Word Embeddings

Lifu Huang

lifuh@vt.edu

Torgersen Hall, Suite 3160E

Slides adapted from Julia Hockenmaierr

What do words mean and how can we represent that?

S1: We met at a *bar* called the Flamingo

S2: He smashed the window with a *bar*

S3: We found our way *barred* by rocks.

Do we want to represent…
… that a "bar" are places to have a drink?
… that a "bar" is a long rods?
… that to "bar" something means to block it?

Two approaches to represent words

- The lexicographic tradition aims to capture the information represented in lexicons, dictionaries, etc.

- The distributional tradition aims to capture the meanings of words based on large amounts of raw text

The lexicographic tradition

- Uses resources such as lexicons, dictionaries, ontologies etc. that capture explicit knowledge about word meanings

- Assumes words have discrete word senses:

  e.g., bank1 = financial institution;   bank2 = river bank, etc.

- May capture explicit relations between words

  e.g., ”dog” is an “animal”, “cars” have “wheels”, etc.

How do we represent words traditionally?

- As atomic symbols?
  - e.g., as in a traditional n-gram language model, or when we use them as explicit features in a classifier

- This is equivalent to very high-dimensional one-hot vectors:

e.g., Bag of Words model

high = [1, 0, 0, …, 0], tall = [0, 1, 0, …, 0], cat = [0, 0, 1, …, 0]

The Distributional Tradition

- Use large corpora of raw text to learn the meaning of words from the contexts in which they occur

- Map words to sparse vectors that capture corpus statistics

- State-of-the-art: use neural nets to learn dense vector embeddings from very large corpora

The Distributional Hypothesis

- Zellig Harris (1954):

"… oculist and eye-doctor … occur in almost the same environments …"

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

- John R. Firth (1957):

"You shall know a word by the company it keeps."

The contexts in which a word appears tells us a lot about what it means.
Words that appear in similar contexts have similar meanings.

Why do we care about contexts?

What is tezgüino?

A bottle of **tezgüino** is on the table.
Everybody likes **tezgüino**.
Tezgüino makes you drunk.
We make **tezgüino** out of corn.
(Lin, 1998; Nida, 1975)

Corpus

A bottle of **wine** is on the table.
There is a **beer** bottle on the table
Beer makes you drunk.
We make **bourbon** out of corn.
Everybody likes **chocolate**
Everybody likes **babies**

We don't know exactly what tezgüino is, but since we understand these sentences, it's likely an alcoholic drink.

Could we *automatically* identify that tezgüino is like beer?

A large corpus may contain sentences such as:

Beer makes you drunk

Word Embeddings

- A word embedding is a function that maps each word to a single vector
  - these vectors are typically dense and have much lower dimensionality than the size of the vocabulary
  - this mapping function typically ignores that the same string of letters may have different senses (dining *table* vs. a *table* of contents)
  - this mapping function typically assumes a fixed size vocabulary (so an UNK token is still required)

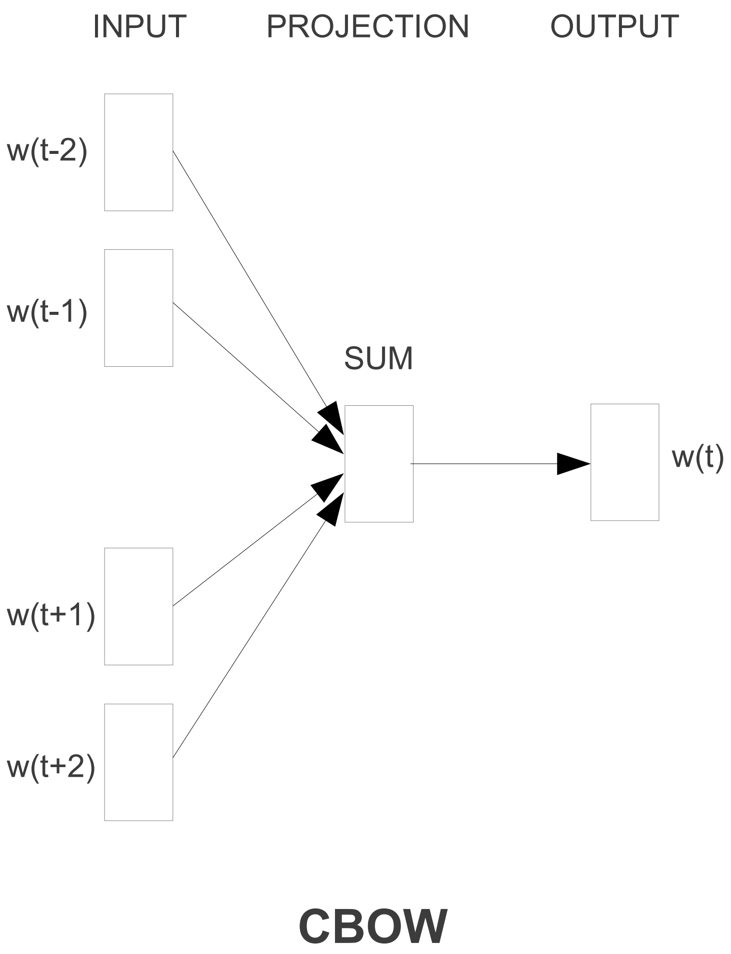
Word2Vec Embeddings

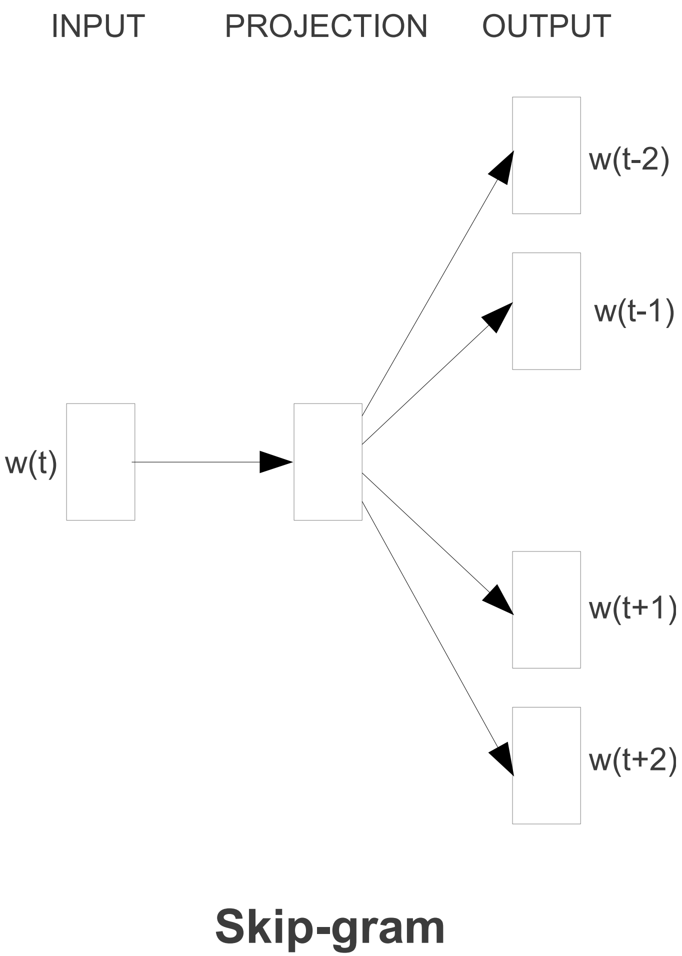
- **Main idea**
  - Use a classifier to predict which words appear in the context of (i.e. near) a target word, or vice versa
    This classifier induces a dense vector representation of words (embedding)
  - Words that appear in similar contexts (that have high distributional similarity) will have very similar vector representations.
  - These models can be trained on large amounts of raw text (and pre-trained embeddings can be downloaded)

Word2Vec (Mikolov et al. 2013)

- The first really influential dense word embeddings
- Two ways to think about Word2Vec:
  - a simplification of neural language models
  - a logistic regression classifier
- Variants of Word2Vec
  - Two different context representations: CBOW or Skip-Gram
  - Two different optimization objectives: Negative sampling (NS) or hierarchical softmax

Word2Vec Architectures





CBOW: predict target from context

(CBOW=Continuous Bag of Words)

Training sentence:

... lemon, a tablespoon of **apricot** jam a pinch ...

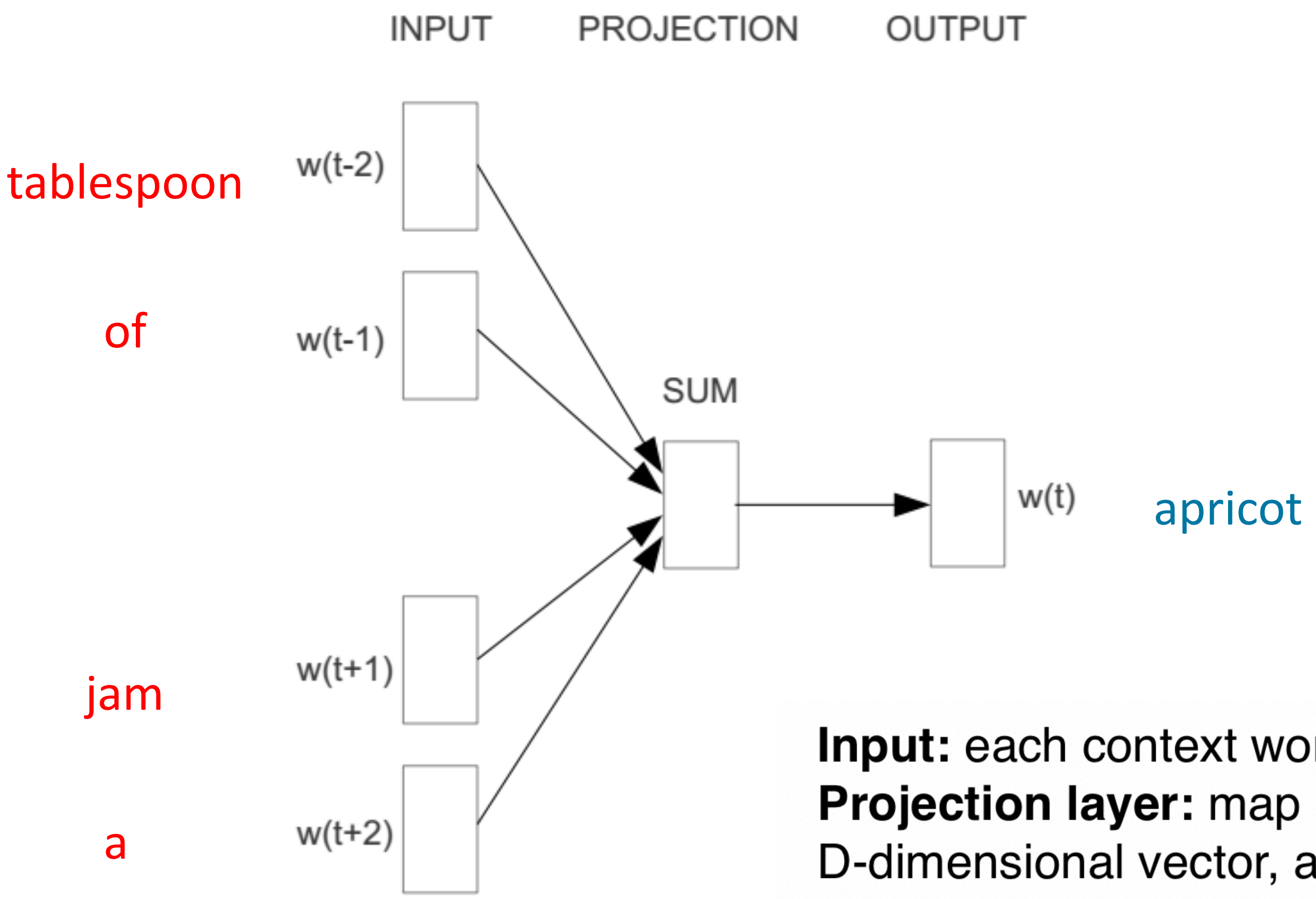
c1 c2 t c3 c4

Given the surrounding context words (tablespoon, of, jam, a), predict the target word (apricot).

Input: each context word is a one-hot vector
Projection layer: map each one-hot vector down to a dense D-dimensional vector, and average these vectors
Output: predict the target word with softmax

CBOW: Continuous Bag of Words

- e.g., *"a tablespoon of apricot jam a pinch"* , window_size = 2



Input: each context word is a one-hot vector
Projection layer: map each one-hot vector down to a dense D-dimensional vector, and average these vectors
Output: predict the target word with softmax

Skipgram: predict context from target

Training sentence:

... lemon, a tablespoon of **apricot** jam a pinch ...

c1 c2 t c3 c4

Given the target word (apricot), predict the surrounding context words (tablespoon, of, jam, a),

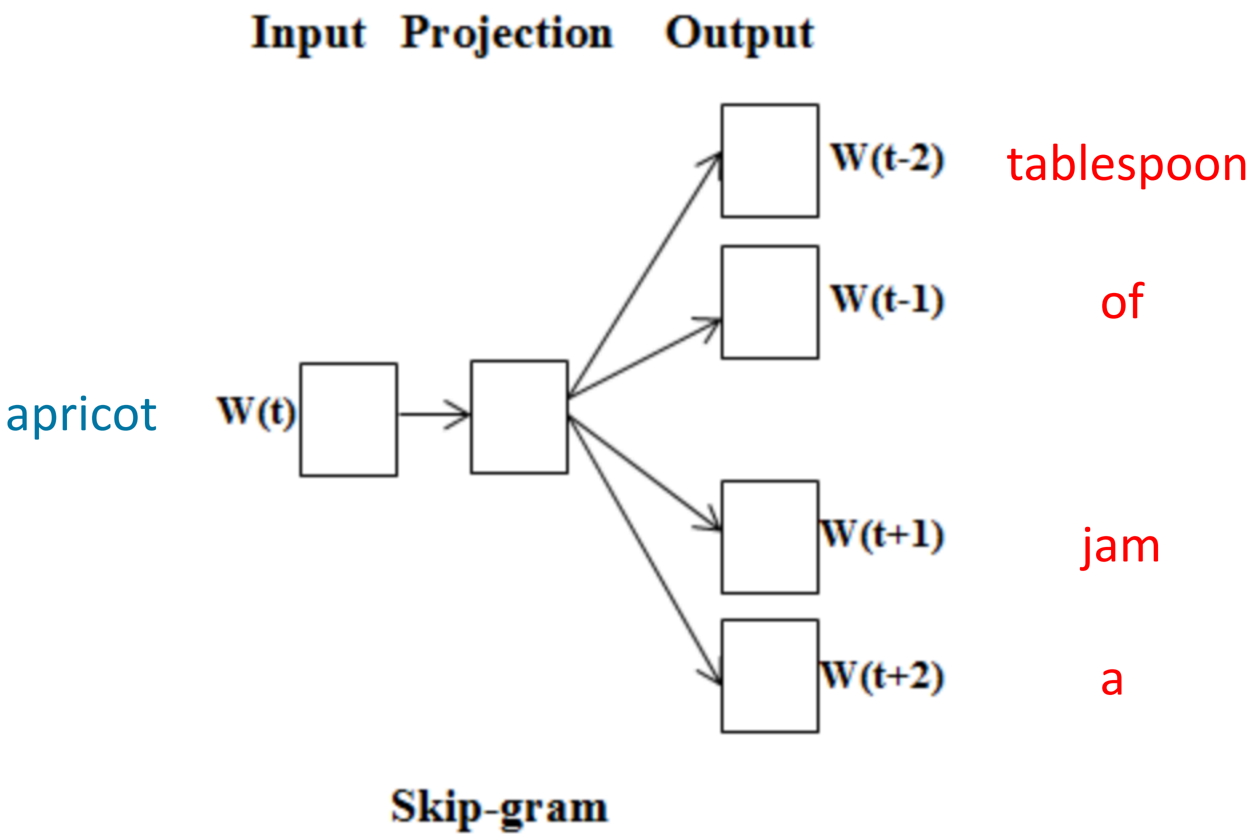
Input: each target word is a one-hot vector

Projection layer: map each one-hot vector down to a dense D-dimensional vector, and average these vectors

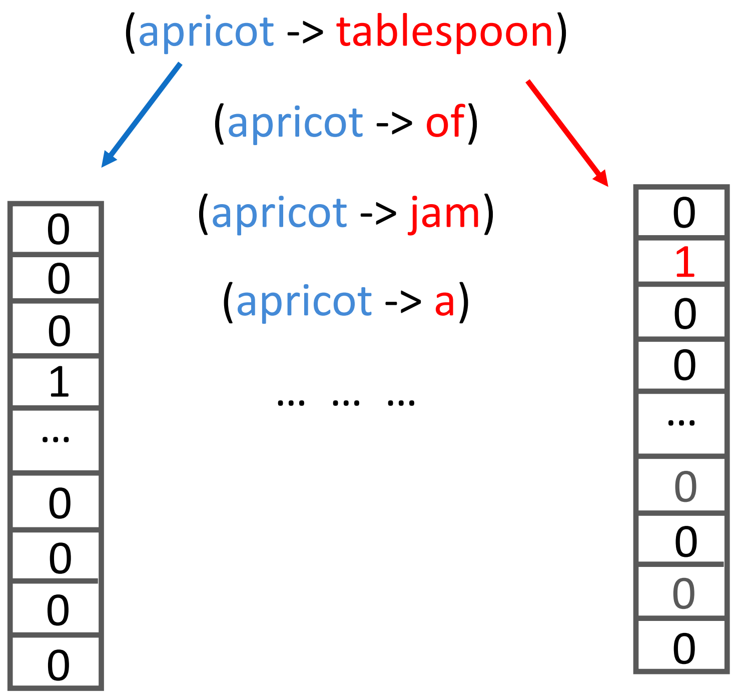
Output: predict the context word with softmax

Skip-gram

- e.g., *“a tablespoon of apricot jam a pinch”* , window_size = 2

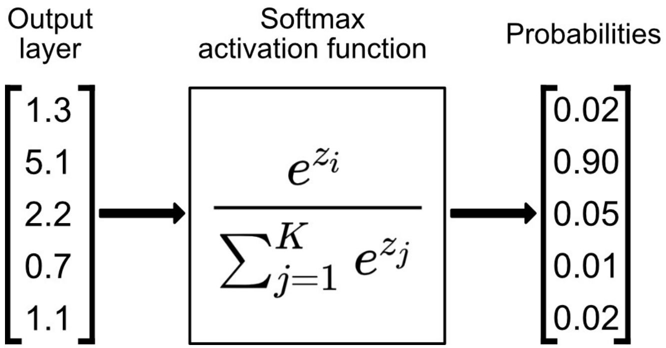


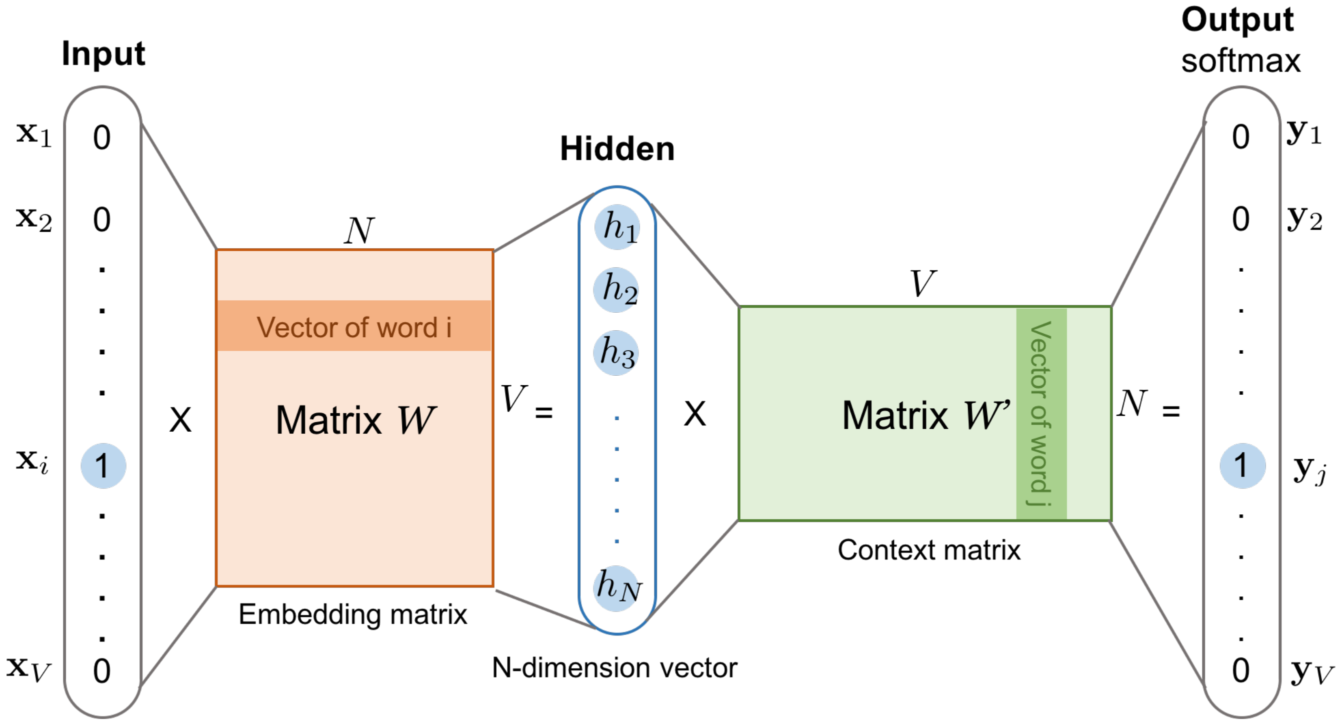
Training Data



Skip-gram

- Embedding matrix W: $V*N$
  - project a one-hot vector into its corresponding low-dimensional embedding;
- Context matrix W^T: $N*V$
  - project a low-dimensional embedding to its context by computing the dot product of two words' embeddings
- softmax:
  - multinomial logistic regression





Negative Sampling

- Skipgram aims to optimize the average log probability of the data

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j} \mid w_t) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log\left(\frac{\exp(w_{t+j}w_t)}{\sum_{k=1}^{V}\exp(w_k w_t)}\right)$$

- But computing the partition function $\sum_{k=1}^{V}\exp(w_k w_t)$ is very expensive

- Why we need the denominator?
- Can we do something simpler? – Negative Sampling

$$\log\sigma(w_T \cdot w_c) + \sum_{i=1}^{k} E_{w_i \sim P(w)}\left[\log\sigma(-w_T w_i)\right] \qquad \text{with } \sigma(x) = \frac{1}{1+\exp(-x)}$$

target word context negative context

Negative Sampling

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0}\log p(w_{t+j}\mid w_t) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0}\log\left(\frac{\exp(w_{t+j}w_t)}{\sum_{k=1}^{V}\exp(w_k w_t)}\right)$$

- Basic Idea
  - For each actual positive target-context word pair, sample k negative examples consisting of the target word and a randomly sampled word
  - Train a model to predict a high conditional probability for the actual positive context words, and a low conditional probability for the sampled negative context words

This can be reformulated as (approximated by) predicting whether a word-context pair comes from the actual positive data or from the sampled negative data

$$\log\sigma(w_T\cdot w_c) + \sum_{i=1}^{k} E_{w_i\sim P(w)}\left[\log\sigma(-w_T w_i)\right]$$

with $\sigma(x) = \dfrac{1}{1+\exp(-x)}$

target word   context   negative context

Skip-gram Training Data

- e.g., *"a tablespoon of apricot jam a pinch"* , window_size = 2

Positive examples

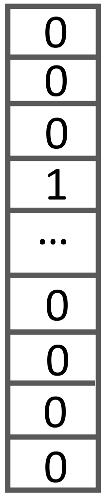
(apricot -> tablespoon)

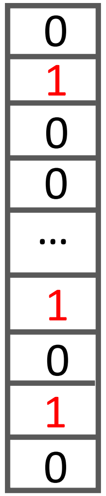
(apricot -> of)

(apricot -> jam)

(apricot -> a)

… … …





Sample K negative examples

(apricot -> aardvark)

(apricot -> puddle)

(apricot -> lemon)

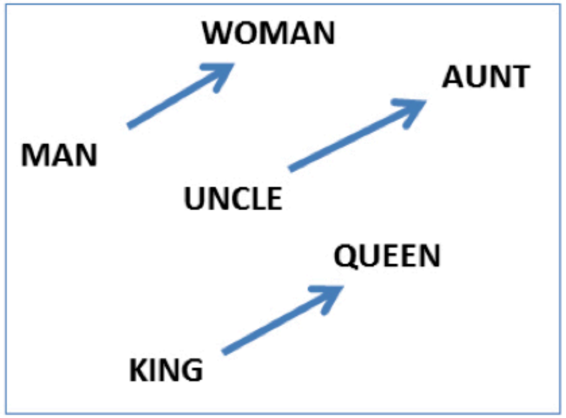
(apricot -> at)

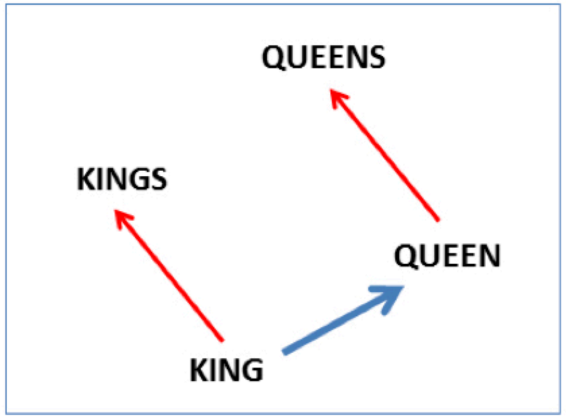
… … …

Some interesting results

- Word Analogy: Embeddings capture relational meaning

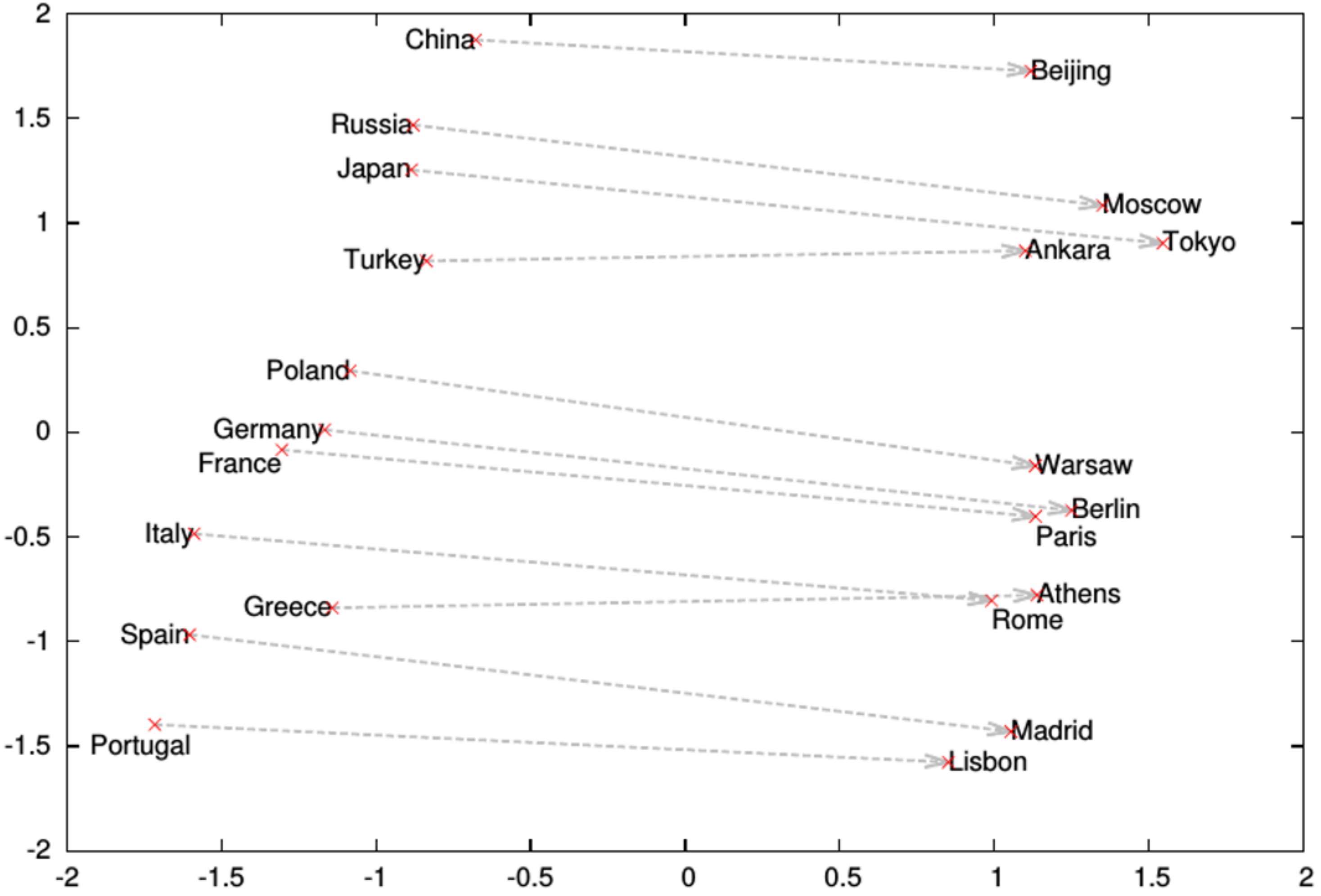
- vector (‘king’) – vector (‘man’) + vector (‘woman’) = vector (‘queen’)
- vector (‘Paris’) – vector (‘France’) + vector (‘Italy’) = vector (’Rome’)





- Why?

Word analogies



www.cs.ucr.edu/~vagelis/classes/CS242/slides/word2vec.pptx

More Analogy Pairs

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Limitations of word embeddings

- Static or dynamic?
  - there is a beer bottle on the table
  - Beer makes you drunk
- A word may have different senses in different sentences, e.g.,
  - the militants opened fire on the troops as the soldiers approached… [Shoot/attack]
  - the car was now on fire … [burning]
  - we have to fire him for dishonesty … [end position]

Limitations of word embeddings

- How to measure the similarity of words using their embeddings
  - cosine similarity? dot product?
- How to learn task specific embeddings?
  - e.g., sentiment related embeddings
- Different words from different languages and even objects/images are referring to the same concept
  - how to learn concept level language universal embeddings?
  - …

Dense embeddings you can download and explore

- Word2vec (Mikolov et al.,)
  - https://code.google.com/archive/p/word2vec/
  - Demo: http://bionlp-www.utu.fi/wv_demo/

- Fasttext:
  - https://fasttext.cc/docs/en/crawl-vectors.html

- Glove:
  - https://nlp.stanford.edu/projects/glove/