Week 3
Distributional Semantics - Word Embeddings - Word Sense Disambiguation

Nhung Nguyen
slides courtesy of Phong Le

Recap

- Introduction to NLP and Text Mining
- Preprocessing:
  - Sentence segmentation and tokenization
  - POS tagging, parsing
- Information extraction:
  - relation extraction

Plan

1. Distributional Semantics
2. Word Embeddings
    a. Count-based approach
    b. Brief introduction to neural networks
    c. Prediction-based approach
3. Word Sense Disambiguation
4. Lab Exercise 3 Overview
5. Introduction to Coursework 1

Targets

1. Understand the concepts of lexical semantics, distributional semantics, word sense disambiguation, and their importance in NLP.
2. Understand the ideas and mechanisms of several word vector models based on term-document matrixes, term-term matrixes, and simple neural networks.
3. Be able to visualise word embeddings (for example debugging)

Materials

- Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft). https://web.stanford.edu/~jurafsky/slp3/ (Chap 6, 7, 18)
- http://www.scholarpedia.org/article/Neural_net_language_models
- Bengio et al. *A Neural Probabilistic Language Model*. J. Machine Learning Research (2003) 3:1137-1155.
- Mikilov et al. *Efficient Estimation of Word Representations in Vector Space*. arXiv preprint arXiv:1301.3781

Distributional Semantics

Slides are mostly based on Chapter 6 (Dan Jurafsky and James H. Martin. Speech and Language Processing. 3rd Edition, 2020)

How do we learn new words?

Do you like **rau-muống**?

- Look in a dictionary (or Google): [...] is a semi-aquatic, tropical plant grown as a vegetable for its tender shoots and it is not known where it originated. (Wikipedia)

How do we learn new words? (cont.)

- How did I learn (when I was a small kid)?

  My mom pointed to

How do we learn new words? (cont.)

Can we use any of the above methods to "teach" computers the meaning of a word?

- Looking into a dictionary: How to teach computers the meanings of *semi-aquatic*, *tropical* and the whole paragraph?
- Pointing to an object: How to teach computers to "understand" images?

Difficult problems!

How do we learn new words? (cont.)

Maybe you have seen

- Rau-muống is delicious sauteed with garlic.
- Rau-muống is superb over rice.
- ...rau-muống leaves with salty sauces...

and have seen

- ...spinach sauteed with garlic over rice...
- ...chard stems and leaves are delicious...
- ...collard greens and other salty leafy greens

→ rau-muống is a leafy green similar to these other leafy greens

How do we learn new words? (cont.)

- Which strategy is used? Similar contexts suggest similar meanings.

  → Distributional hypothesis

- But
  - do we need to understand the meaning of contexts ("sauteed with garlic")?
  - do we need to know the meanings of "spinach", "chard", "collard"?

- Turns out there are some simple workaround (computational) solutions

Distributional hypothesis (history)

The meaning of a word is its use in the language

Ludwig Wittgenstein (1889- 1951)

- It doesn't matter *computer* is called a fool as long as the new name is used indifferently with the old one.
    - How to make *computers* smart?
    - How to make *a fool* smart?

Distributional hypothesis (history, cont.)

If A and B have almost identical environments we say that they are synonyms.

Zellig Harris (1954)

We shall know a word by the company it keeps.

Firth (1957)

Distributional hypothesis (formalisation)

Given word w and all the contexts $C(w) = \{c_1, c_2, \ldots\}$ it appears within, then

$$\textbf{meaning}(w) = f(c_1, c_2, \ldots)$$

where f is a function compressing some statistics of $C(w)$ into a vector.

The ultimate goal: find f!

Co-occurrence vectors: word-word matrixes

1. Collect a lot of documents / sentences (from, e.g. Wikipedia)
   a. .... the first *digital* computers were developed.
   b. … the system stores enough *digital* data ...
2. Apply basic pre-processing steps: lowercase, tokenisation, lemmatisation
3. Count how many times a word *u* appearing with a word *v*

   count(*digital*, *computer*) = 1670

4. The meaning of word *u* is vector [count(u,v_1), count(u,v_2),...]

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	
strawberry	0	...	0	0	1	60	19	
digital	0	...	1670	1683	85	5	4	
information	0	...	3325	3982	378	5	13	

Pros

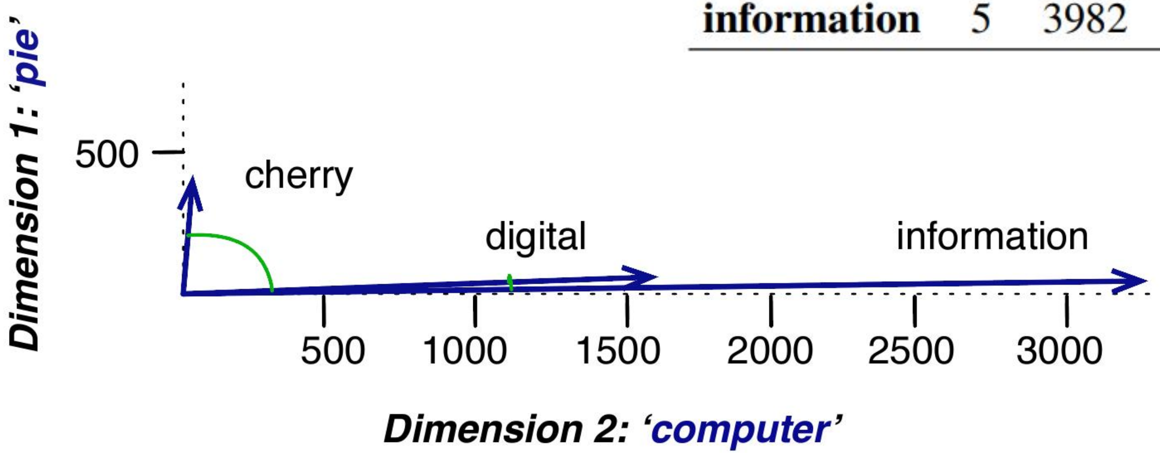
The meaning of a word is represented by a vector (named word vector), therefore

- we can compute the similarities between word meanings

cos(digital, information) = .996

cos(cherry, information) = .017

	pie	data	computer
cherry	442	8	2
digital	5	1683	1670
information	5	3982	3325



Pros (cont.)

The meaning of a word is represented by a vector (named word vector), therefore

- we can compute the similarities between word meanings
- we can visualise word meanings

[Demo: word visualisation]

Pros (cont.)

The meaning of a word is represented by a vector (named word vector), therefore

- we can compute the similarities between word meanings
- we can visualise word meanings
- we can directly use words as inputs to most machine learning algorithms

Cons

- Distributional semantics beyond words?
- Can distributional semantics capture all aspects of semantics?

Summary

- Distributional semantics is originated from distributional hypothesis
  - Tell me who your friends are, I’ll tell you who you are
- A word is represented by a co-occurrence word vector
- Co-occurrence word vectors can be used to:
  - detect the similarity among words
  - visualise word meanings
  - input to machine learning models
- But do they really capture all semantics aspects and beyond?