Neural Language Models

Sargur N. Srihari
srihari@cedar.buffalo.edu

This is part of lecture slides on Deep Learning:
http://www.cedar.buffalo.edu/~srihari/CSE676

Topics

1. N-gram Models
2. Neural Language Models
3. High-dimensional Outputs
4. Combining Neural Language Models with n-grams
5. Neural Machine Translation
6. Other Applications

Neural Language Models (NLMs)

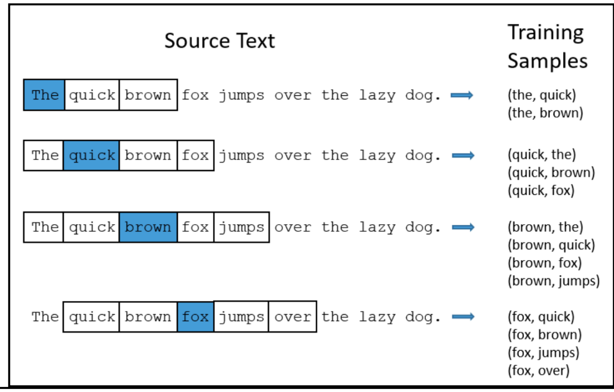
- Overcome the curse of dimensionality of n-gram modes
  - By using a distributed representation of words
- Unlike class-based n-gram models
  - NLMs are able to recognize that two words are similar
  - without losing the ability to encode each word as distinct from others

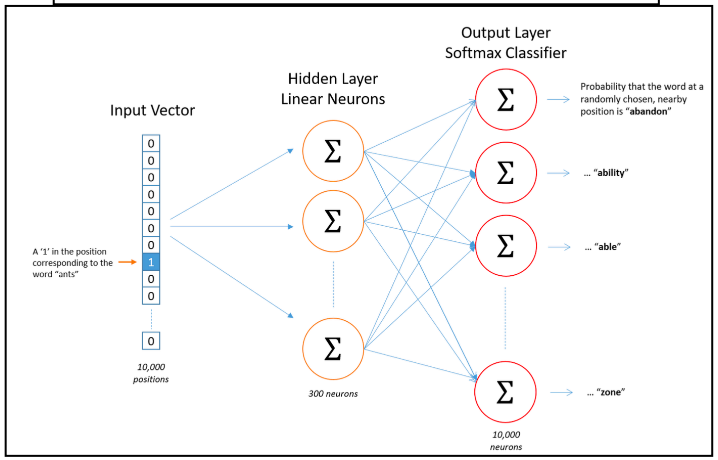
Strength of NLMs

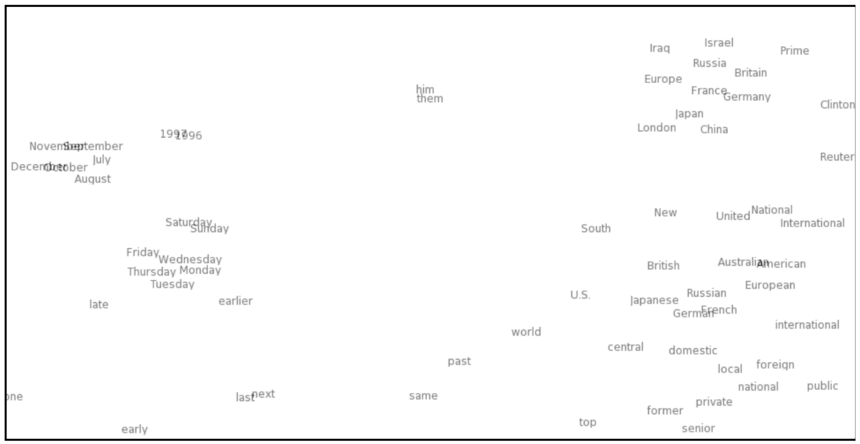
- Share statistical strength between one word (and its context) and other similar words and contexts
- Distributed representation allows model to treat words that have features in common similarly
- Curse of dimensionality handled by relating each training sentence to an exponential number of similar sentences

Word-to-Vec

- Training Data
- Word-to-vec
  - One-hot vector mapped to vector of 300
- Word embedding
  - Similar words are close together







Word-to-vec:

Represent noun by co-occurrences with 25 verbs*

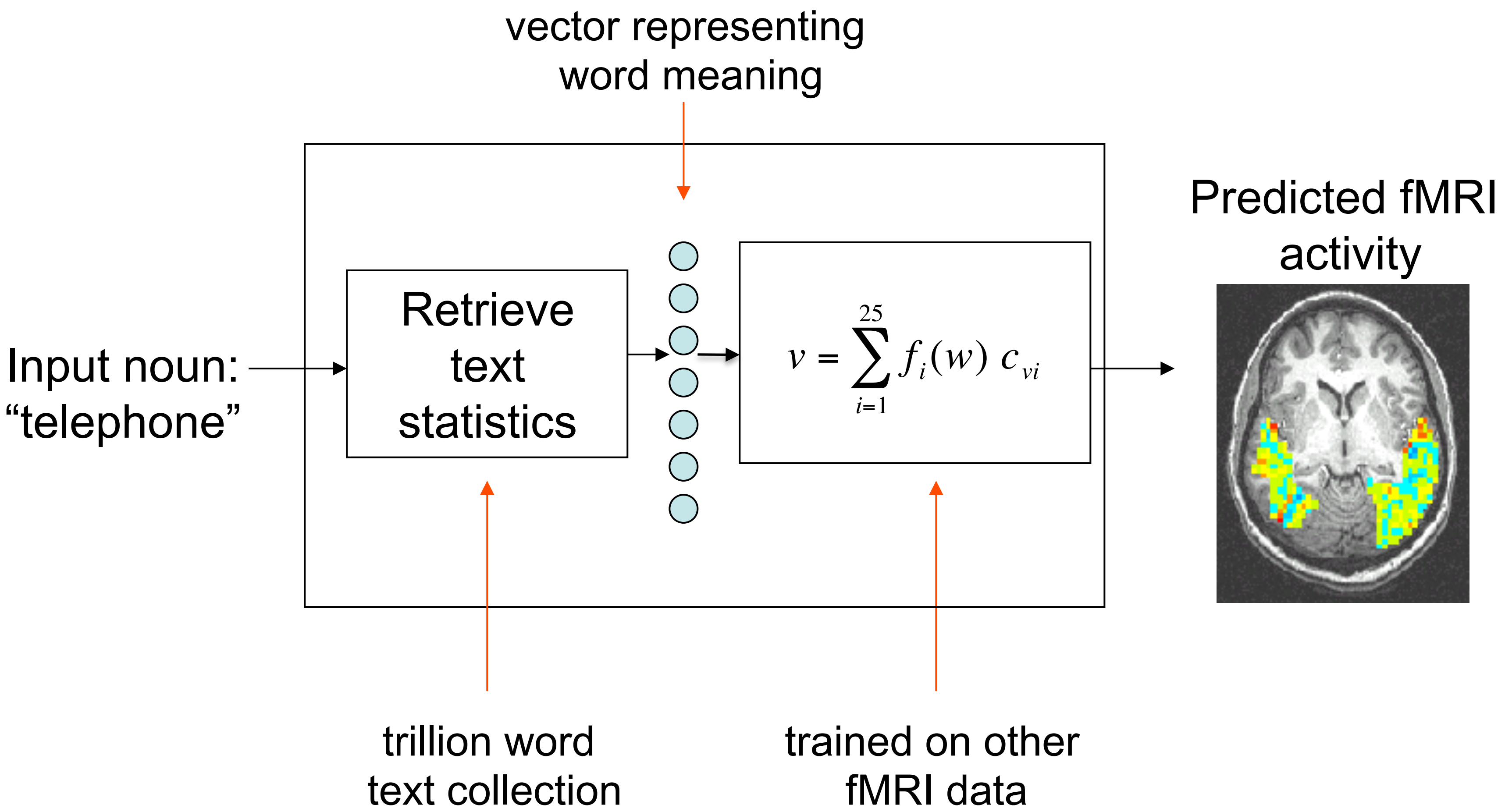
Semantic feature values:
“celery”
0.8368, eat
0.3461, taste
0.3153, fill
0.2430, see
0.1145, clean
0.0600, open
0.0586, smell
0.0286, touch
…
…
0.0000, drive
0.0000, wear
0.0000, lift
0.0000, break
0.0000, ride

Semantic feature values:
“airplane”
0.8673, ride
0.2891, see
0.2851, say
0.1689, near
0.1228, open
0.0883, hear
0.0771, run
0.0749, lift
…
…
0.0049, smell
0.0010, wear
0.0000, taste
0.0000, rub
0.0000, manipulate

* in a trillion word text collection

Neural model of language

[Mitchell et al., *Science*, 2008]

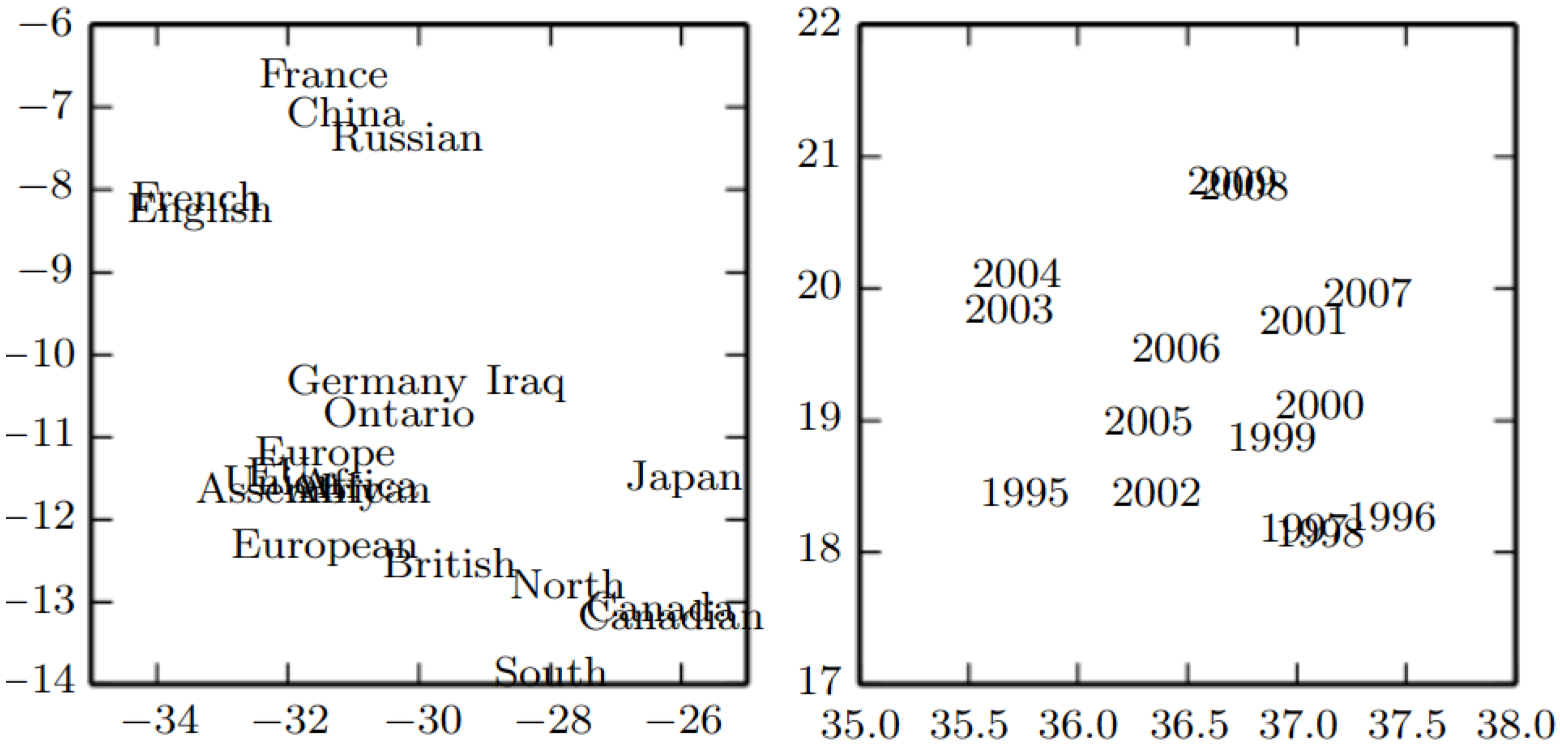


Word Vectors and Embedding

- View raw symbols as points in a space whose dimensionality is vocabulary size
- Embed those points in a space of lower dimension
- In original space every word is at distance $\sqrt{2}$ from every other word
- In embedding space words that appear frequently appear in similar contexts are close to each other

Word embedding

- 2-D visualization of word embeddings from a machine translation model
  - Zoom in where semantically related words have embeddings close to each other (countries, dates)



Word to Vec

- https://www.quora.com/How-does-word2vec-work
- From corpus to co-occurrence matrix
- SVD converts word to a fixed-length vector

Importance of Word Embedding

- Neural networks in other domains also define embeddings
  - E.g., convolutional neural network provides an image embedding
- Embedding in NLP is more interesting since natural language does not originally lie in a real-valued vector space

Word Embedding

- A word embedding $\mathrm{W: words} \rightarrow \mathrm{R}^n$ is a parameterized function mapping words in some language to high dimensional vectors (perhaps 200 to 300 dimensions), e.g.,

  $\mathrm{W}(\text{'cat'})=(0.2, -0.4, 0.7,\ldots)$ $\mathrm{W}(\text{'mat'})=(0.0,0.6,-0.1,\ldots)$
- Typically the function is a lookuptable, parameterized by a matrix θ, with a row for each word: $\mathrm{W}_\theta(\mathrm{w}_\mathrm{n})=\theta_\mathrm{n}$

Learning Word Embeddings

- W initialized with random vectors for each word
- It learns to have meaningful vectors in order to perform some task
  - Task: train network to tell whether 5-gram is valid
  - Training data: legal 5-grams,e.g., $\mathrm{cat\ sat\ on\ the\ mat}$)
- Make half of them nonsensical by switching with a random word ($\mathrm{cat\ sat}\ \mathbf{song}\ \mathrm{the\ mat}$)

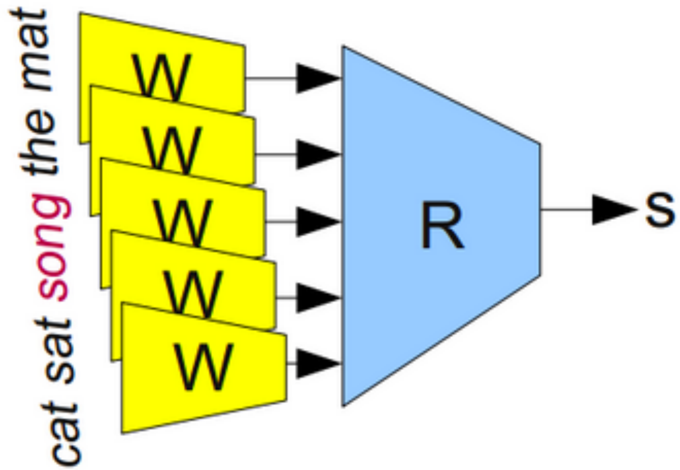
Network to determine valid 5-grams

- Model runs each word in 5-gram through W to get vector representing it
- Feed those into

  R which predicts if

  5-gram is valid or broken.

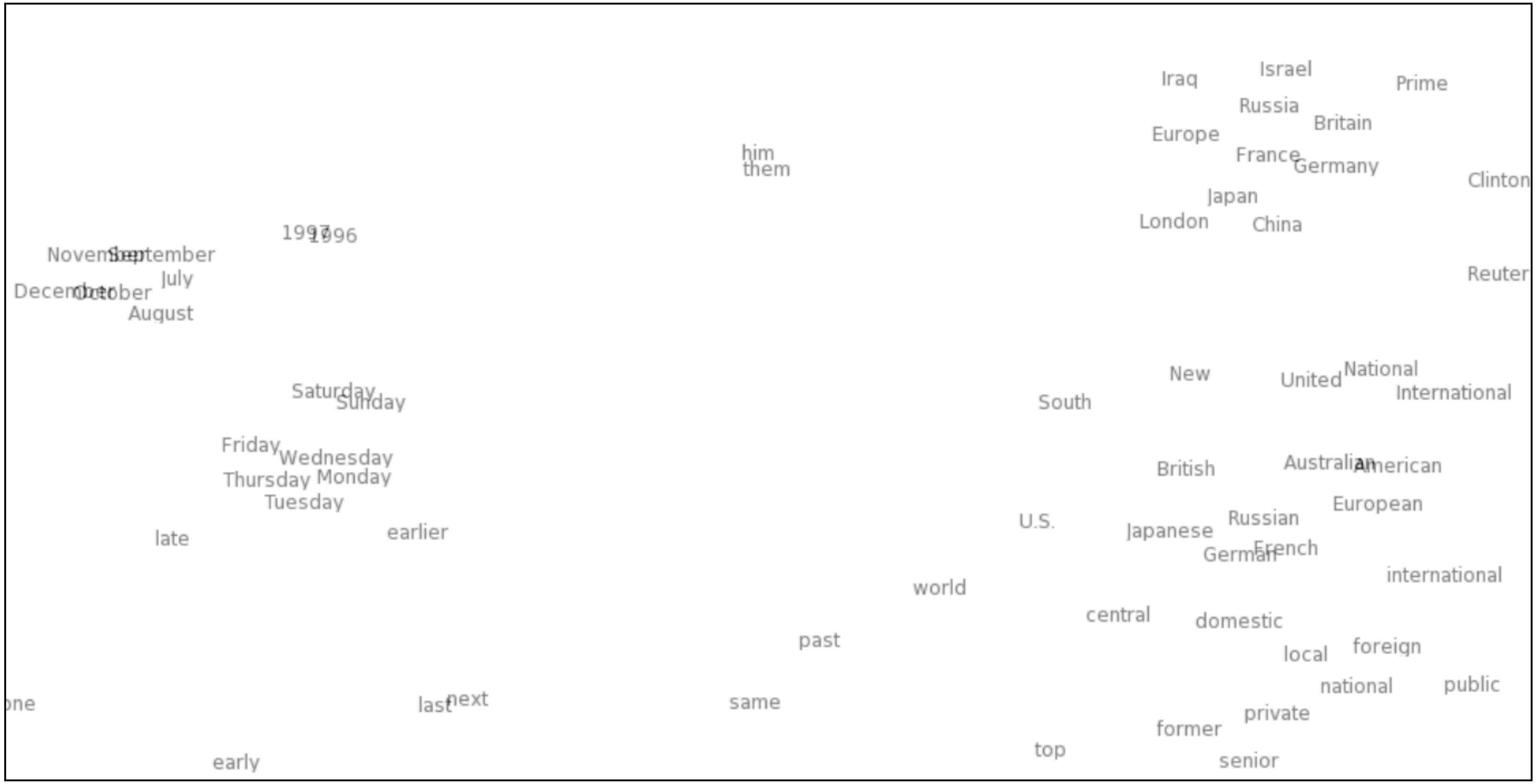
  We would like



  - $\mathrm{R(W(cat),W(sat),W(on)W(the)W(mat))=1}$
  - $\mathrm{R(W(cat),W(sat),W(song)W(the)W(mat))=1}$
- Need to learn parameters for W and R
  - R is not as interesting as W
  - Entire point of task is to learn W

Visualizing word embedding

- t-SNE: a sophisticated technique for visualizing high-dimensional data



  - Map makes a lot of intuitive sense to us. Similar words are close together

Words closest in the embedding

- Which words have embeddings closest to a given word?

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

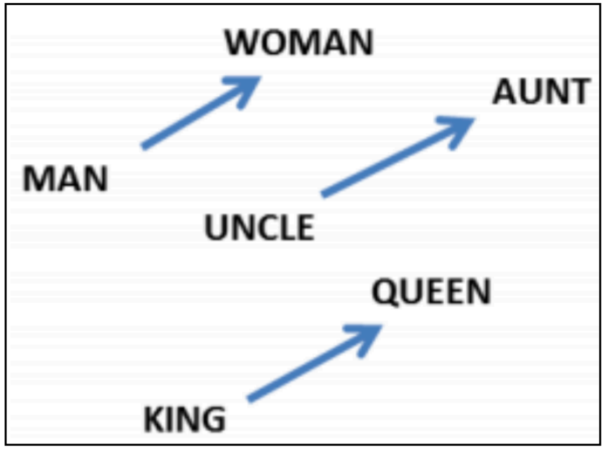
Power of Word Embeddings

- Similar words being close together allows us to generalize from one sentence to a class of similar sentences
- Not just word for synonym but switching a word for a word in a similar class
- E.g., wall is blue→ wall is red

  wall is blue→ ceiling is red

Word embeddings and analogies

- Analogies between words are encoded in difference vectors between words
  - E.g., constant male-female difference vector
    - W(woman)-W(man)≈W(aunt)-W(uncle)
    - W(woman)-W(man)≈W(queen)-W(king)
- Not surprising, since
  - we write "she is the aunt" but "he is the uncle"



Word embeddings & relationship pairs

- More sophisticated relationships are encoded

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

All these are side-effects

- All these properties of W are side-effects
  - We didn’t try to have similar words close together
  - We didn’t try to have analogies encoded with difference vectors
- All we tried to do was a simple task, whether a sentence was valid
  - These properties popped out of optimization process
- Neural networks learn better ways to represent data automatically

Importance of Word Embedding

- Neural networks in other domains also define embeddings
  - E.g., convolutional neural network provides an image embedding
- Embedding is NLP is more interesting since natural language does not originally lie in a real-valued vector space
- Using distributed representations is also used with PGM hidden variables