3:00:00 | to | - | - |
| do | Make %2:41:00 | Make %2:41:00 | information | Information %1:10:00 | Information %1:10:00 | speak | Talk %2:32:00 | Talk %2:32:00 |
| you | - | - | , | - | - | the | - | - |
| train | Train %2:31:01 | Train %2:31:01 | but | - | - | truth | - | Truth %1:09:00 |
| for | - | - | know | - | Know %2:31:01 | and | - | - |
| creative | Creative %3:00:00 | Creative %5:00:00 | so | - | So %4:02:02 | expose | Expose %2:39:02 | Expose %2:39:02 |
| inquiry | Inquiry %1:09:01 | Inquiry %1:09:01 | little | Small l%3:00:00 | Little %5:00:00 | lies | - | Lie %1:10:00 |

We are considering like relevant only the words that retrieve a Sense key for the recall calculation

| Quote | Total | Correct | Relevant | Correct | Accuracy | Recall | F1 |
|---|---|---|---|---|---|---|---|
| 1 | 13 | 7 | 8 | 7 | 7/13=0.5385 | ⅞=0.875 | 0.6667 |
| 2 | 13 | 4 | 9 | 4 | 4/13=0.3077 | 4/9=0.4444 | 0.3636 |
| 3 | 13 | 5 | 8 | 5 | 5/130.3846 | ⅝=0.625 | 0.4762 |

[Synset('information_technology.n.01'), Synset('be.v.01'), Synset('duty.n.01'), Synset('intellectual.n.01'), Synset('talk.v.02'), Synset('expose.v.01')]
[Lemma('information_technology.n.01.information_technology'), Lemma('be.v.01.be'), Lemma('duty.n.01.duty'), Lemma('intellectual.n.01.intellectual'), Lemma('talk.v.02.talk'), Lemma('expose.v.01.expose')]
['information_technology%1:09:00::', 'be%2:42:03::', 'duty%1:04:00::', 'intellectual%1:18:00::', 'talk%2:32:00::', 'expose%2:39:02::']

[Synset('information_technology.n.01'), Synset('be.v.01'), Synset('have.v.01'), Synset('much.a.01'), Synset('information.n.01'), Synset('small.a.01')]
[Lemma('information_technology.n.01.information_technology'), Lemma('be.v.01.be'), Lemma('have.v.01.have'), Lemma('much.a.01.much'), Lemma('information.n.01.information'), Lemma('small.a.01.small')]
['information_technology%1:09:00::', 'be%2:42:03::', 'have%2:40:00::', 'much%3:00:00::', 'information%1:10:00::', 'small%3:00:00::']

[Synset('make.v.01'), Synset('train.v.01'), Synset('pass.v.01'), Synset('trial.n.02'), Synset('creative.a.01'), Synset('inquiry.n.01')]
[Lemma('make.v.01.make'), Lemma('train.v.01.train'), Lemma('pass.v.01.pass'), Lemma('trial.n.02.trial'), Lemma('creative.a.01.creative'), Lemma('inquiry.n.01.inquiry')]
['make%2:41:00::', 'train%2:31:01::', 'pass%2:38:00::', 'trial%1:09:00::', 'creative%3:00:00::', 'inquiry%1:09:01::']

**Question # 2**
**Part a**

| | Does | Racing | Zelda | Fighting | Nintendo | Is | Red |
|---|---|---|---|---|---|---|---|
| Does | 0 | 0 | 9 | 1 | 1 | 16 | 0 |
| Racing | 0 | 2 | 0 | 1 | 1 | 17 | 0 |
| Zelda | 9 | 0 | 32 | 0 | 10 | 112 | 0 |
| Fighting | 1 | 1 | 0 | 0 | 1 | 23 | 0 |
| Nintendo | 1 | 1 | 10 | 1 | 6 | 45 | 1 |
| Is | 16 | 17 | 112 | 23 | 45 | 430 | 11 |
| Red | 0 | 0 | 0 | 0 | 1 | 11 | 4 |

**Part c**

**(c1) For each given word, determine the three words which are closest in the embedding space, i.e. the three most similar words:**

| **Word** | **PCA model** | **Wiki-50d** |
|---|---|---|
| Playstation | ‘n64’, ‘tomb’, ‘raider’ | 'xbox', 'nintendo', 'gamecube' |
| Walk | 'travel', 'collect', 'explore' | 'walking', 'walks', 'walked' |
| Love | 'enjoy', 'playing', 'got' | 'dream', 'life', 'dreams' |

**(c2) For each of the given sets of words, let the models determine the outlier:**

| **Set of words** | **Truth** | **PCA model** | **Wiki-50d** |
|---|---|---|---|
| (mario, zelda, kirby, microsoft) | Microsoft | mario | microsoft |
| (january, july, december, year) | year | year | year |

**(c3) Determine the word that completes the following anaologies.**

| **Set of words** | **Truth** | **PCA model** | **Wiki-50d** |
|---|---|---|---|
| son:daughter :: boy:? | girl | nearly | girl |
| n64:nintendo :: playstation:? | sony | n64 | xbox |
| acceptable:unacceptable :: reasonable:? | unreasonable | mesmerized | acceptable |
| good:better :: tough:? | tougher | anyone | hard |
| go:going :: play:? | playing | own | playing |

## Task(d)

We have tried to improve our vector embeddings by using larger dimensions (100d) and changing some parameters in PCA like what we read in documentation of PCA that setting value of random_state either 0 or 42 have optimally resulted in good results so this time we ran our code with value 0. We have repeated the evaluation and here are the results:

1. 3-words which are close to (playstation) in the embedding space word2vec10d are : ['n64', 'tomb', 'raider']

2. 3-words which are close to (walk) in the embedding space word2vec10d are :  ['travel', 'collect', 'explore']
3. 3-words which are close to (love) in the embedding space word2vec10d are :  ['enjoy', 'playing', 'got']
4. Outliner found with using w2v50d is : mario
5. Outliner found with using w2v50d is : year
6. With using w2v50d :- boy : nearly
7. With using w2v50d :- playstation : n64
8. With using w2v50d :- reasonable : mesmerized
9. With using w2v50d :- tough : anyone
10. With using w2v50d :- play : own