Australian National University

COMP4650/6490 Document Analysis

Representation in NLP

ANU School of Computing

Administrative matters

- Assignment 2
  - Release: Monday 21 August
  - Due: 5pm Wednesday 19 September
- Lab 3
  - Closely related to Assignment 2
  - Solutions will be released after the due date of Assignment 2

Outline

- Motivation
- Simple document representation
  - BoW model
  - Sparse representation
- Word representation
  - One-hot word representation
  - Context-based word representation
  - Co-occurrence & PPMI
  - Word2Vec

Outline

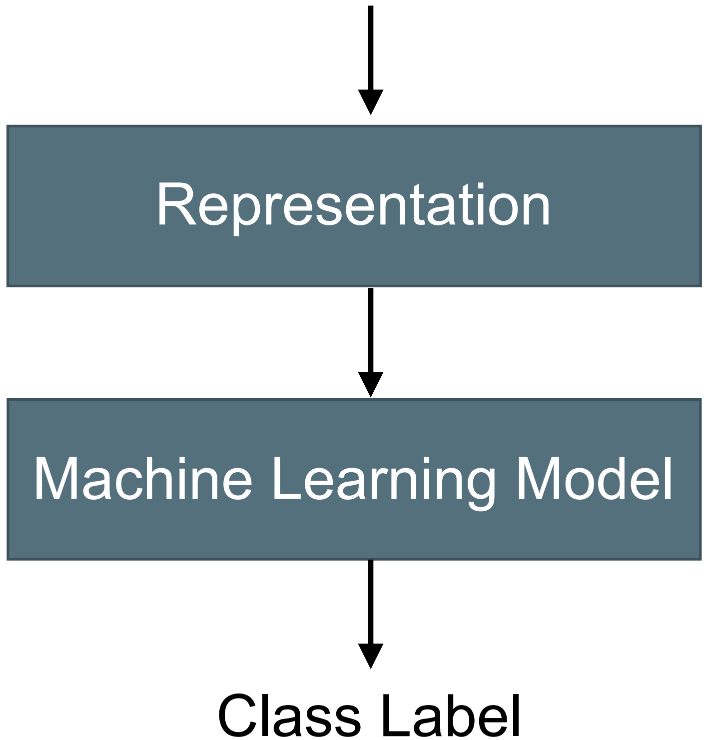
- Motivation
- Simple document representation
  - BoW model
  - Sparse representation
- Word representation
  - One-hot word representation
  - Context-based word representation
  - Co-occurrence & PPMI
  - Word2Vec

Motivation

Australian National University - Wikipedia

The Australian National University (ANU) is a national research university located in Canberra, the capital of Australia. Its main campus in Acton encompasses ...

Students: 20,892
Motto in English: "First to learn the nature of thi...
Motto: Naturam Primum Cognoscere Rerum
Administrative staff: 3,753



Motivation

Representing text

- Text represented as a string of bits/characters/ words is hard to work with
  - Variable length
  - High dimensional
  - Similar representations may have very different meanings
- How can we represent text a different way?

Motivation

Meaning

- Definition of meaning
  - What is meant by a word, text, concept, or action (a useless recursive definition from the dictionary)
- Meaning in language is
  - Relational (based on relationships)
  - Compositional (built from smaller components)
  - Distributional (related to usage context)

Australian National University

Outline

- Motivation
- Simple document representation
  - BoW model
  - Sparse representation
- Word representation
  - One-hot word representation
  - Context-based word representation
  - Co-occurrence & PPMI
  - Word2Vec

Vector space model in IR

Query

Where is the Australian National University

…	research	Australian	National	where
…	0	1	1	1

Document

Australian National University - Wikipedia

The Australian National University (ANU) is a national research university located in Canberra, the capital of Australia. Its main campus in Acton encompasses ...

Students: 20,892
Motto: Naturam Primum Cognoscere Rerum
Motto in English: "First to learn the nature of thi...
Administrative staff: 3,753

…	research	Australian	National	where
…	1	2	2	0

$$\textbf{score}(V_{\text{Query}}, V_{\text{Document}})$$

BoW model

- We can represent documents as vectors of the words / terms they contain (BoW model)
- Vector may be
  - Binary occurrence
  - Word count
  - TF-IDF scores
  - …

Australian National University - Wikipedia

The Australian National University (ANU) is a national research university located in Canberra, the capital of Australia. Its main campus in Acton encompasses ...

Students: 20,892 **Motto**: Naturam Primum Cognoscere Rerum

Motto in English: "First to learn the nature of thi... **Administrative staff**: 3,753

…	research	Australian	National	where
…	1	2	2	0

Sparse representation

- The vector representation may be the size of the vocabulary (e.g. 50k words is common)
  - Very inefficient for many documents
- Sparse representation
  - Most documents do not contain most words
  - We can use a sparse representation with tuples of the form: (term_id, term_count) for tuples where term_count > 0

Index	0	1	2	3	4	…
Term	research	Australian	National	where	Canberra	…
Count	1	2	2	0	1	…

  - The above document: (0, 1), (1, 2), (2, 2), (4, 1) …

Sparse representation

- We can then use our high dimensional sparse vector for document classification and other tasks

- You typically do not have to roll your own sparse vector representation, there are many libraries
  - *scikit-learn* returns a *scipy* sparse matrix when you call one of the vectorisers such as *CountVectorizer*
  - The *LogisticRegression* class in *scikit-learn* will accept sparse matrices as input (it will also accept dense matrices)

Outline

- Motivation
- Simple document representation
  - BoW model
  - Sparse representation
- Word representation
  - One-hot word representation
  - Context-based word representation
  - Co-occurrence & PPMI
  - Word2Vec

One-hot word representation

- In traditional NLP, we regard words as discrete symbols, e.g.
  - `hotel = 10 = [0 0 0 0 0 0 0 0 0 1 0 0 0 0]`
  - `motel =  7 = [0 0 0 0 0 0 0 1 0 0 0 0 0 0]`
- Vector dimension = number of words in vocabulary (e.g. 50k)
- Similar words do not have similar vectors
  - We need a new way of representing words

Context-based word representation

"You shall know a word by the company it keeps!"

J. R. Firth (1957). A synopsis of linguistic theory, 1930-1955.

Context-based word representation

I am a **student** at the Australian National **University**.
Research school of Computer Science is part of our **university**.
Our **university** is an **organisation** for **education**.
…

The context words will represent '**university**'!

Context-based word representation

- Context window:
  Use k words on *either side* of the *focal word* as the context, for example ($k = 2$):

In Australia a large university will often have a big Campus.

The university employs academics and professional staff.

Context-based word representation

Student

Knowledge

Research

University

Organisation

School

Education

Word co-occurrence matrix

- A co-occurrence matrix of words gives a vector representation for each word
  - This gives equal importance to all context words

context word

focal word

	drive	run	fluffy	broken
car	18	3	5	22
bus	20	2	1	25
cat	1	32	35	1
dog	3	41	38	1

Word co-occurrence matrix: Weighting

- We want to weight each of the terms in the word vector
- We can use TF-IDF (refer to IR lectures), to compute document frequency
  - Use documents to compute DF, or
  - Consider each context as a document
- A more common method for co-occurrence matrices is to weight by *Positive Pointwise Mutual Information* (PPMI)

Australian National University

Pointwise Mutual Information

- The amount of information the occurrence of a word w_i gives us about the occurrence of another word w_j (and vice versa)
- The ratio of the probability that the words occur together compared to the probability that they would occur together by chance

Probability that w_i, w_j occur together

$$\text{PMI}_{i,j} = \log \frac{P(w_{i,j})}{P(w_i)\,P(w_j)}$$

Probability that w_i occurs ignoring context

Probability that w_j occurs ignoring context

Pointwise Mutual Information

	drive	run	fluffy	broken
car	17	3	5	18
bus	20	2	2	16
cat	1	40	35	4
dog	2	35	38	2

$$P(w_i) = \frac{\#w_i}{\sum_i \#w_i}$$

$$P(w_{i,j} = \frac{\#w_{i,j}}{\sum_i \sum_j \#w_{i,j}}$$

$$\text{PMI}_{i,j} = \log \frac{P(w_{i,j})}{P(w_i)\, P(w_j)}$$

$$P(\text{cat}) = \frac{80}{240} = \frac{1}{3}$$

$$P(\text{run}) = \frac{80}{240} = \frac{1}{3}$$

$$P(\text{cat, run}) = \frac{40}{240} = \frac{1}{6}$$

$$\text{PMI}_{\text{cat, run}} = \log \frac{\frac{1}{6}}{\frac{1}{3}\,\frac{1}{3}} = \log \frac{3}{2}$$

Positive Pointwise Mutual Information

- PMI can be negative when words occur together less frequently than by chance
- Typically, negative values are not reliable.
- We set them to 0: $\text{PPMI} = \max(0, \text{PMI})$
- Similarity with PPMI

	drive	run	fluffy	broken
car				
bus				
cat		$40 \log(3/2)$		
dog				

$\mathbf{v}_{\text{cat}} = \text{PPMI}_{\text{cat},:}$ $\mathbf{v}_{\text{dog}} = \text{PPMI}_{\text{dog},:}$

$\text{sim(cat, dog)} = \cos(\mathbf{v}_{\text{cat}}, \mathbf{v}_{\text{dog}})$

Drawbacks of a sparse representation

- The raw count / TF-IDF / PPMI matrix suffers from sparsity, for example, if
  - "*puppy*" occurs frequently with "*training*", "*food*", "*woofing*"
  - "*dog*" occurs frequently with "*working*", "*eating*", "*barking*"
  - We want to say "*puppy*" is similar to "*dog*" because the words that occur with each are similar (even if the words they occur with are not the same)
- High dimensional, thus more difficult to use in practice

Word representation: Word vectors

$$\text{University} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Note: dense word vectors are sometimes called word embeddings or word representations. They are distributed representations produced by matrix factorisation (e.g. LSA) or learning to perform synthetic classification tasks (e.g. word2vec).

Typical number of dimensions: 64, 128, 256, 300, 512, 1024

Word2Vec

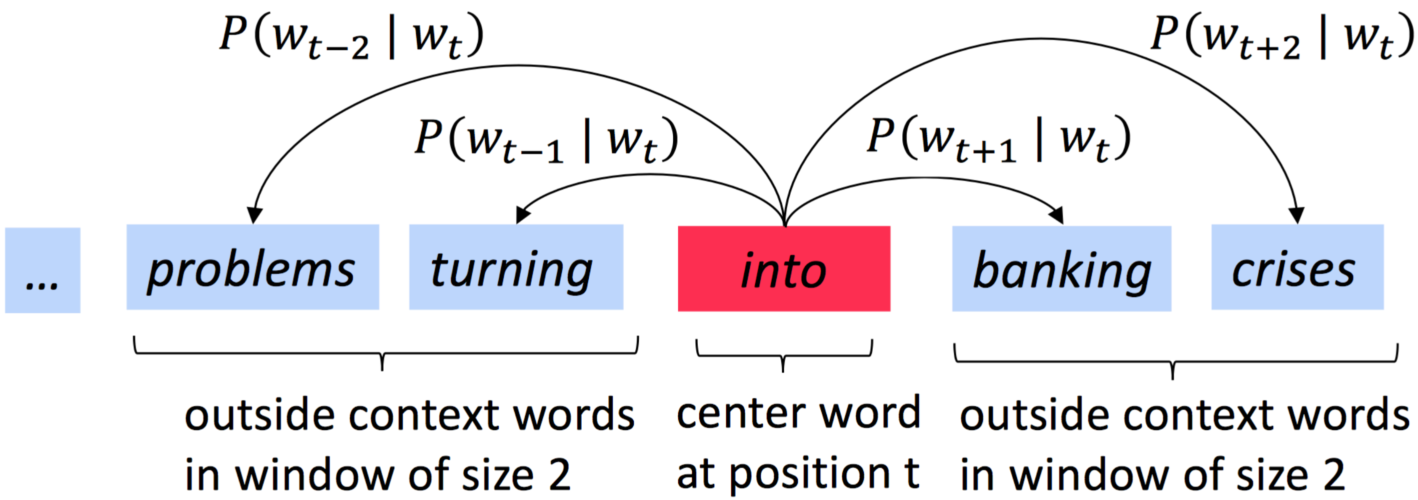
- Several word2vec Algorithms
  - Skip-gram with negative sampling
  - Continuous bag-of-words (CBoW)
- Approach:
  - A self-supervised classification problem
  - We learn word embeddings to do classification
  - In the end we care only about the embeddings

Also see this blog post for a good explanation of word2vec: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

A good source for word2vec loss function derivation: https://cs224d.stanford.edu/lecture_notes/notes1.pdf

Word2Vec

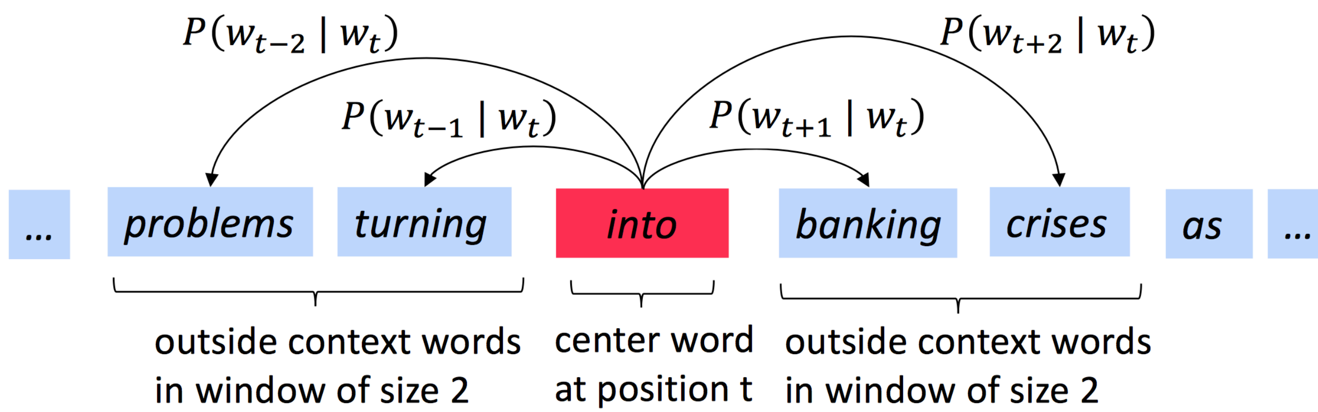
- Skip-gram: Given centre word w_t, predict context words in window $j \in [-m, m]$, by $P(w_{t+j} \mid w_t)$



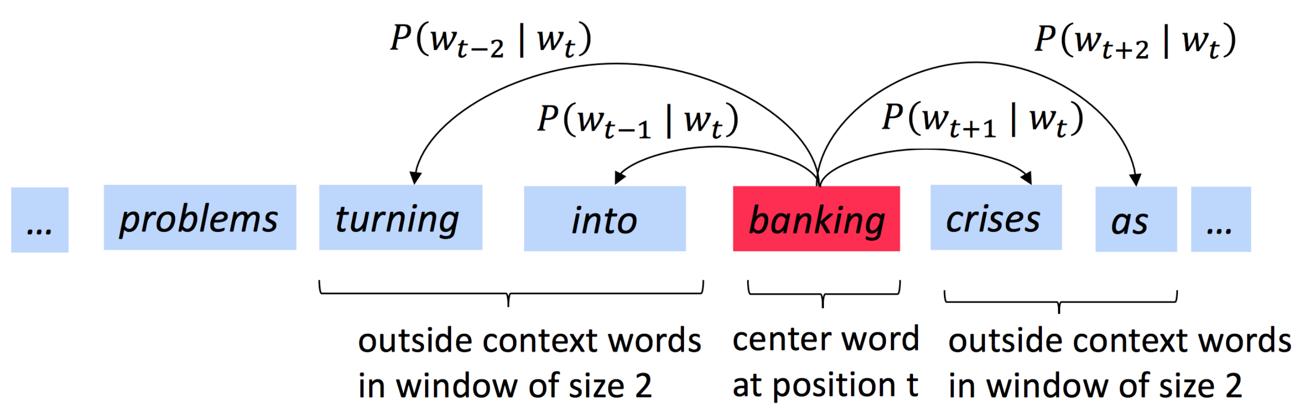
- CBoW: Predict the (next) centre word given (previous) context words
- Both can be extended to sentence or document vectors (i.e. doc2vec)

Word2Vec: Skip-gram

- One step:



- Next step:



Word2Vec: Training data

- Skip-gram training data
  - Context window: $k = 2$
  - Sentence:
    … problems turning into banking crisis as …

problem *turning* *into* *banking* *crisis* *as*

Input	Output	
Word	**Context**	**Group**
…	…	
into	problems	3
into	turning	3
into	banking	3
into	crisis	3
banking	turning	4
banking	into	4
banking	crisis	4
banking	as	4
…	…	4

Need to record context group to calculate loss

Word2Vec as logistic regression

- Word2Vec uses a multinomial logistic regression classifier (without a bias term)

$$P(\mathbf{y} \mid \mathbf{x}) = \text{softmax}(W\mathbf{x})$$

  - Here $\mathbf{y}$ is the context word
  - $\mathbf{x}$ is an embedding of the centre word
  - Matrix W can be interpreted as being composed of embeddings of the context words

Word2Vec: Model

- How to calculate $P(w_{t+j} \mid w_t; \boldsymbol{\theta})$?
  - $\mathbf{v}_o$: Context (observed) word vector
  - $\mathbf{v}_c$: Centre word vector

$$\left.\begin{array}{l}\mathbf{v}_o \\ \mathbf{v}_c\end{array}\right\} \boldsymbol{\theta}$$

$$P(o \mid c; \boldsymbol{\theta}) = \frac{\exp(\mathbf{v}_o^\top \mathbf{v}_c)}{\sum_{w \in V} \exp(\mathbf{v}_w^\top \mathbf{v}_c)}$$

- Once trained with cross-entropy loss
  - Two matrices of trained parameters: W for context words, X for centre words
  - We may use $\mathbf{v}_c$ (from X) as the word embedding and throw everything else away
  - Or use the sum of centre word embedding (from X) and context word embedding (from W) of the same word

Word2Vec: Training

- When we train word2vec we also train the centre word embeddings (i.e. X)
- How:
  - Start with random embeddings
  - Back-propagation to compute gradient (next week)
  - (Stochastic) gradient descent

Word2Vec: Objective

- Loss function (average cross entropy) given a sequence of training words $w_1, w_2, w_3, \ldots, w_T$

$$J(\boldsymbol{\theta}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m \le j \le m,\, j \ne 0} \log P(w_{t+j} \mid w_t; \boldsymbol{\theta})$$

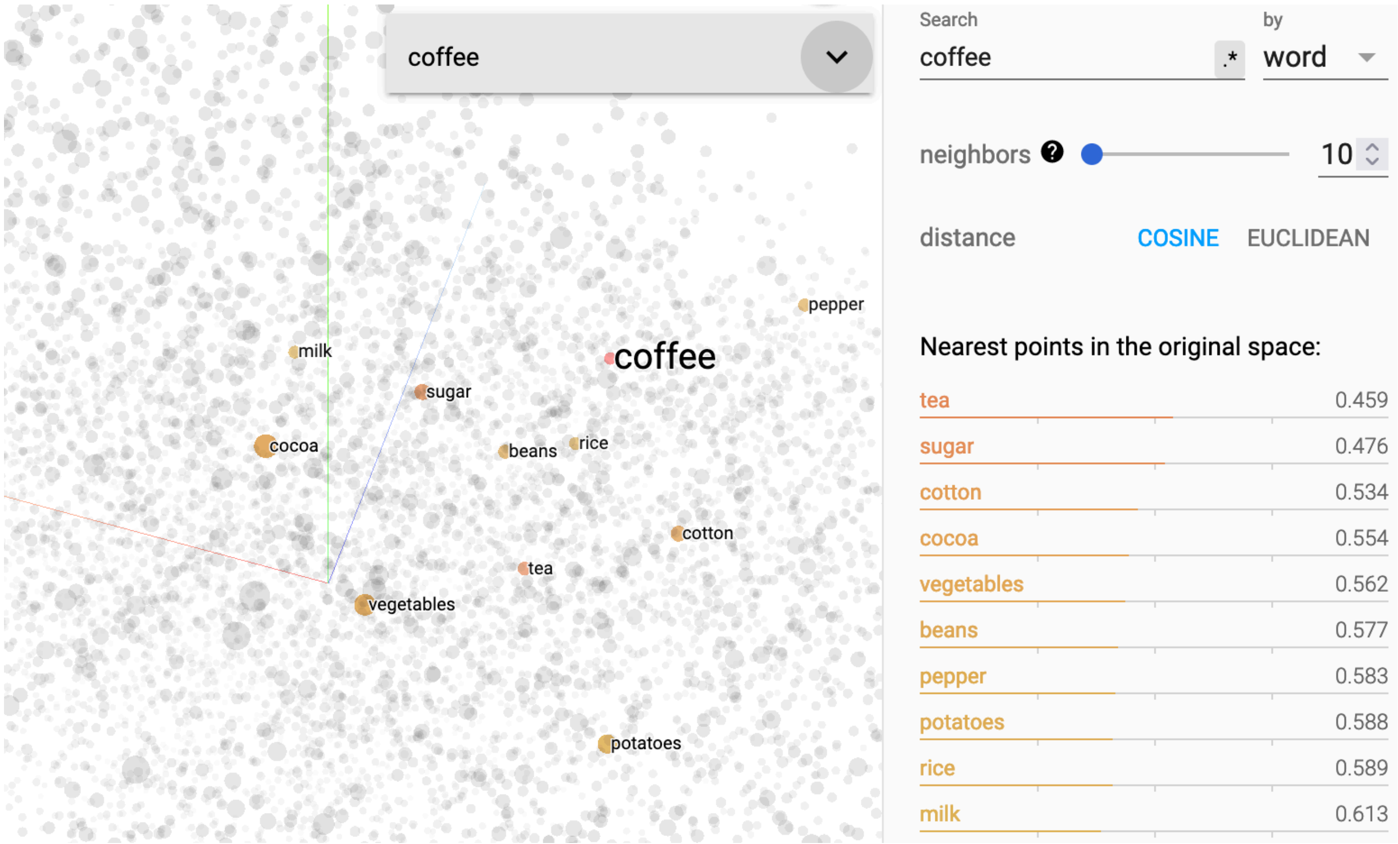
  - The normalising factor in $P(w_{t+j} \mid w_t; \boldsymbol{\theta})$ is often approximated by negative sampling
  - See the textbook if you want to know more details

Word2Vec: Practical considerations

- Context window size is very important
  - Word2Vec randomly samples different sized windows
  - This implicitly increases the weight of words that are close to the centre word
- Word2Vec also down-samples common words

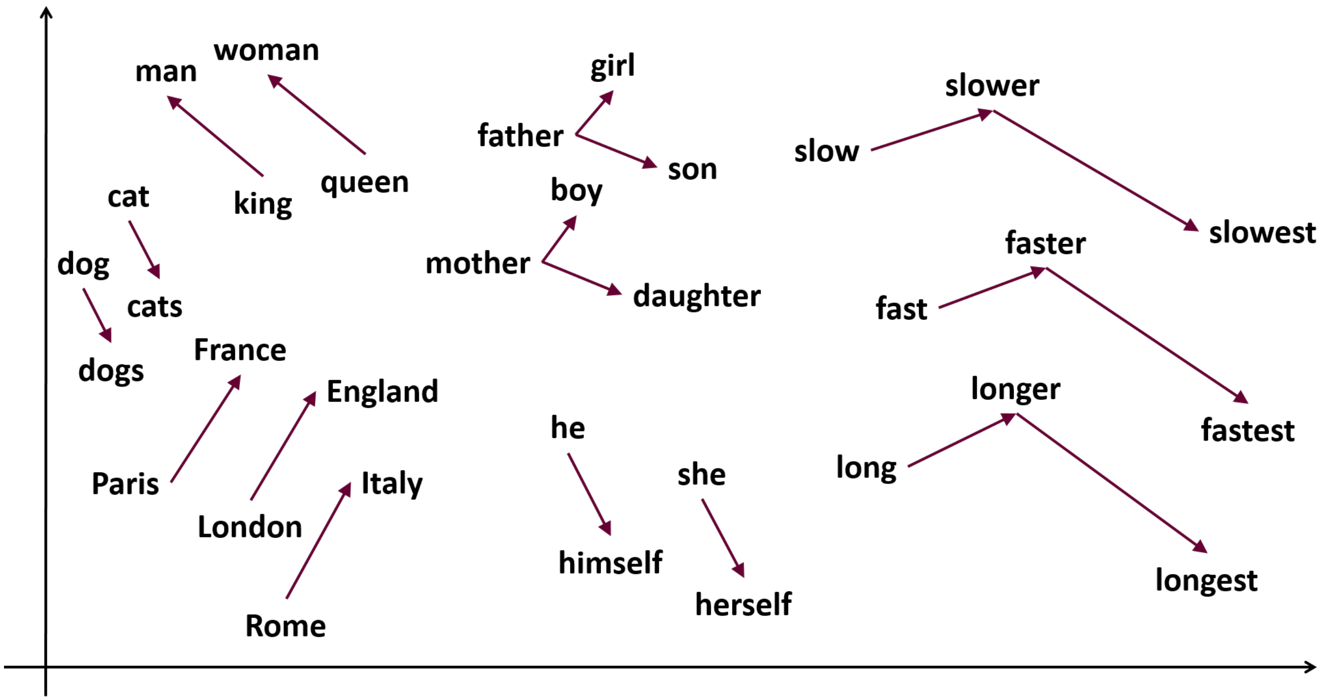
Word2Vec: Visualisation

- Low dimensional projection to 2D or 3D using PCA, T-SNE, UMAP



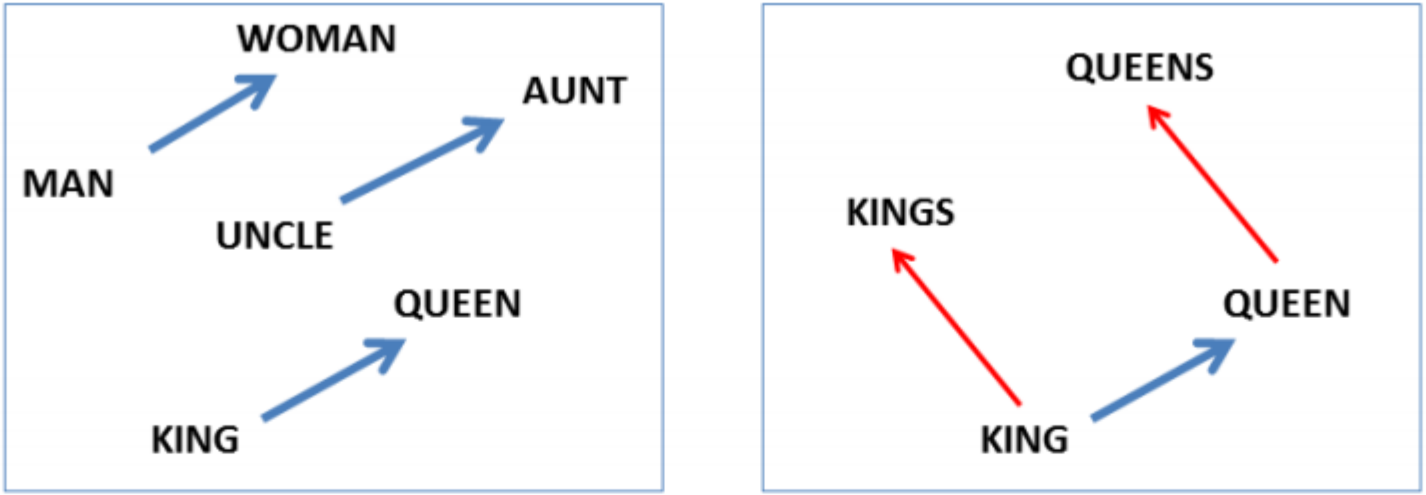
https://projector.tensorflow.org/

Word2Vec: Visualisation



From https://samyzaf.com/ML/nlp/nlp.html

Word2Vec: Visualisation



(Mikolov et al., NAACL HLT, 2013)

KING + WOMAN – MAN = QUEEN

Better document representations

- Meaning is compositional:
  - Go from word embeddings to document embeddings by combining word embeddings
- Methods for combining embeddings:
  - Simple Aggregation (sum, mean, max …)
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Transformers

Summary

- Motivation
- Simple document representation
  - BoW model
  - Sparse representation
- Word representation
  - One-hot word representation
  - Context-based word representation
  - Co-occurrence & PPMI
  - Word2Vec

References

- Chapter 6, Speech and Language Processing
- Word2Vec:
  Distributed Representations of Words and Phrases and their Compositionality
  https://arxiv.org/pdf/1310.4546.pdf
- Doc2Vec:
  Distributed Representations of Sentences and Documents
  https://proceedings.mlr.press/v32/le14.pdf