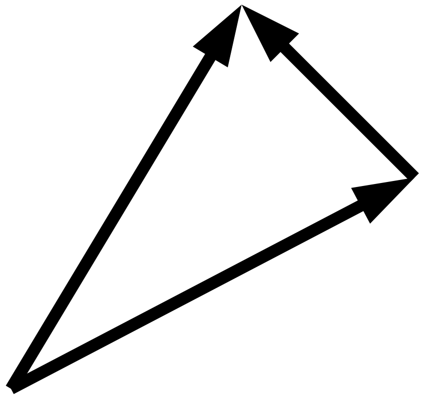
(딥러닝 실습)
Vector Representations of Words

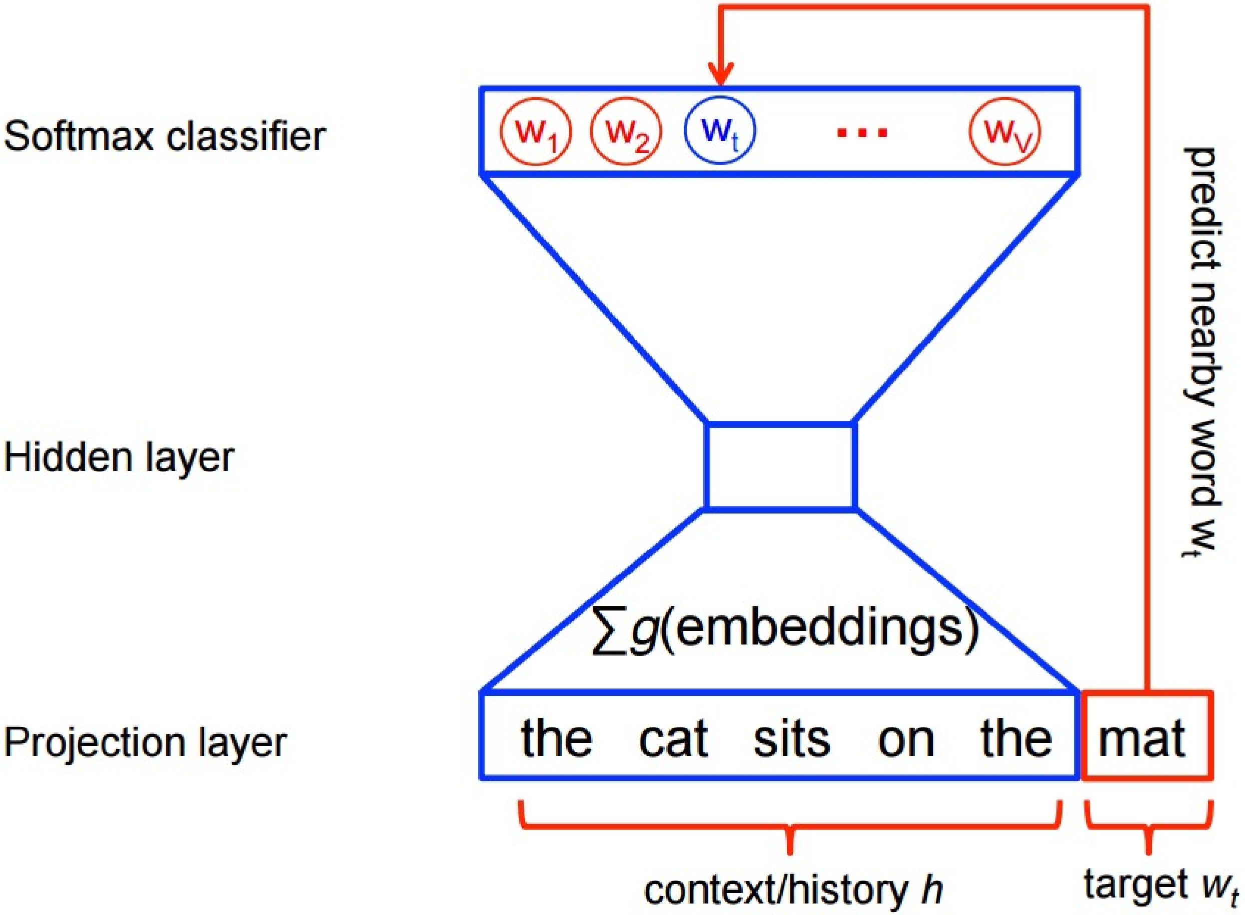
vec(“King”) - vec(“Man”) + vec(“Woman”) = ?



한성국 /2016-4-19

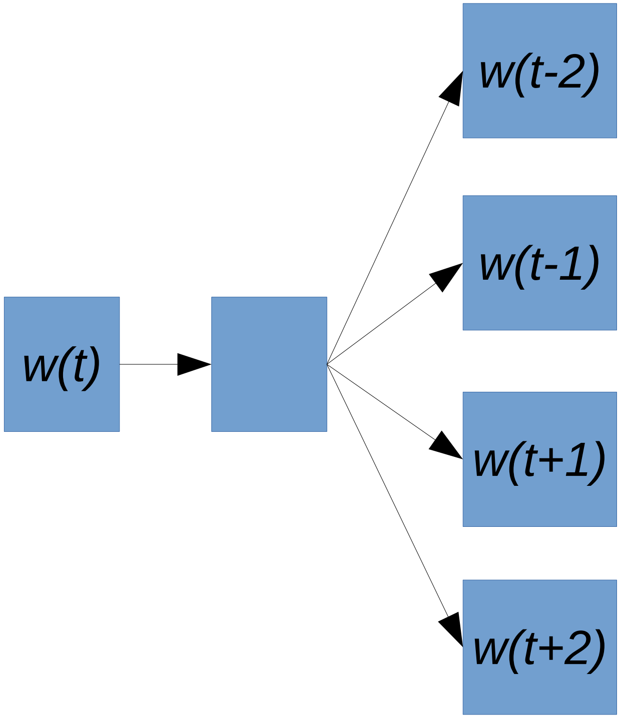
- Skip-gram model
  - Full softmax
  - Likelihood: Hierarchical softmax
  - NCE(Noise Constrastive Estimation)
  - Negative Sampling
- Tensorflow: word2vec.py

Skip-gram model: Classification



W_O W_I

Skip-gram model: Learning



p(w(t-2)|w(t))

p(w(t-1)|w(t))

p(w(t+1)|w(t))

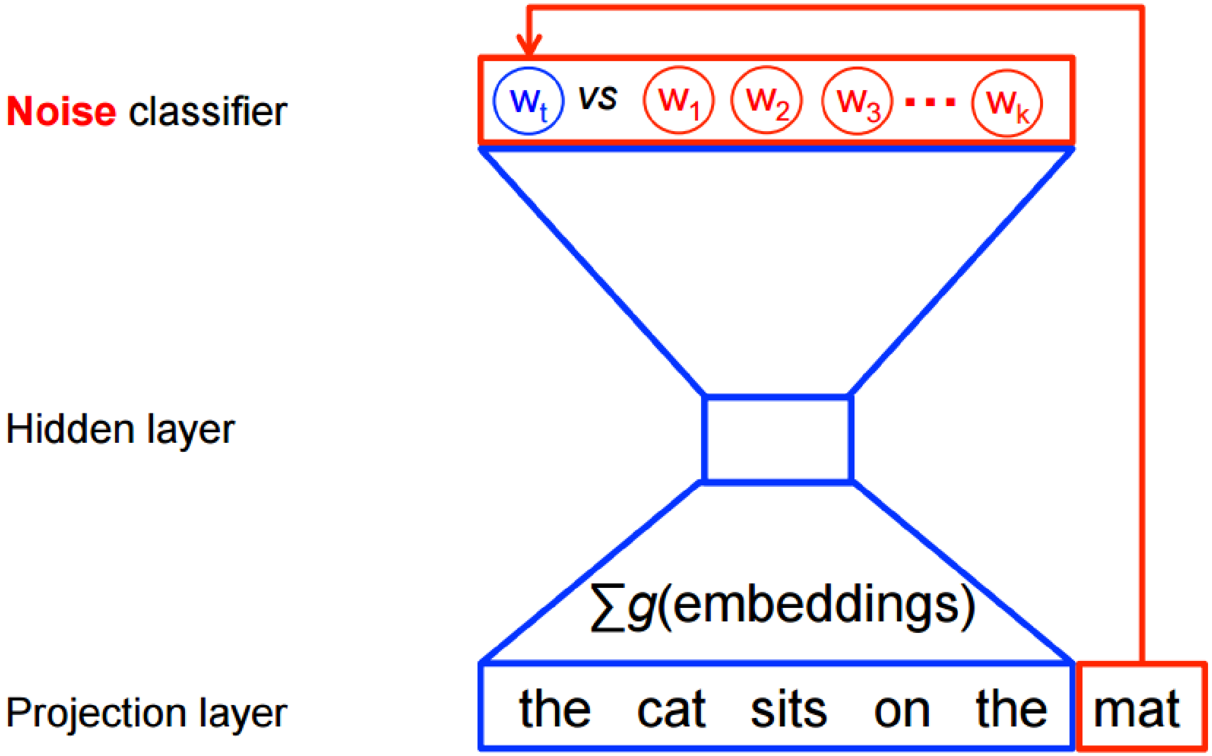
p(w(t+2)|w(t))

$$\text{new } w(t) = w(t) + \alpha \nabla J \begin{pmatrix} \blacktriangledown p(w(t-2)|w(t)) \\ \blacktriangledown p(w(t-1)|w(t)) \\ \blacktriangledown p(w(t+1)|w(t)) \\ \blacktriangledown p(w(t+2)|w(t)) \end{pmatrix}$$

Inefficient to train

- Training words (vocabulary): w1, w2, w3,…,wN.
- Loss $$J=\sum_{t=1}^{N}\sum_{-c\leqslant j\leqslant c,\, j\neq 0}\log p(w_{t+j}|w_t)$$
- Softmax: $p(w_O|w_I)=\frac{\exp({v'_{w_O}}^T v_{w_I})}{\sum_{w=1}^{W}\exp({v'_w}^T v_{w_I})}$
- To slowly train since the computational cost of $\nabla\log p(w_O|w_I)\sim O(N)$

Noise Contrastive Estimation(NCE)



- Not all words, just k words $\{w_1, w_2, \ldots, w_k\}$.
- Randomly pick k words {w1, w2, w3,…,w_k} from P_{noise}
- Logistic regression
- Loss function

An Instance of training the Skip-gram model

The quick brown fox jumped over the lazy dog

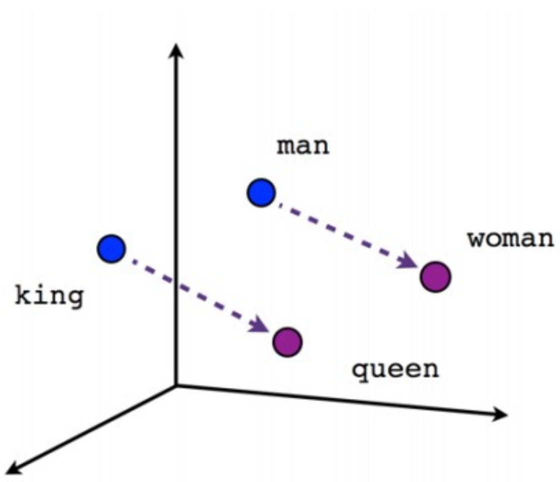
- Three words context
  - ([the, brown], quick)
  - ([quick,fox],brown)
  - ([brown, jumped], fox)
  - …
- Data set:

  (input, output)=(target,context)
  - (quick,the)
  - (quick,brown)
  - (brown, quick)
  - (brown,fox)
  - (fox, brown)
  - (fox,jumped)

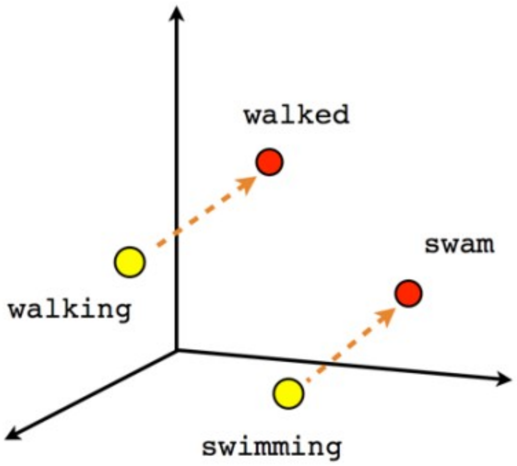
- Target: 'quick', context: 'the'
- A noise word 'sheep' taken from the unigram distribution: P(w).
- Sigmoid activation function

$$J^t = \log \underbrace{Q_\theta(D=1|the\,, quick)}_{\displaystyle\frac{1}{1+\exp(-x)}} + \log \underbrace{Q_\theta(D=0|sheep\,, quick)}_{\displaystyle\frac{1}{1+\exp(x)}}$$

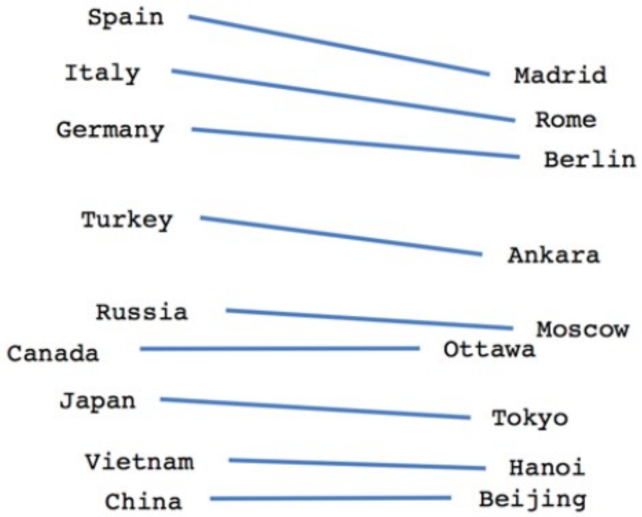
Well represented(learned) vectors



Male-Female



Verb tense



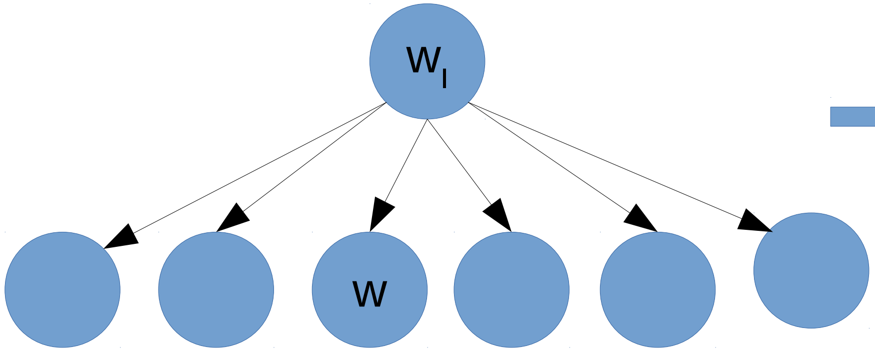
Country-Capital

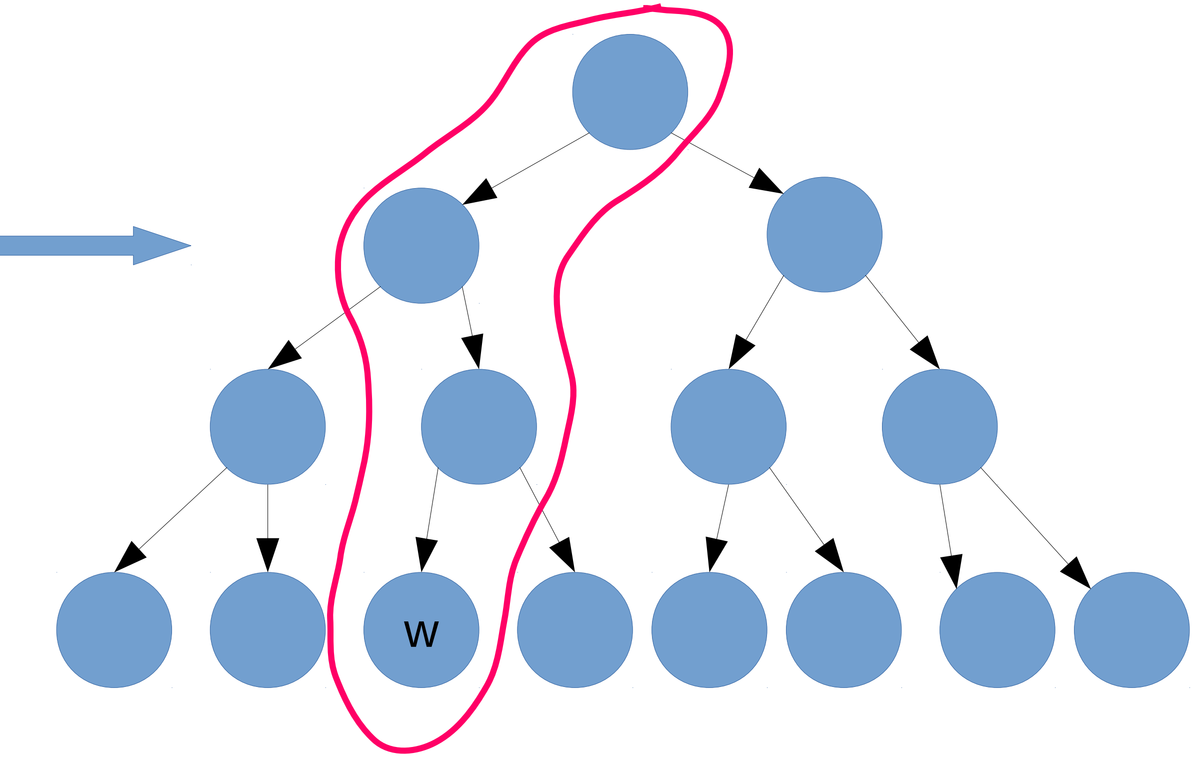
Loss1 and Loss2:
NEG and NCE (ref. Mnih(2013))

Loss function of NEG

$$\log \sigma \left({v'_{w_O}}^T v_{w_I} \right) + \sum_{i=1}^{k} \mathrm{E}_{w_i \sim P_n(w)} \left[\log \sigma \left(-{v'_{w_i}}^T v_{w_I} \right) \right]$$

Loss3: Hierarchical softmax





$$p(w_O|w_I)=\frac{\exp\left(v'^{\,T}_{w_O} v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left(v'^{\,T}_{w} v_{w_I}\right)}$$

~ O(w)

$$p(w|w_I)=\prod_{j=1}^{L(w)-1}\sigma\left([\![n(w,j+1)=ch(n(w,j))]\!]\, v'^{\,T}_{n(w,j)} v_{w_I}\right)$$

~ O(Log(W))

Accuracy of various Skip-gram 300-dimensional models

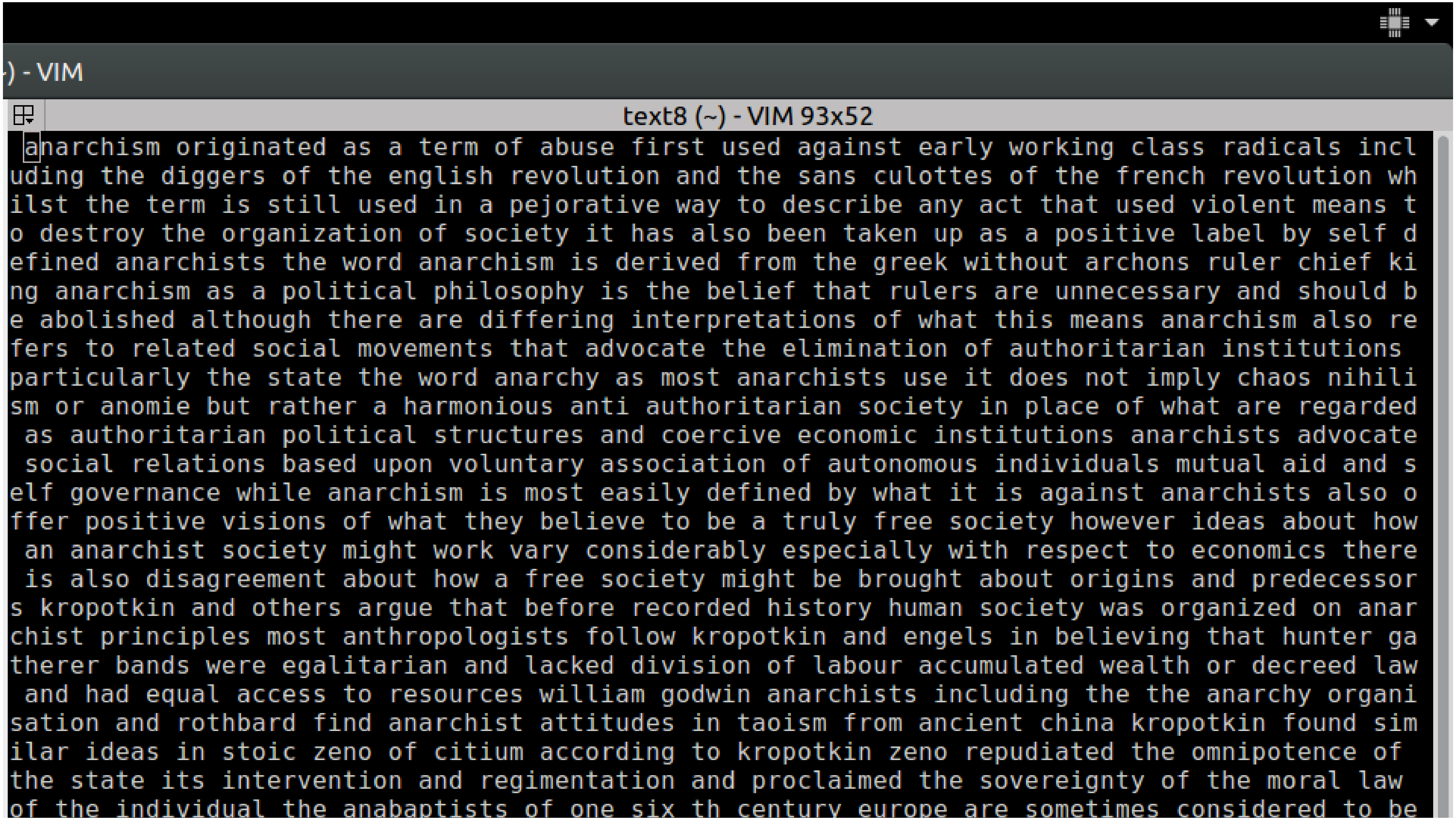
Method	Time [min]	Syntactic[%]	Semantic[%]	Total accuracy
NEG-5	38	63	54	59
NEG-15	97	63	58	**61**
HS-Huffman	41	53	40	47
NCE-5	38	60	45	53
The following results use 10 subsampling				
NEG-5	14	61	58	60
NEG-15	36	61	61	**61**
HS-Huffman	21	52	59	55

Mikolov et. al. 2013

How to training the Skip-gram model

- Preparation of data set: batch and label
- Hyper parameters:
- Loss functions:
  - tf.nn.nce_loss(), tf.nn.sampled_softmax()
- Visualization of embeddings: t-SNE

Data set: Text corpus



```
anarchism originated as a term of abuse first used against early working class radicals incl
uding the diggers of the english revolution and the sans culottes of the french revolution wh
ilst the term is still used in a pejorative way to describe any act that used violent means t
o destroy the organization of society it has also been taken up as a positive label by self d
efined anarchists the word anarchism is derived from the greek without archons ruler chief ki
ng anarchism as a political philosophy is the belief that rulers are unnecessary and should b
e abolished although there are differing interpretations of what this means anarchism also re
fers to related social movements that advocate the elimination of authoritarian institutions 
particularly the state the word anarchy as most anarchists use it does not imply chaos nihili
sm or anomie but rather a harmonious anti authoritarian society in place of what are regarded
 as authoritarian political structures and coercive economic institutions anarchists advocate
 social relations based upon voluntary association of autonomous individuals mutual aid and s
elf governance while anarchism is most easily defined by what it is against anarchists also o
ffer positive visions of what they believe to be a truly free society however ideas about how
 an anarchist society might work vary considerably especially with respect to economics there
 is also disagreement about how a free society might be brought about origins and predecessor
s kropotkin and others argue that before recorded history human society was organized on anar
chist principles most anthropologists follow kropotkin and engels in believing that hunter ga
therer bands were egalitarian and lacked division of labour accumulated wealth or decreed law
 and had equal access to resources william godwin anarchists including the the anarchy organi
sation and rothbard find anarchist attitudes in taoism from ancient china kropotkin found sim
ilar ideas in stoic zeno of citium according to kropotkin zeno repudiated the omnipotence of 
the state its intervention and regimentation and proclaimed the sovereignty of the moral law 
of the individual the anabaptists of one six th century europe are sometimes considered to be
```

- http://mattmahoney.net/dc/textdata

Preparation for training data set

- Total number of words 17,005,207
- Vocabulary size: 50,000
- Learning the nearest words (w_{t-1}, w_t, w_{t+1})

- batch and label
  - $w_t \rightarrow w_{t-1}$
  - $w_t \rightarrow w_{t+1}$

batch[i]=index(w_t)

label[i]=index(w_{t-1})

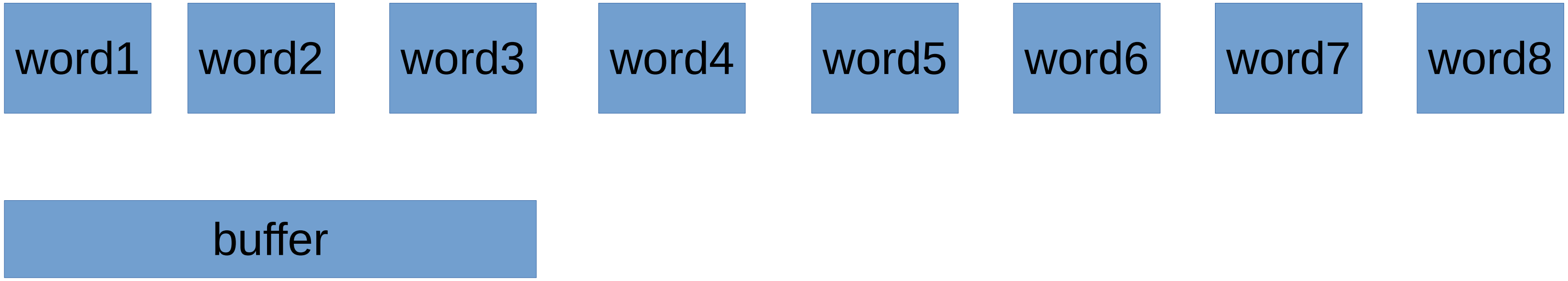
batch[i+1]=index(w_t)

label[i+1]=index(w_{t+1})

Hyper parameters for training

- Batch size =128
- Embedding size (dimension of word vector) = 128
- skip_window = 1
- num_skips=2
- Loss function: tf.nn.nce_loss()

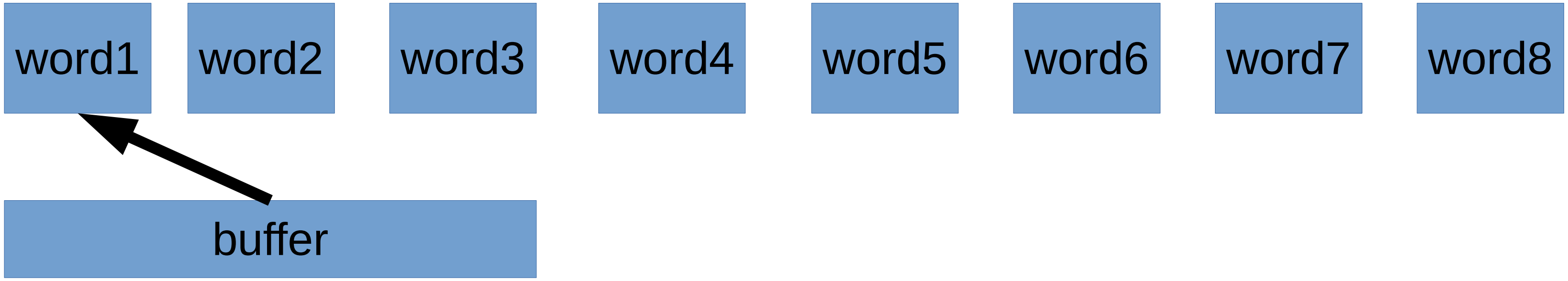
Build data set of batch and label



word1 word2 word3 word4 word5 word6 word7 word8

buffer

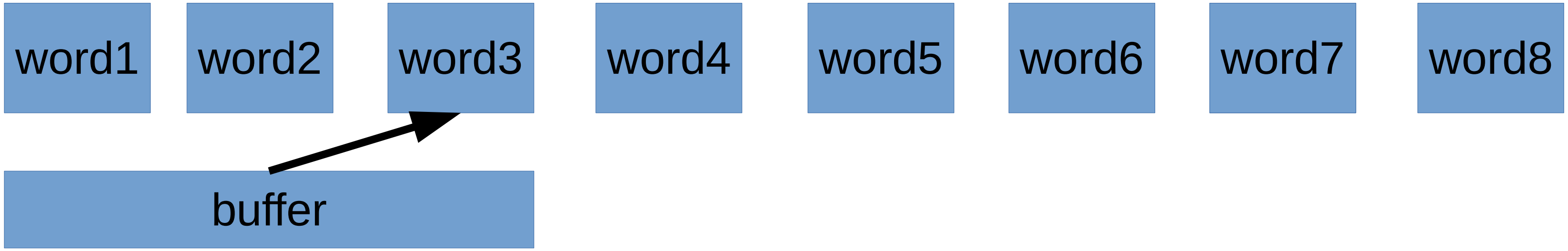
Build data set of batch and label



batch[i]=index(word2)

label[i]=index(word1)

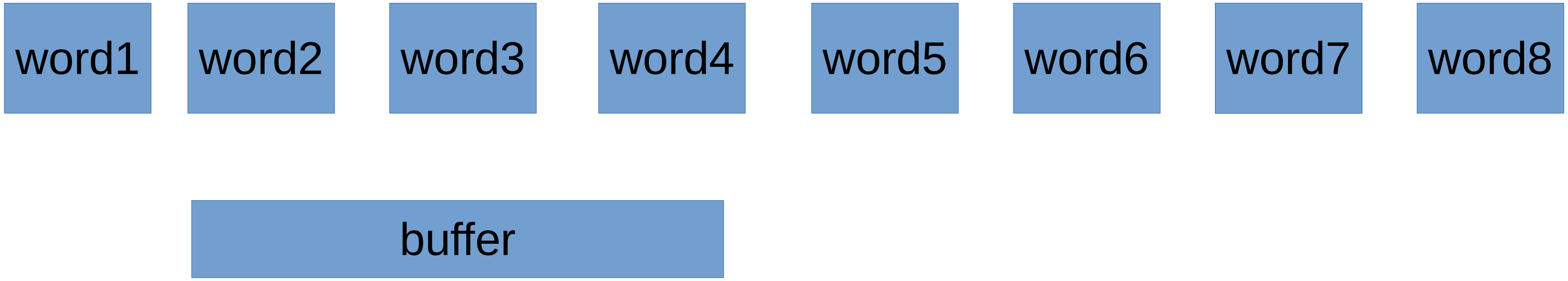
Build data set of batch and label



batch[i]=index(word2)
label[i]=index(word1)

batch[i+1]=index(word2)
label[i+1]=index(word3)

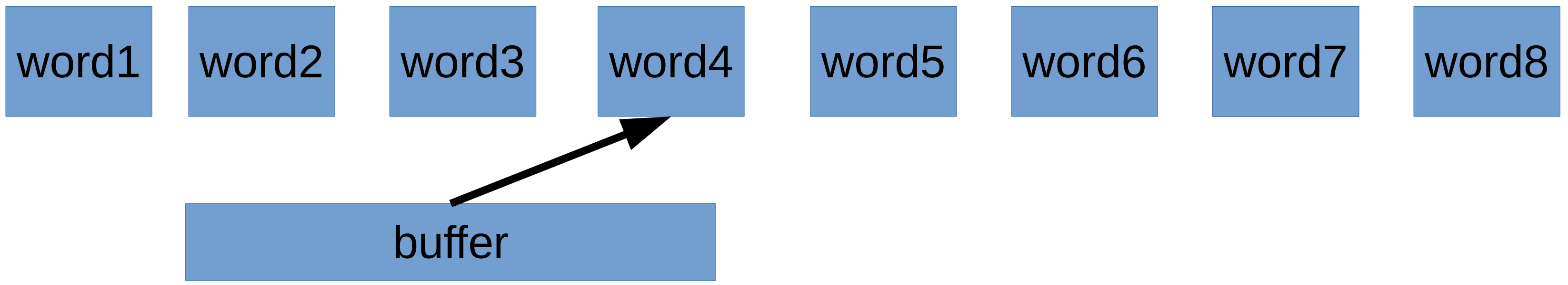
Build data set of batch and label



word1 word2 word3 word4 word5 word6 word7 word8

buffer

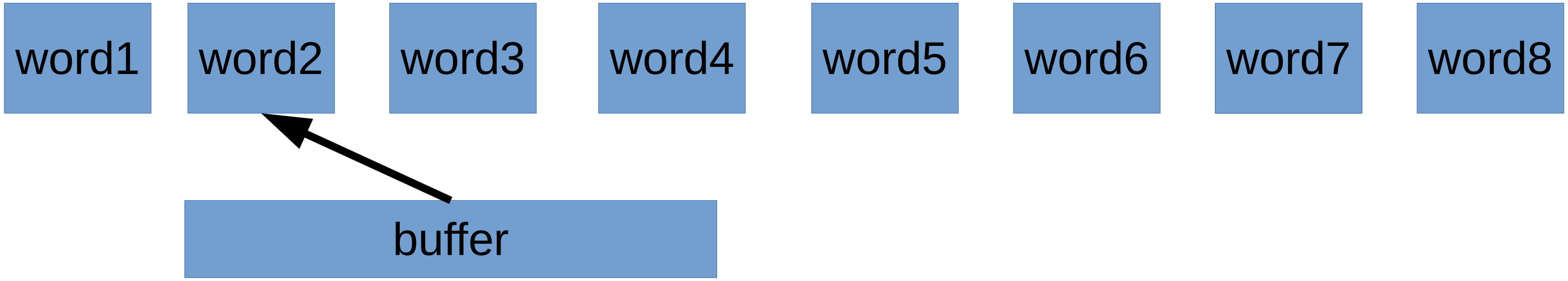
Build data set of batch and label



batch[i+3]=index(word3)

label[i+3]=index(word4)

Build data set of batch and label



word1 word2 word3 word4 word5 word6 word7 word8

buffer

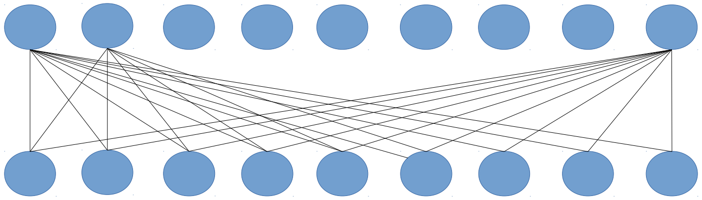
batch[i+3]=index(word3)
label[i+3]=index(word4)

batch[i+4]=index(word3)
label[i+4]=index(word2)

tf.nn.nce_loss function

tf.nn.nce_loss(weights, biases, inputs, labels, num_sampled, num_classes, num_true=1, sampled_values=None, remove_accidental_hits=False, partition_strategy='mod', name='nce_loss')

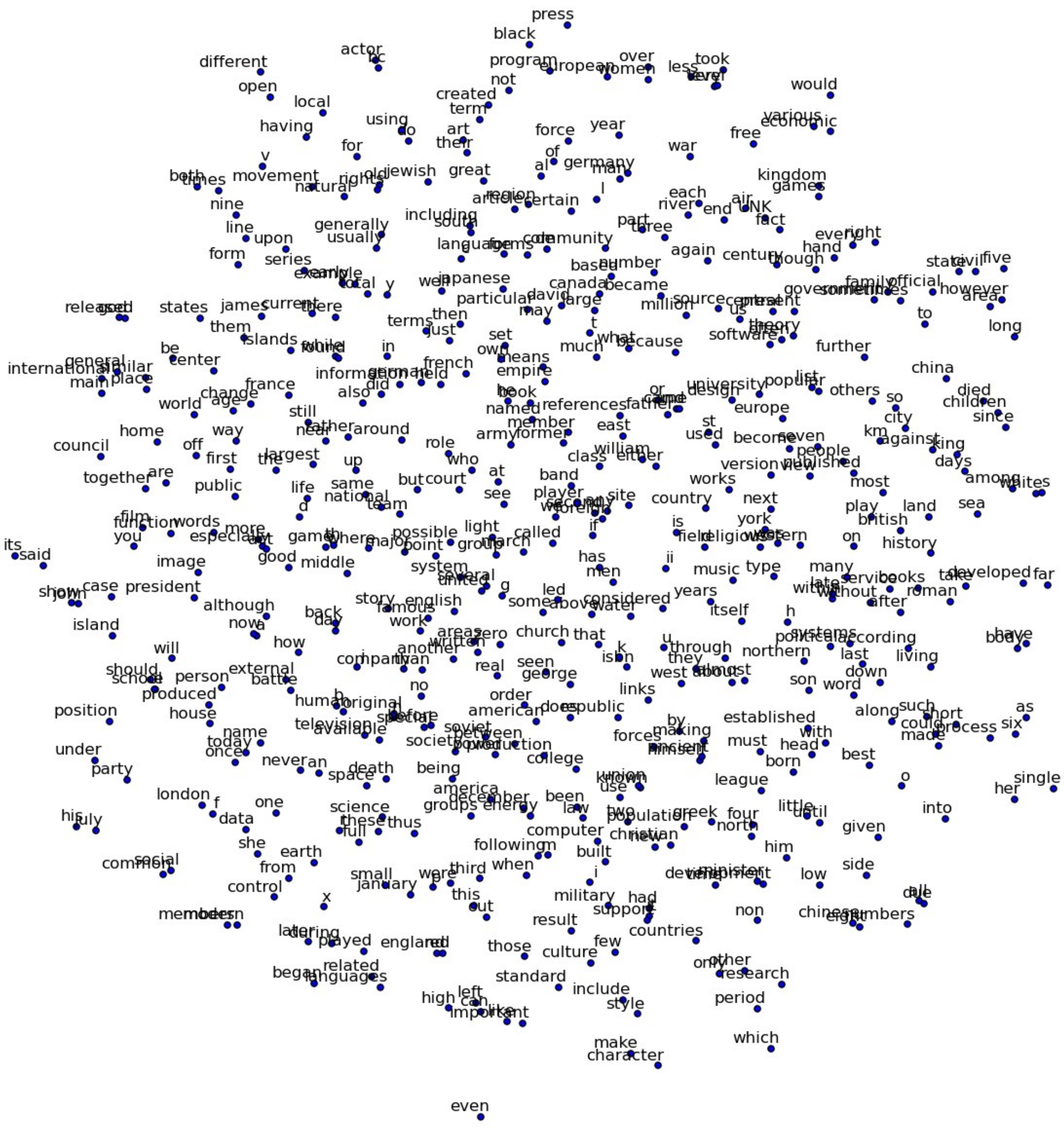
num_classes → Output vectors



Dim → Vector components

- Arg:
  - Weights: A Tensor, shape [num_classes, dim].
  - Biases: A Tensor, shape [num_classes].
  - Inputs: A Tensor, shape [batch_size,dim].
  - Labels: A Tensor, int64, shape[batch_size, num_true].
  - num_sampled: The number of noise words
  - num_classes: The number of possible classes.
  - num_true
  - …

Embeddings in two dimensions: t-SNE(wiki)



Discussion

- Watching the two vectors during the learning whether they become close or away.
- Input feature vector into CNNs to text classification: http://www.wildml.com/2015/12/implementing-a-cnn -for-text-classification-in-tensorflow/

References

- Mikolov et. al., Distributed Representations of Words and Phrases and Their Compositionality, 2013.
- Mikolov et. al., Efficient estimation of word representations in vector space, 2013.
- Mnih et. al., Learning word embeddings efficiently with noise-contrastive estimation, 2013.