AIML428

- ONLINE ONLY this week
- Check presentation sign ups
- New section Mon April 4th
- Submit the excel file for “peer review” after each presentation lectures.

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

Simple classifier for book reviews

Short version with simple classifiers is attached at lecture page.

- Load data
- Split data into train, test
- Prepare the data:
  - X:
    - CountVectorizer
    - TfidfVectorizer
  - Y: LabelEncoder
- Create a learning model:
  - KNeighborsClassifier or naïve_bayes or LogisticRegression
  - fit
  - predict
  - accuracy_score

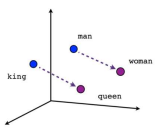
Bag-of-words model

What are the limitations or disadvantages?

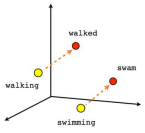
The distance between any two words

- 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.

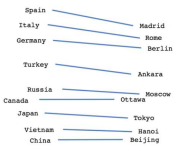
Linguistic Regularities



Male-Female



Verb tense



Country-Capital

58

Vector space: Language independent

59

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

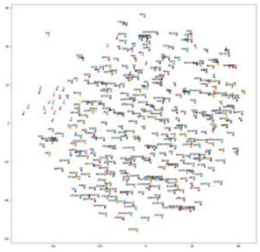
Tomas Mikolov

Jun 2013 — Vector offset
Sep 2013 — Word2Vec
Sep 2013 — Machine Translation
Oct 2013 — Phrases++
May 2014 — Doc2Vec
Nov 2015 — Sense2Vec
Jul 2016 — FastText

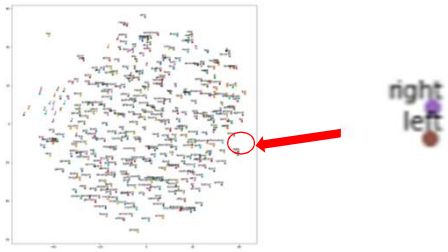
Mikolov, T., Yih, W. T., & Zweig, G. (2013, June). Linguistic regularities in continuous space word representations. In *hlt-Naacl* (Vol. 13, pp. 746-751).

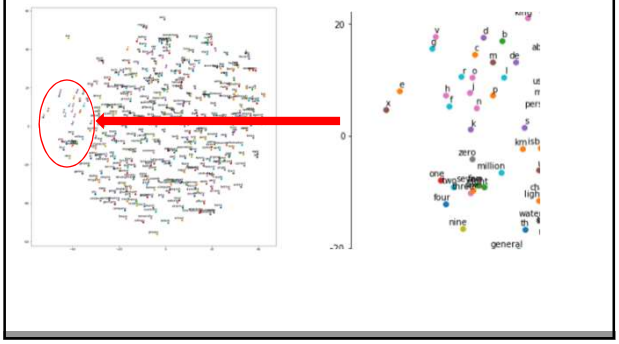
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

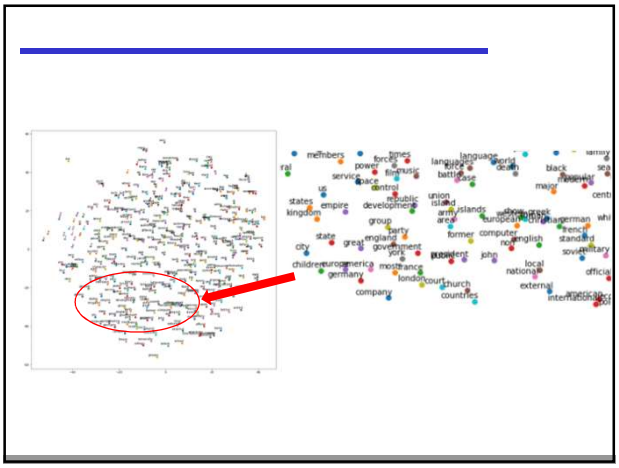
PCA of training on text (May move this to the end)



Visualisation of a pre-trained embedding







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:

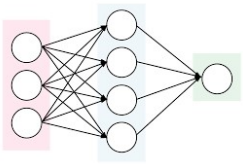
If you train a model to predict the likelihood of a word appearing in a context, then you are training it to find the meaning of the word.

This is exactly what word2vec does!

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:

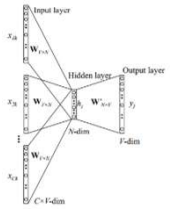
CBOW model (Continuous Bag Of Words)

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

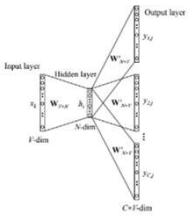
- https://ronxin.github.io/wevi/
- CBOW
  - Input: two words as context
  - Output: one word as the word
- Skip gram

74

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.*

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)