AIML428

- Project baseline due in week 4
- Typical text classification
  - Text representation: Word embedding
  - Classification algorithm: CNN

- Today
  - Introduction to word embedding
    - Motivation
    - Word2Vec: Two models

Review

- Text classification
  - Text representation
    - Bag-of-words model
      - Each unique word is a feature:
      - Each document is a vector
      - Term weight:
        - count
        - TFIDF:
  - Classification algorithms
    - K Nearest Neighbour
    - Naïve Base
    - Support Vector Machine
  - One classical model for many traditional algorithms

	a	b	c	d
aacab	1	1	1	0
bcdaa	1	1	1	1

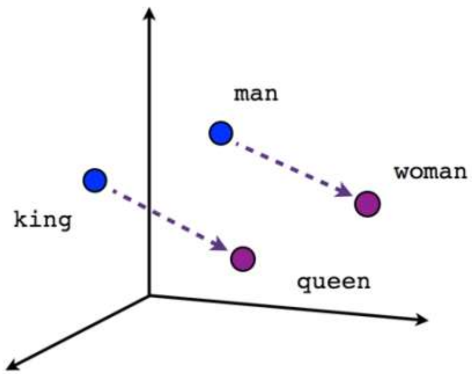
aacab	3	1	1	0
bcdaa	2	1	1	1

aacab	3	1	1	0
bcdaa	2	1	1	1.41

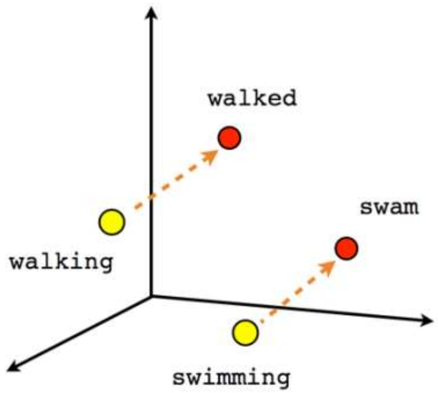
The distance between any two words

- Previously
  - Two words are either the same: 0
  - Two words are not the same: indefinite
- But some words are semantically related
  - good and excellent, bad and terrible
  - day and night, good and bad
- Key question: how to decode the meaning of a word
  - Cat
  - The cat (Felis catus) is a domestic species of small carnivorous mammal.

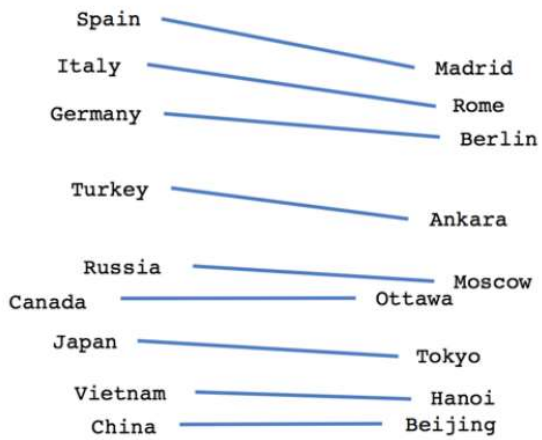
Linguistic Regularities



Male-Female

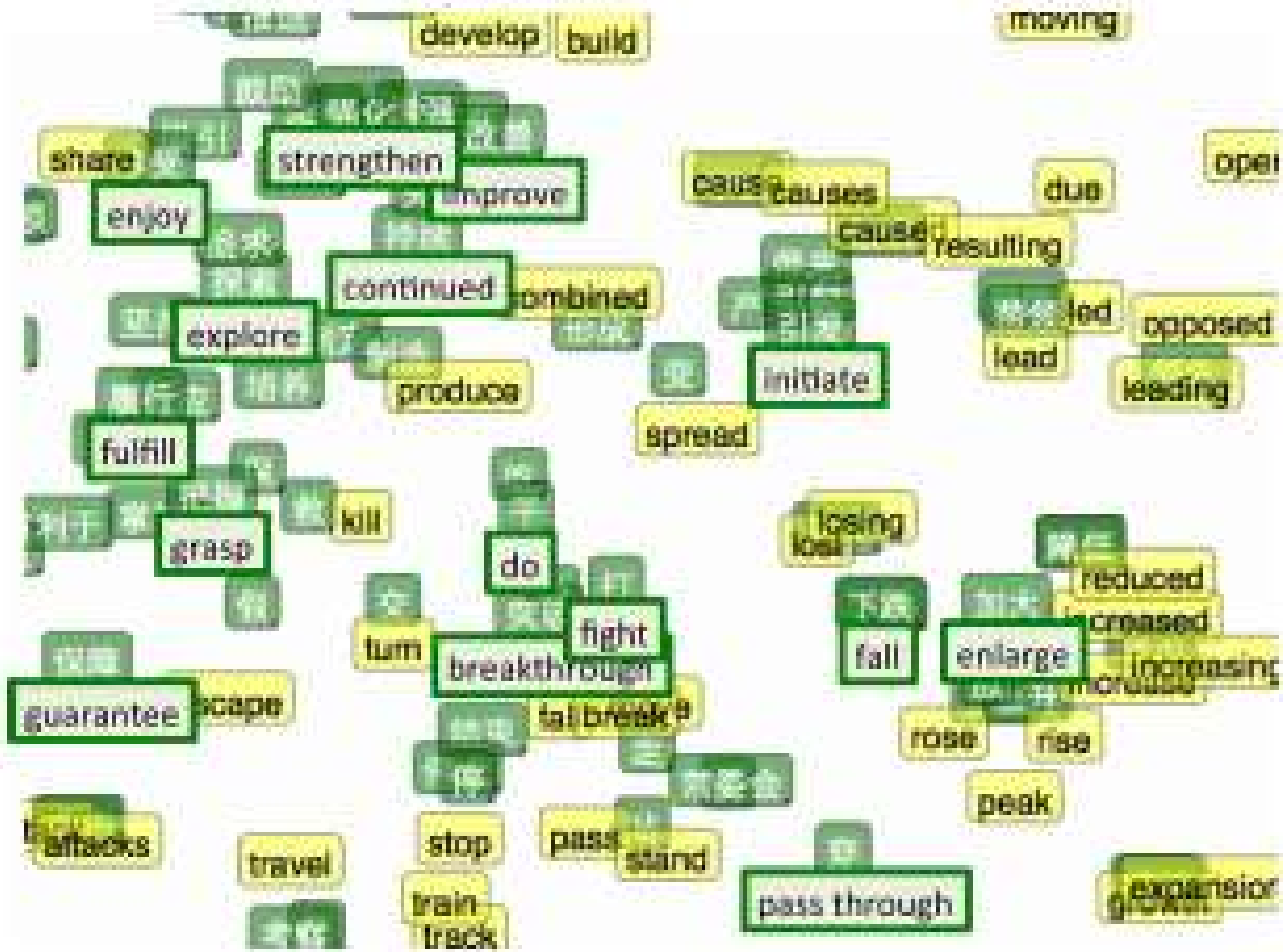


Verb tense



Country-Capital

Vector space: Language independent



Represent each word as a vector

- Cat= [0.83, 0.52, -1.63, 0.07, -0.36, ... -1.2, 0.02]

- So we can use cosine similarity to measure the distance
- We can even do math on it
  - king + women - man = queen

- Questions:
  - What are the dimensions
  - How many dimensions
  - How to get the value for each dimension

Word Embeddings

- Word Vectors
- Word Embeddings
- Vector-space word representations
- Continuous space word representations models
- …

A word embedding is a form of representing words using a dense vector representation.

[0.83, 0.52, -1.63, 0.07, -0.36, ... -1.2, 0.02]

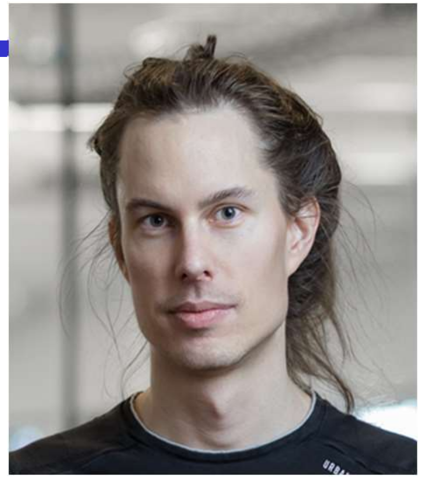
=

cat

- Examples
  - wiki-news-300d-1M.vec
  - globe.6B.50d
- Word2Vec, Glove, FastText

Word2Vec

Tomas Mikolov



Jun 2013 — Sep 2013 — Sep 2013 — Oct 2013 — May 2014 — Nov 2015 — Jul 2016

Vector offset | Word2Vec | Machine Translation | Phrases++ | Doc2Vec | Sense2Vec | FastText

Mikolov, T., Yih, W. T., & Zweig, G. (2013, June). Linguistic regularities in continuous space word representations. In *hlt-Naacl* (Vol. 13, pp. 746-751).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Distribution Hypothesis

“You shall know a word by the company it keeps”

John Rupert Firth

Consider the Context: (phrase minus word)

The _____ hurt its paw.

What would make sense here?

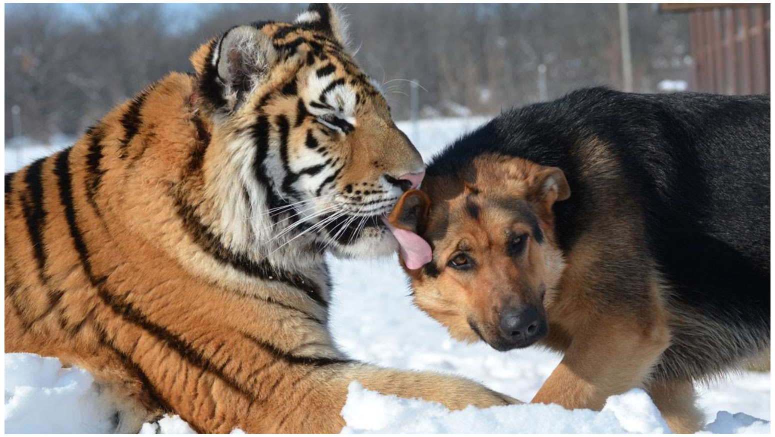
Cat, Dog, or Siberian_Tiger? YES

X-Wing, Lollygag? NO

What does this mean?

It means that Dog and Tiger

are similar to each other:



And not to ‘X-Wing’

How is Distribution Hypothesis relevant?

It means that:

If you know how well any two words fit all contexts, then you know how similar they are in meaning.

Therefore:

If you train a model to predict the likelihood of a word appearing in a context, then you are training it to find the meaning of the word.

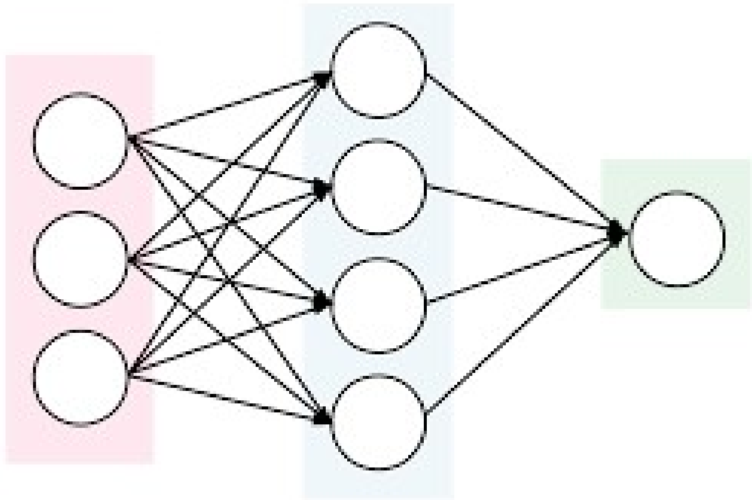
<u>**This is exactly what word2vec does!**</u>

Conceptualise Word2Vec

Given what we have learned, we need to:

- Define how the model predicts the likelihood of a word in a context.

- Cover how the word vectors are trained to maximise predictive accuracy.



Two approaches

There are two ways to train the vectors:

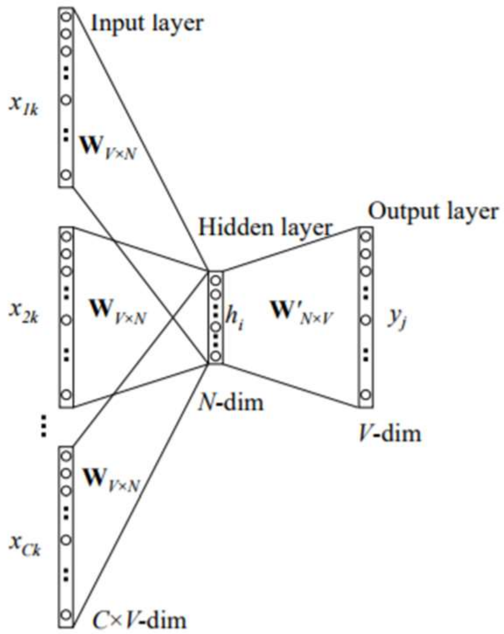
CBOW model (Continuous Bag Of Words)

Input is the context, output is the word

Skip Gram Model

Input: the word, output: the context

Word2Vec Model 1: CBOW



The cat sat on the mat

CBOW (continuous bag of words)
The weights near the output layer are extracted as vector values

Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Visualisation

- https://ronxin.github.io/wevi/

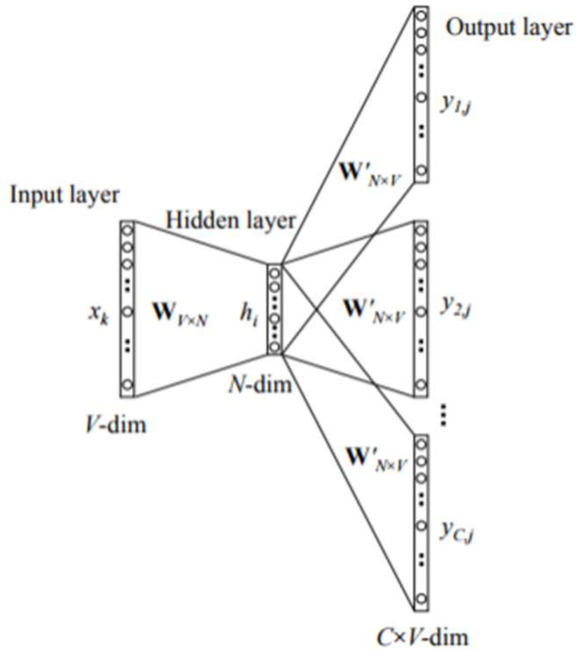
- CBOW
  - Input: two words as context
  - Output: one word as the word

- Skip gram

Word2Vec Model 2: Skipgram

Input: the word,
output: the context

The cat sat on the mat

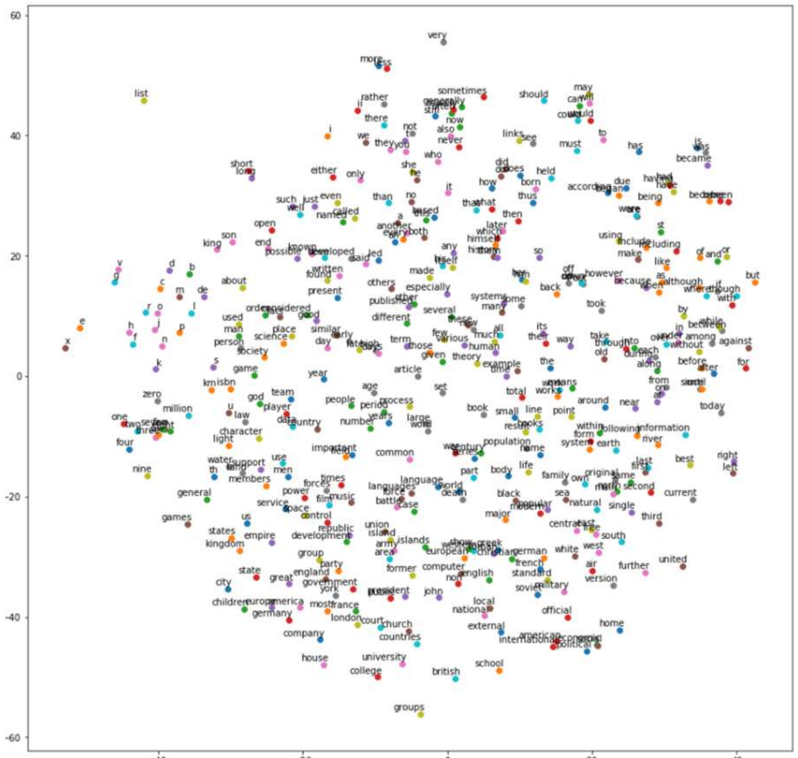


Skipgram

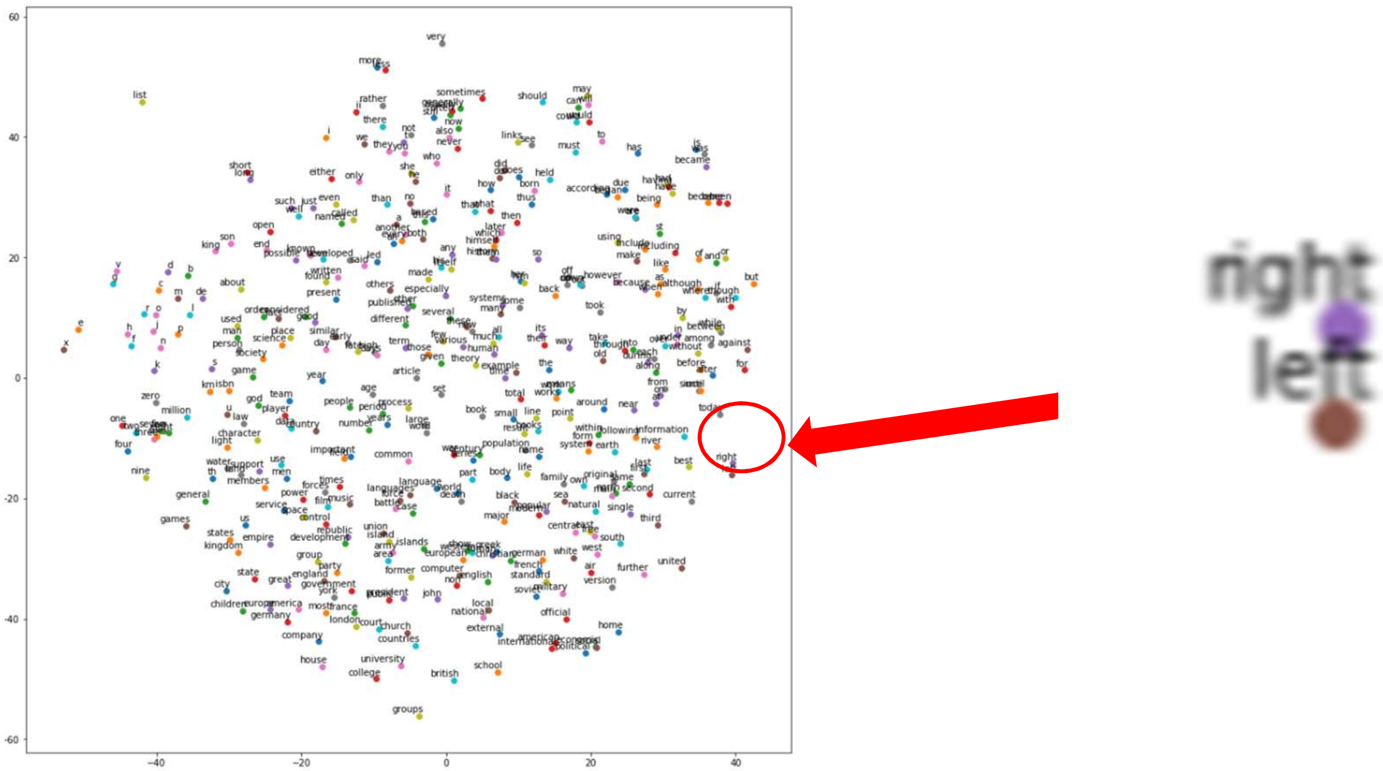
The weights at the input layer side are extracted as vector values

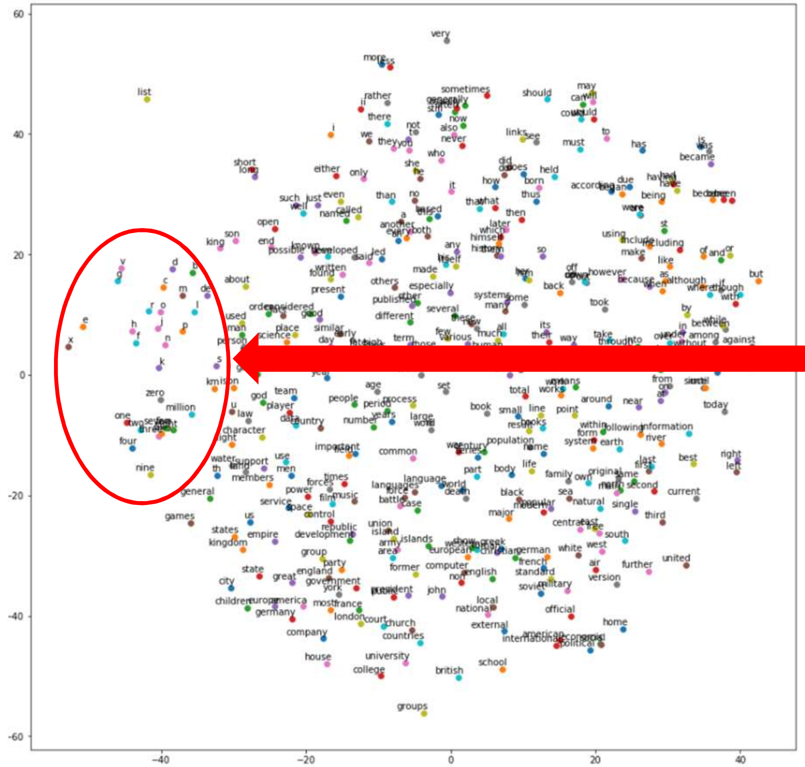
Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

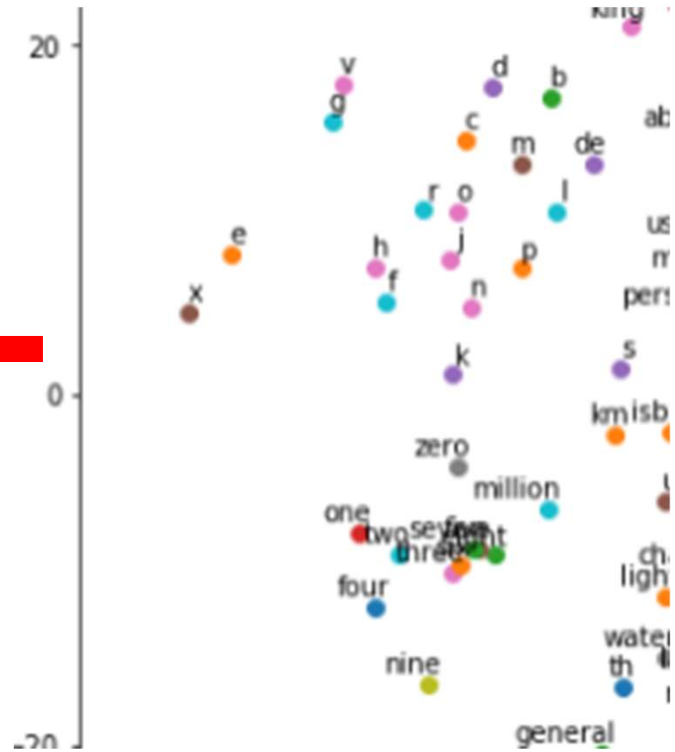
PCA of training on text

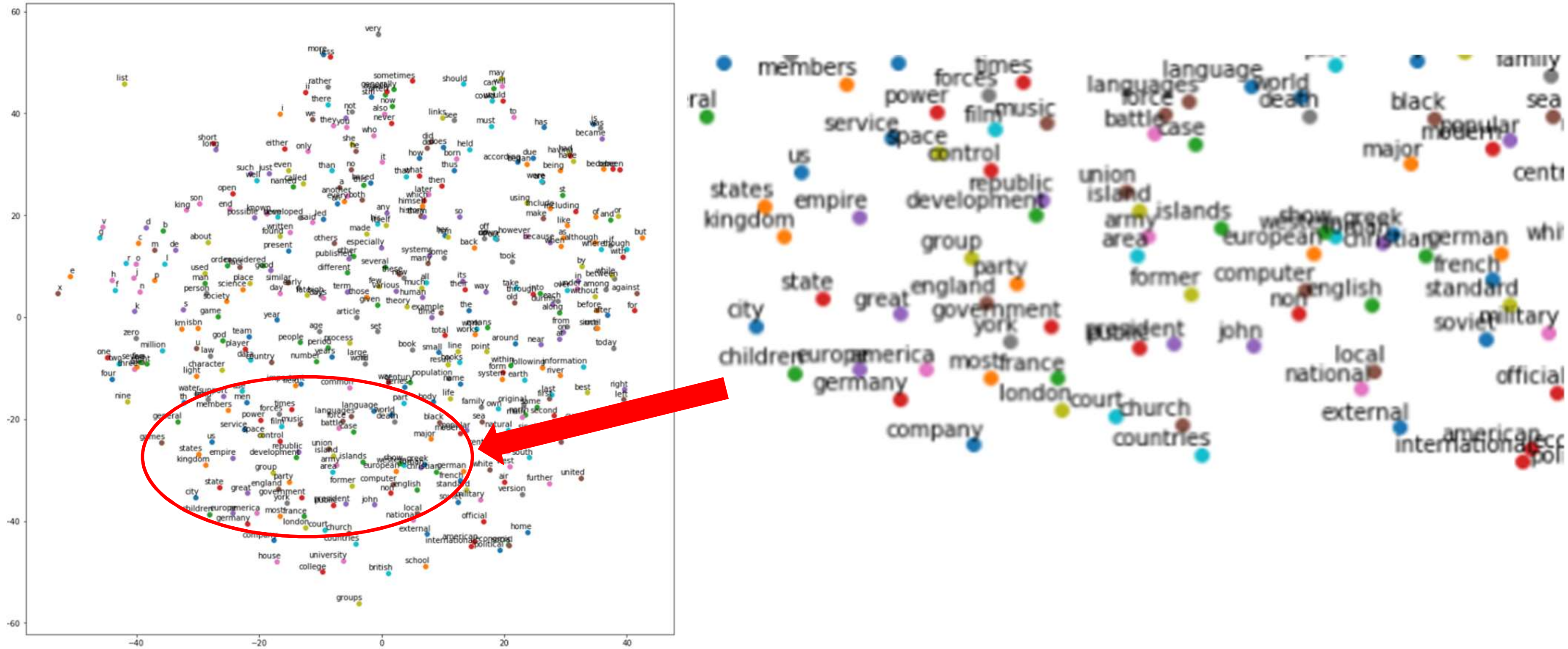


Visualisation of a pre-trained embedding






20
0
v
d
b
g
c
m
de
r
o
l
e
h
j
p
x
f
n
k
s
zero
million
one
four
nine
general



60
40
20
0
-20
-40
-60
-40
-20
0
20
40
members
power
service
space
control
us
states
kingdom
empire
development
republic
union
island
islands
army
area
european
group
party
england
state
great
government
york
city
children
america
germany
most
france
london
court
church
countries
company
former
computer
english
john
local
national
external
official
international
american
black
sea
major
german
french
standard
military
soviet
languages
language
force
battle
case
music
film
times
forces

Sources

An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec

- https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/

A visualisation at https://ronxin.github.io/wevi/

A tutorial on Word2Vec as implemented in Tensorflow:
https://www.tensorflow.org/tutorials/word2vec

(Contains the link to the original paper by Mikolov and the Google team)