Natural Language Processing Lab
Week #4

Jason S. Chang 張俊盛 jason@nlplab.cc

TAs： Kevin Tuan 段凱文 kevintuan@nlplab.cc

Course Website:

Date:

How to represent the meaning of a word?

- Definition: meaning (Webster dictionary)
  - the idea that is represented by a word, phrase, etc.
  - the idea that a person wants to express by using words, signs, etc.
  - the idea that is expressed in a work of writing, art, etc

- However, definitions are not useful for a computer

Bag of Words

- Definition
  - The bag-of-words model is a simplifying representation used in natural language processing and information retrieval (IR).

Bag of words (BoW)

Very good drama although it appeared to have a few blank areas leaving the viewers to fill in the action for themselves. I can imagine life being this way for someone who can neither read nor write. This film simply smacked of the real world: the wife who is suddenly the sole supporter, the live-in relatives and their quarrels, the troubled child who gets knocked up and then, typically, drops out of school, a jackass husband who takes the nest egg and buys beer with it. 2 thumbs up… very very very good movie.

('the', 8),
(',', 5),
('very', 4),
('.', 4),
('who', 4),
('and', 3),
('good', 2),
('it', 2),
('to', 2),
('a', 2),
('for', 2),
('can', 2),
('this', 2),
('of', 2),
('drama', 1),
('although', 1),
('appeared', 1),
('have', 1),
('few', 1),
('blank', 1)
…..

Representing words as discrete symbols

We can regard words as discrete symbols:

hotel, conference, motel – a localist representation

Means one 1, the rest 0s

Such symbols for words can be represented by one-hot vectors:

```
motel = [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
hotel = [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
```

- What are the problems with one-hot encoding?

Problems with one-hot encoding

- Vector dimension = vocabulary size (~100,000)
  - Curse of Dimensionality
- Vectors are orthogonal to one another
  - Lack of similarity
- Low Information Content (too many zeros)

Problems with one-hot encoding

Example: if a user searches for “Seattle motel”, we would like to match documents containing “Seattle hotel”

```
motel = [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
hotel = [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
```

These two vectors are orthogonal
There is no natural notion of similarity for one-hot vectors!

Solution:

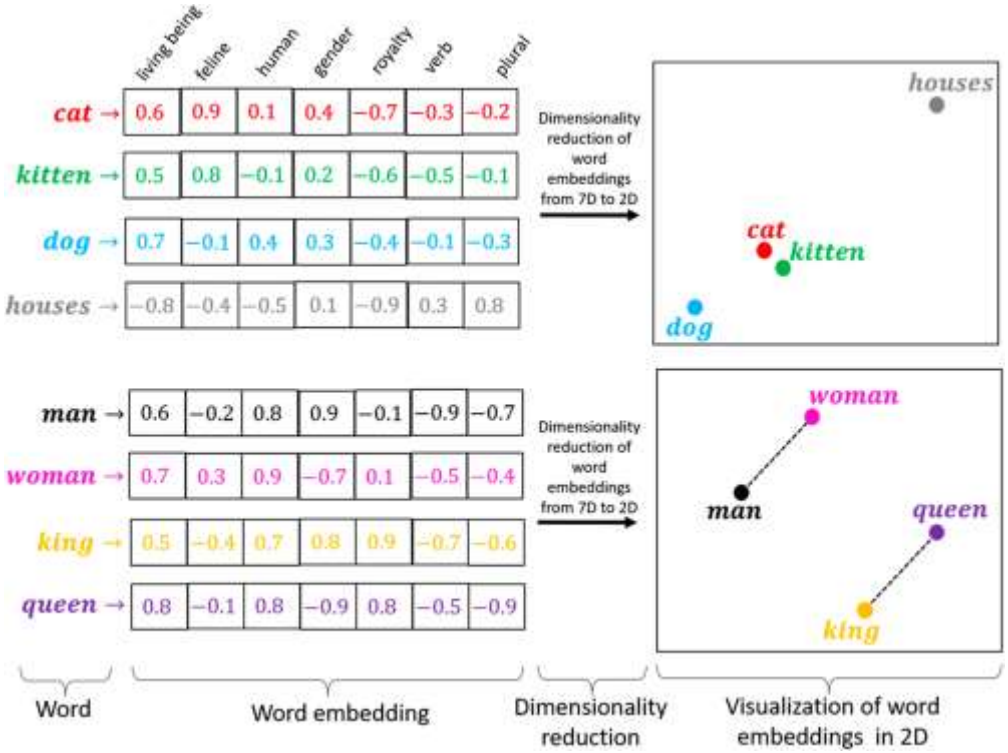
- Could try to rely on WordNet’s list of synonyms to get similarity?
- Learn to encode similarity in the vectors themselves

Representing words by their context

- **Distributional semantics:**
  - A word's meaning is given by the words that frequently appear close-by
  - One of the most successful ideas of modern statistical NLP!
- When a word *w* appears in a text, its context is the set of words that appear nearby (within a fixed-size window).
- We use many contexts of *w* to build up a representation of *w*

Word embedding

In NLP, word embedding is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning.



Word Vectors

- We can build a dense vector for each word, chosen so that it is similar to vectors of words that appear in similar contexts, measuring similarity as the vector dot (scalar) product

$$\textit{banking} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix} \qquad \textit{monetary} = \begin{pmatrix} 0.413 \\ 0.582 \\ -0.007 \\ 0.247 \\ 0.216 \\ -0.718 \\ 0.147 \\ 0.051 \end{pmatrix}$$

word2vec

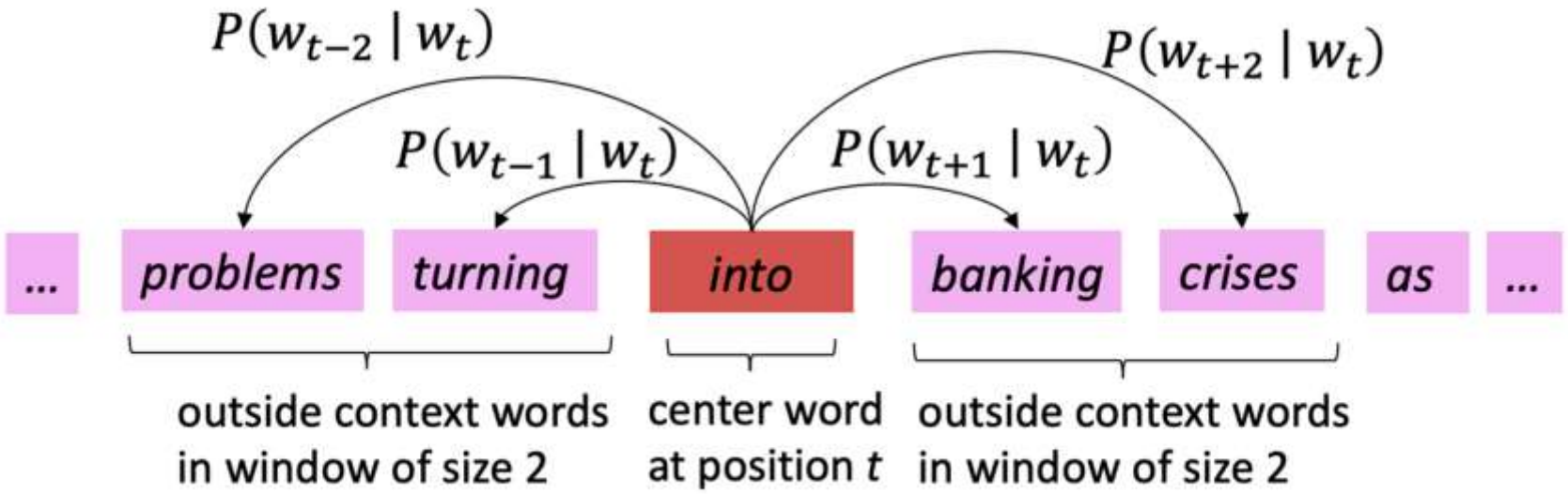
Word2vec (Mikolov et al. 2013) is a framework for learning word vectors

Idea:

- We are given a large corpus (“body”) of text with a large vocabulary
- Represent every word in the vocabulary by a vector
- Go through each position *t* in the text, which has a center word *c* and context (“outside”) words *o*
- Use word vectors *c* and *o* to calculate the probability of *o* given *c* (or vice versa) based on similarity(vector(*c*), vector(*o*))
- Keep adjusting the word vectors to maximize this probability

word2vec

Example windows and process for computing $P(w_{t+j} \mid w_t)$



Your turn

This week you will build a model to distinguish between good and bad phrases of the word "earn" (e.g., earn money), using word2vec.

Reminder:

- Make an appointment with TA to demo your implementation
- Change the filename to {studentID}.ipynb and submit to elearn
- Deadline: 15:29 p.m. Oct 13 (Thu)

References

Stanford NLP Course Material

- https://web.stanford.edu/class/cs224n/slides/cs224n-2022-lecture01-wordvecs1.pdf
- https://web.stanford.edu/class/cs224n/slides/cs224n-2022-lecture02-wordvecs2.pdf

Gensim word2vec documentation

- https://radimrehurek.com/gensim/models/word2vec.html

Relevant Researches

- Efficient Estimation of Word Representations in Vector Space
- GloVe: Global Vectors for Word Representation
- Improving Distributional Similarity with Lessons Learned from Word Embeddings