Word Embeddings

CS 6956: Deep Learning for NLP

THE UNIVERSITY OF UTAH

Overview

- Representing meaning
- Word embeddings: Early work
- Word embeddings via language models
- Word2vec and Glove
- Evaluating embeddings
- Design choices and open questions

Overview

- Representing meaning
- Word embeddings: Early work
- Word embeddings via language models
- Word2vec and Glove
- Evaluating embeddings
- Design choices and open questions

Vector space representations of words

Historically, a diverse collection of ideas and methods

- 1980s/1990s/2000s
  - Latent semantic analysis (LSA)
  - Probabilistic LSA, topic models

- 2000s/2010s
  - Word embeddings via neural language models
  - word2vec
  - Glove

What defines the context of a word?

Several answers possible

What defines the context of a word?

Several answers possible

1. **Entire documents**: Words that occur in the same documents are related
   - Example: `soccer` and `referee` may show up in the same document often because they share a topic

What defines the context of a word?

Several answers possible

1. **Entire documents**: Words that occur in the same documents are related
   - Example: `soccer` and `referee` may show up in the same document often because they share a topic

2. **Neighboring words**: Words that occur in the context of the same words carry similar meanings
   - Example: `NYC` and `Yankees` may be used in interchangably in certain contexts, but `NYC` and `baseball` may not.

Documents as context

- Arose in the information retrieval world
- Led to latent semantic analysis (LSA), topic models, latent Dirichlet analysis
- Captures relatedness between words

Neighboring words as context

- Typically uses a window around a word
- For example, suppose we consider a window of size 2 to the left and right

John sleeps during the **day** and works at night.
Mary starts her **day** with a cup of coffee.
John starts his **day** with an angry look at his inbox.

Neighboring words as context

- Typically uses a window around a word
- For example, suppose we consider a window of size 2 to the left and right

John sleeps during the **day** and works at night.
Mary starts her **day** with a cup of coffee.
John starts his **day** with an angry look at his inbox.

Neighboring words as context

- Typically uses a window around a word
- For example, suppose we consider a window of size 2 to the left and right

John sleeps during the **day** and works at night.
Mary starts her **day** with a cup of coffee.
John starts his **day** with an angry look at his inbox.

We have a co-occurrence vector

during	the	and	works	starts	her	his	with	a	an
1	1	1	1	2	1	1	2	1	1

Not showing entries with zeros, which will include all other words

Neighboring words as features

Commonly seen in NLP, especially with linear models

- Standard features before neural networks became common

However:

1. Sparsity can cause problems
2. High dimensionality can cause problems

In both cases, with regard to generalization and memory

Addressing sparsity and dimensionality

- Dimensionality reduction

- Project the word-word co-occurrence matrix to a lower dimensional space
  - Perform singular value decomposition
  - Suppose C is the co-occurrence matrix, then
    - $U, \Sigma, V^T = svd(C)$
    - Treat the rows of U as word embeddings

- Key idea: Word embeddings as ***dense, low dimensional*** vectors

Variants on this theme

1. Frequent words can dominate counts
   - Words like a, the, is, in, etc will occcur in the context of nearly every word
   - Control for this by putting an upper limit on the count. For eg: If a word occurs more than 100 times in a context, then restrict its count to 100.

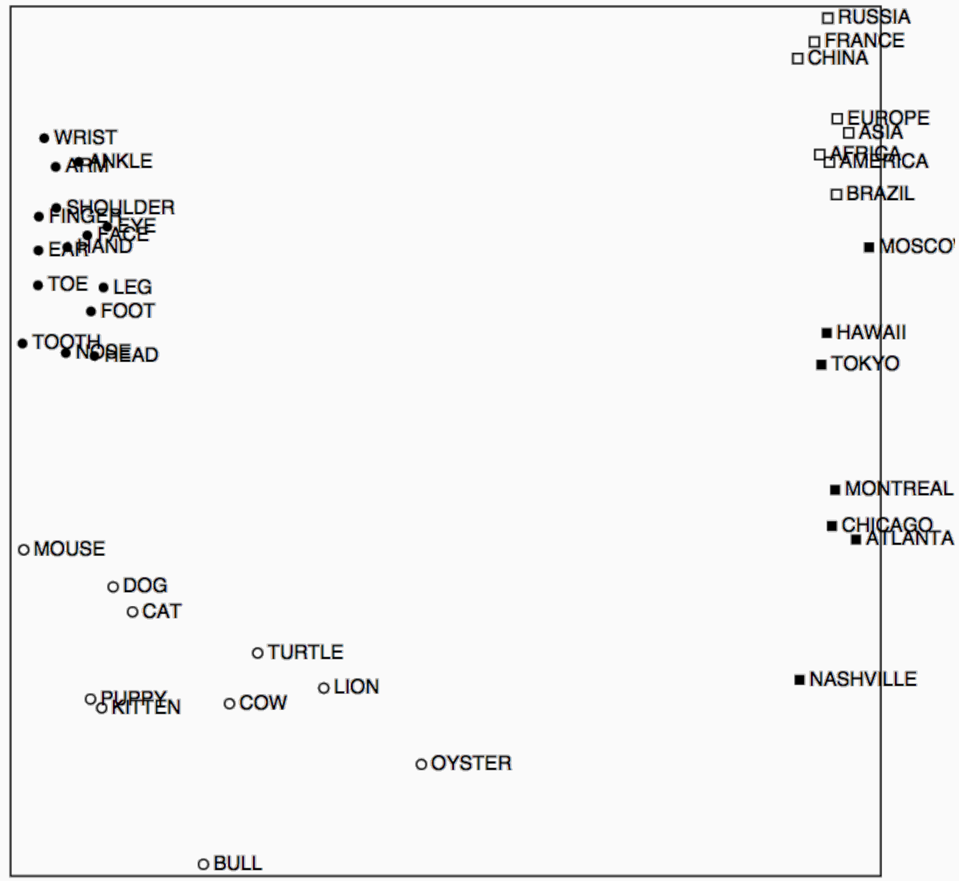
Variants on this theme

1. Frequent words can dominate counts
   – Words like a, the, is, in, etc will occcur in the context of nearly every word
   – Control for this by putting an upper limit on the count. For eg: If a word occurs more than 100 times in a context, then restrict its count to 100.

2. Instead of counts, we can use other properties of words in contexts
   – Eg: log frequencies, correlation coefficients, etc
   – All these will give us different embeddings
   – We will revisit this idea soon

Good news: The embeddings capture meaningful regularities

Both syntactic and semantic



Rohde, Douglas LT, Laura M. Gonnerman, and David C. Plaut. "An improved model of semantic similarity based on lexical co-occurrence." *Communications of the ACM* 8, no. 627-633 (2006): 116.

Bad news: SVD is slow

- The matrix at hand is huge
  - Rows/columns = Number of words

- Time complexity of SVD is cubic in this number
  - However, various incremental SVD algorithms exist

- But do we need to perform this computation at all?

Rohde, Douglas LT, Laura M. Gonnerman, and David C. Plaut. "An improved model of semantic similarity based on lexical co-occurrence." *Communications of the ACM* 8, no. 627-633 (2006): 116.