Lecture 7

WORD REPRESENTATION LEARNING

Contents

- Word Representation
  - Word Vectorization
  - Word Embedding
- Word2Vec
- Other models
  - GLoVE
  - FastText
  - ELMo
- Building a representation word model
- Application: Plagiarism Detection

Review - Text Document Vectorization Approaches

The meaning of a word

- Meaning:
    - the idea that is represented by a word, phrase, etc.
    - the idea that a person wants to express using words, phrases, etc.
    - the idea that is expressed in a work of writing, art, etc.
- Some commonest linguistic ways of thinking of meaning:

The picture can't be displayed.

Discrete Vector Representation

- Bag-of-words model
- Co-occurrence matrix
- TF-IDF
- One-hot encoding vector

Bag-of-words Model

The picture can't be displayed.

One-hot Vector

The picture can't be displayed.

TF-IDF

The picture can't be displayed.

Window-based co-occurrence matrix

- Window length = 1 (more common: 5-10)
- Symmetric (irrelevant whether left or right context)
- Example corpus:
  - I like deep learning
  - I like NLP
  - I enjoy flying

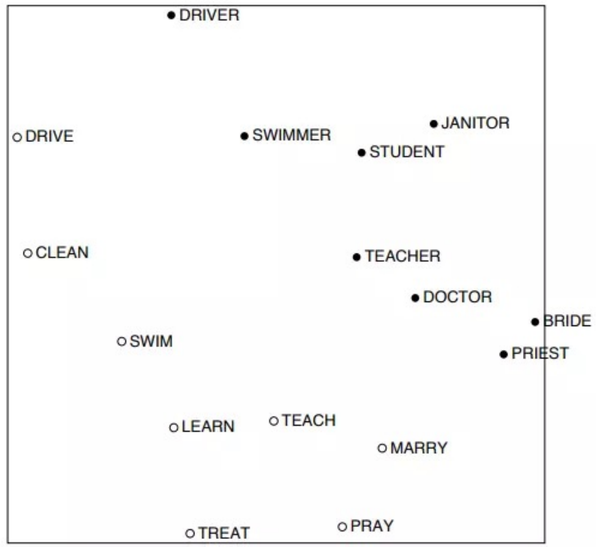
The picture can't be displayed.

Problems with words as discrete vectors

- Example: search for the keywords “Seattle Motel” will result also “Seattle Hotel”

The picture can't be displayed.

- However:
- These two vectors are orthogonal, without any notion of similarity for one-hot vectors
- Solution: **encode the similarity inside the word vector**



An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence
Rohde et al. 2005

Word Vector Representation

Representing word by their context

- Distributional semantic: ***A word’s meaning is given by the words that frequently appear close-by***
- When a word w appears in a text, its **context** is the set of words that appear nearby (within a fixed-size window)
- We will use the many contexts of w to build up a representation of w

The picture can't be displayed.

Word vectors

- Each word is represented by a dense vector which is chosen so that it's similar to vectors of words that appear in similar contexts
- The similarity is measured by the vector dot (scalar) product

The picture can't be displayed.

Word Embedding Model: word2vec

The main idea:

- Start with random vectors
- Iterate through each word position in the whole corpus
- Try to predict surrounding words using word vectors

The picture can't be displayed.

The picture can't be displayed.

- Learning: update vectors so they can predict actual surrounding words better
- The word embedding model try to learn word vectors that capture well word similarity and meaningful directions in a word space!

Word2vec maximizes objective function by putting similar words nearby in space

The picture can't be displayed.

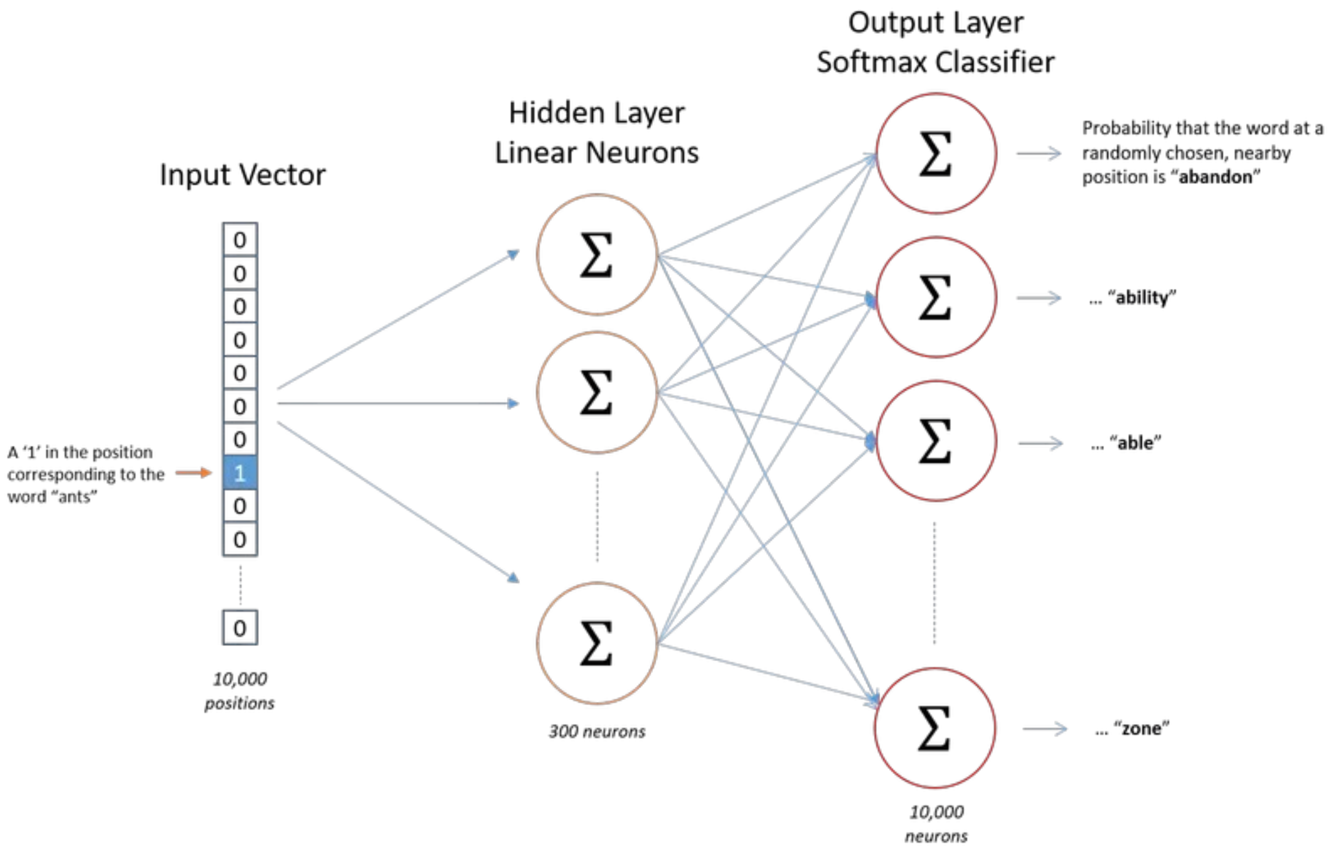
Two variant approaches for word2vec

The picture can't be displayed.

Skip-gram vs. CBOW

- Skip-gram
  - Predict context (‘outside’) words (position independent) given center word
- CBOW
  - Predict center word from (bag of) context words

Skip-gram



Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
0
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
10,000 positions
300 neurons
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"
10,000 neurons

Skip-gram

- Current word is used as input to a log-linear classififier
- Predict words within certain range before and after of this current word
- The normalization term is computationally expensive as:

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

A big sum over words

Skip-gram

Source Text	Training Samples
The quick brown fox jumps over the lazy dog.	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog.	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog.	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown fox jumps over the lazy dog.	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

Skip-gram

- Current word is used as input to a log-linear classififier
- Predict words within certain range before and after of this current word
- The normalization term is computationally expensive as:

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

A big sum over words

- Skip gram model is typically implemented with "**negative sampling**"

Skip-gram with negative sampling

- Main idea: train binary logistic regression to differentiate a true pair (center word and a word in its context window) versus some “noise” pairs (the center word paired with a random word)
- Maximize the objective function:

- We take k negative samples, maximize the probability that real outside word appears, minimize the probability that random words appear around center word

CBOW (Continuous BOW) (1)

- Predict a word using context

The picture can't be displayed.

The picture can't be displayed.

CBOW (2)

- |V| is the size of the vocabulary corpus
- x_i represents the one-hot vector of the i^{th} word
- y_i represents the one-hot vector of the expected word

The picture can't be displayed.

CBOW (3)

- |V| is the size of the vocabulary corpus
- x_i represents the one-hot vector of the i^{th} word
- y_i represents the one-hot vector of the expected word

The picture can't be displayed.

CBOW (4)

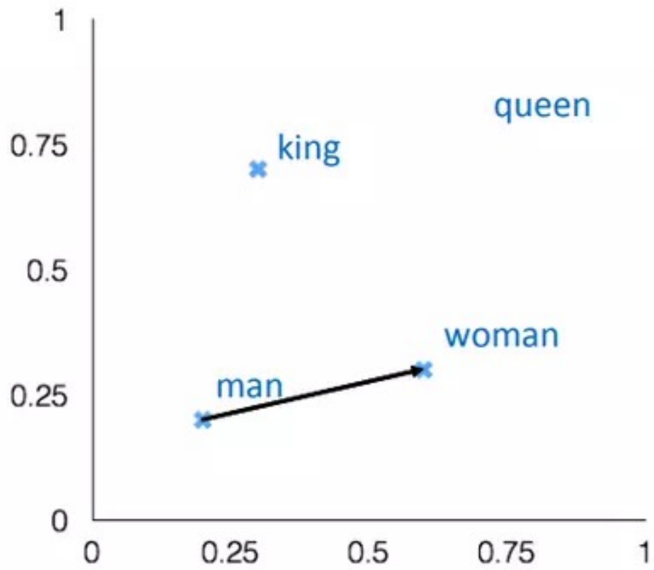
The picture can't be displayed.

is the one-hot vector of the expected word

Example from word2vec

man:woman :: king:?

+	king	[0.30 0.70]
-	man	[0.20 0.20]
+	woman	[0.60 0.30]
	queen	[0.70 0.80]



GLoVe – Global vector for word representation

- Idea: Encoding meaning components in vector differences

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

- Consider the co-occurrence probabilities for target words ice and steam with various probe words from the vocabulary
  - **ice** co-occurs more frequently with **solid** than it does with **gas**
  - **steam** co-occurs more frequently with **gas** than it does with **solid**
  - Both words co-occur with their shared property **water** frequently, and both co-occur with the unrelated word **fashion** infrequently

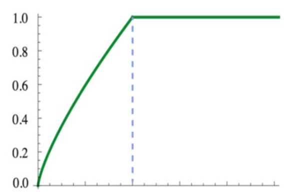
GLoVe – Global vector for word representation

- 2 assumptions:
  - Global context (global matrix factorization)
    - Co-occurrence of words through documents
  - Local context
    - A pre-fixed size slide window
- Idea: Encoding meaning components in vector differences

A: Log-bilinear model: $w_i \cdot w_j = \log P(i|j)$

with vector differences $w_x \cdot (w_a - w_b) = \log \frac{P(x|a)}{P(x|b)}$

Loss: $J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$

$f \sim$ 

- Fast training
- Scalable to huge corpora

Word2Vec vs. Glove

The picture can't be displayed.

Pros and cons

- Pros:
  - Fast training
  - Well scaled on big corpus data
  - Work well on small data
  - Early stopping
- Cons
  - Memory consuming
  - Learning rate may affect the accuracy of the model

Fast Text

- An extension of Word2vec
- Facebook
- Multi-language support
- Sub-word
  - Based on n-grams model
  - Allow to short word learning
  - Allow to represent prefixes and postfixes of word
- Work well with rare-words

Word vector of Fast text

Pros and cons of Fast Text

- Capture rare-words thanks to sub-word representation
- Memory consuming for sub-word representations

ELMo – Embedding from Language Model

The picture can't be displayed.

ELMo - Architecture

- Using character-level CNN to represent word (raw word vectors)
- Bi-directional LSTM
- Combine forward and backward directions to represent intermediate word vectors
- Second layer using intermediate word vectors as input with the same architecture as first layer
- Final word vector:
  - Combine first representation (raw word vector) and two outputs of two layers – ELMo vector

Pros and cons

- Better capture the context of word than word2vec and Glove

The picture can't be displayed.

- The same words may have the different representation depending on the context
- Rare-word representation (as Fast Text)
- Memory consuming

Building word vector model

- A large text corpus
  - Wikipedia
- Choose an appropriate model
- Training model
- Using the model to solve problems of NLP

Application - Plagiarism Detection

Problem of Plagiarism Detection

The picture can't be displayed.

Input Document

Document Corpus

Similarity Comparison

The picture can't be displayed.

Document A

Document B

Challenges

- Comparing the similarity of different parts between two documents is much more difficult
- Long document
- Structure and vocabulary
- Focus on semantics and syntactics

Summary

- Text Vectorization
- Word Embedding
- Word Embedding models
  - Word2vec
  - Glove
  - Fast Text
  - EMLo
- Building a model for word representation