# Improving Word Representations via Global Context and Multiple Word Prototypes (Huang et al. 2012)

Dang Hoang Dung Nguyen, Jennifer Mell

Ruprecht-Karls-Universität Heidelberg

*Seminar: Embeddings*

*Dozenten: Prof. Dr. Katja Markert, Dr. Ines Rehbein*

*Sommersemester 2019*

09.07.2019

# Overview

# Motivation - Homonyms

- S: (n) **bank** (sloping land (especially the slope beside a body of water)) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*

excerpt of WordNet search for 'bank'

# Motivation - Homonyms

- S: (n) **bank** (sloping land (especially the slope beside a body of water)) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*

excerpt of WordNet search for 'bank'

- S: (n) **vector** (a variable quantity that can be resolved into components)
- S: (n) **vector** (a straight line segment whose length is magnitude and whose orientation in space is direction)
- S: (n) **vector**, transmitter (any agent (person or animal or microorganism) that carries and transmits a disease) *"mosquitos are vectors of malaria and yellow fever"; "fleas are vectors of the plague"; "aphids are transmitters of plant diseases"; "when medical scientists talk about vectors they are usually talking about insects"*
- S: (n) **vector** ((genetics) a virus or other agent that is used to deliver DNA to a cell)

WordNet search for 'vector'

# Recap: homographs, homonyms

## Homographs

Two words that have the same spelling but different meanings
ex.: bass, close, minute, ...

## Homonyms

Two words that have the same spelling and pronunciation but different meanings
ex.: bat, just, patient, ...

# POS tags

Same spelling but different POS tag and meaning:

- beat (n.) - to beat so. (v.)
- bear (n.) - to bear sth. (v.)
- light (n.) - light (adj.)
- bark (n.) - to bark (v.)

More on that later!

# Motivation - global context

- **Apple** is a kind of fruit.
- **Apple** releases its new ipads.

- Basalt is the commonest volcanic **rock**.
- **Rock** is the music of teenage rebellion.

from: Li & Jurafsky, 2015

# Motivation - global context

How similar are ...

# Motivation - global context

How similar are ...

1. mouse - cat

# Motivation - global context

How similar are ...

1. mouse - cat
2. mouse - keyboard

# Motivation - global context

How similar are ...

1. mouse - cat
2. mouse - keyboard
3. cat - keyboard

# Motivation - global context

How similar are ...

1. mouse - cat
2. mouse - keyboard
3. cat - keyboard

## Context

... changes meaning of a word
... changes which meaning of a word comes to mind
... provides topical information
... should be taken into account when creating embeddings!

# Motivation

- different word senses so far not represented in vector space
- worst case: embedding doesn't capture any sense well
- homonomy and context dependency: wide-spread
- word senses in datasets?

# Training Objective

## Given

**Input:** A word sequence $s$, document $d$

**Output:** correct last word in $s$

## Ranking Cost *(Collobert & Weston, 2008)*

$$C_{s,d} = \sum_{w \in V} \max(0, 1 - g(s,d) + g(s^w, d)) \quad (1)$$

$g(\cdot,\cdot)$: scoring function $s^w$: $s$ with last word replaced by word $w$

$g(s,d)$ should be larger than $g(s^w,d)$ by a margin of 1
$\Rightarrow$ minimize ranking loss for each $(s,d)$ found in corpus

# Neural Network Architecture

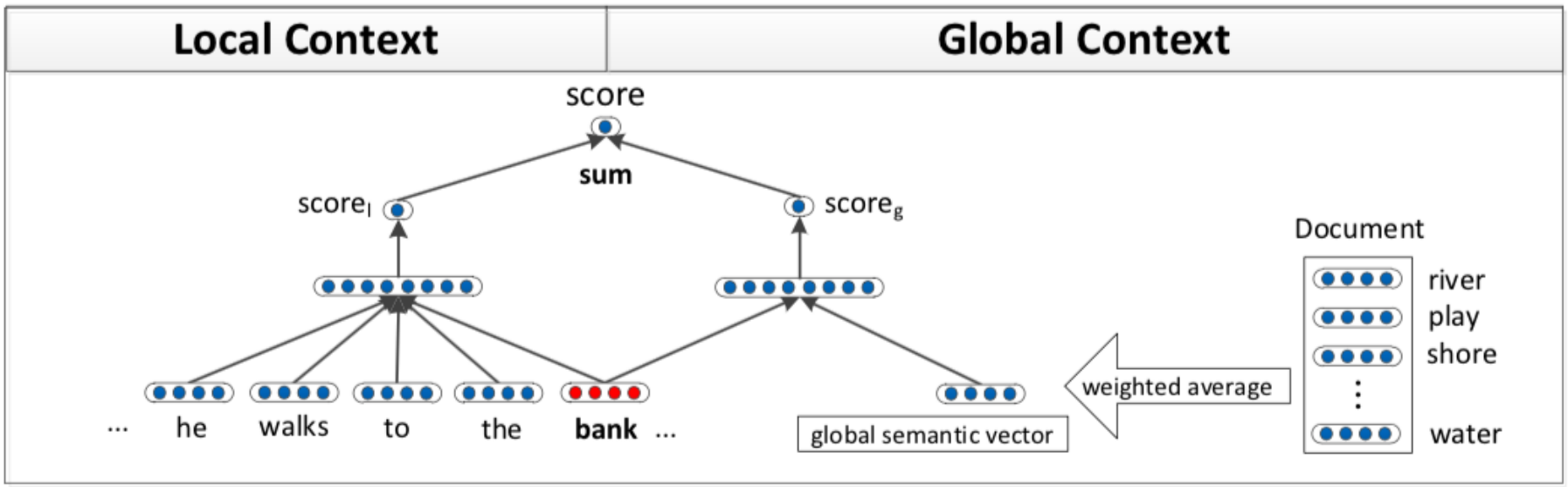



Figure 1: An overview of our neural language model. The model makes use of both local and global context to compute a score that should be large for the actual next word (*bank* in the example), compared to the score for other words. When word meaning is still ambiguous given local context, information in global context can help disambiguation.

# Local Context

**Local score** preserves word order and syntactic information.

**Input:** ordered list of vectors $x = (x_1, x_2, \ldots, x_m)$

**Output:**

$$a_1 = f(W_1[x_1; x_2; \ldots; x_m] + b_1) \quad (2)$$

$$score_l = W_2 a_1 + b_2 \quad (3)$$

- a neural network with one hidden layer
- $[x_1; x_2; \ldots; x_m]$: Concatenation of $m$ word embeddings representing $s$
- $x_i$: a column in embedding matrix $L \in \mathbb{R}^{n \times |V|}$, $|V|$: size of the vocabulary
- $f$: element-wise activation function
- $a_1 \in \mathbb{R}^{h \times 1}$: activation of hidden layer with $h$ hidden nodes
- $W_1 \in \mathbb{R}^{h \times (mn)}$, $W_2 \in \mathbb{R}^{1 \times h}$: first, second layer weights
- $b_1$, $b_2$: biases of each layer

# Global Context

**Global score** captures more of the semantics and topics of the document (similar to bag-of-words features).

**Input:** document as an ordered list of word embeddings $d = (d_1, d_2, \ldots, d_k)$

**Output:**

$$a_1^{(g)} = f(W_1^{(g)}[c; x_m] + b_1^{(g)}) \tag{4}$$

$$score_g = W_2^{(g)} a_1^{(g)} + b_2^{(g)} \tag{5}$$

* a two-layer neural network

- $[c; x_m]$: Concatenation of weighted average document vector and vector of last word in $s$
- $a_1^{(g)} \in \mathbb{R}^{h^{(g)} \times 1}$: activation of hidden layer with $h^{(g)}$ hidden nodes
- $W_1^{(g)} \in \mathbb{R}^{h^{(g)} \times (2n)}$, $W_2^{(g)} \in \mathbb{R}^{1 \times h^{(g)}}$: first, second layer weights
- $b_1^{(g)}$, $b_2^{(g)}$: biases of each layer

# Global Context

**Weighted average of all word vectors in document**

$$c = \frac{\sum_{i=1}^{k} w(t_i) d_i}{\sum_{i=1}^{k} w(t_i)} \tag{6}$$

$w(\cdot)$: weighting function that captures importance of word $t_i$ in document
here: **idf-weighting**

# Final Score

$$score = score_l + score_g \tag{7}$$

# Learning

- Randomly choose a word from dictionary as *corrupt* example for each sequence-document pair $(s, d)$ to sample gradient of the objective
- Take derivative of ranking loss with respect to parameters:
  - weights of the neural network: updated via backpropagation
  - embedding matrix $L$: word representations

# Motivation

- Words have multiple meanings
- Single-prototype models represent only one representation for each word
  $\Rightarrow$ can not capture different meanings

# Motivation

- Words have multiple meanings
- Single-prototype models represent only one representation for each word
  $\Rightarrow$ can not capture different meanings
- Representation of one of the meanings is influenced by all meanings of the word
- Using all contexts of a homonymous or polysemous word to build a single prototype
  $\Rightarrow$ none of the meanings is well represented

# Motivation

- Words have multiple meanings
- Single-prototype models represent only one representation for each word
  $\Rightarrow$ can not capture different meanings
- Representation of one of the meanings is influenced by all meanings of the word
- Using all contexts of a homonymous or polysemous word to build a single prototype
  $\Rightarrow$ none of the meanings is well represented

$\longrightarrow$ Multi-prototype model uses multiple representations to capture different senses and usages of a word. *(Reisinger and Mooney, 2010b)*

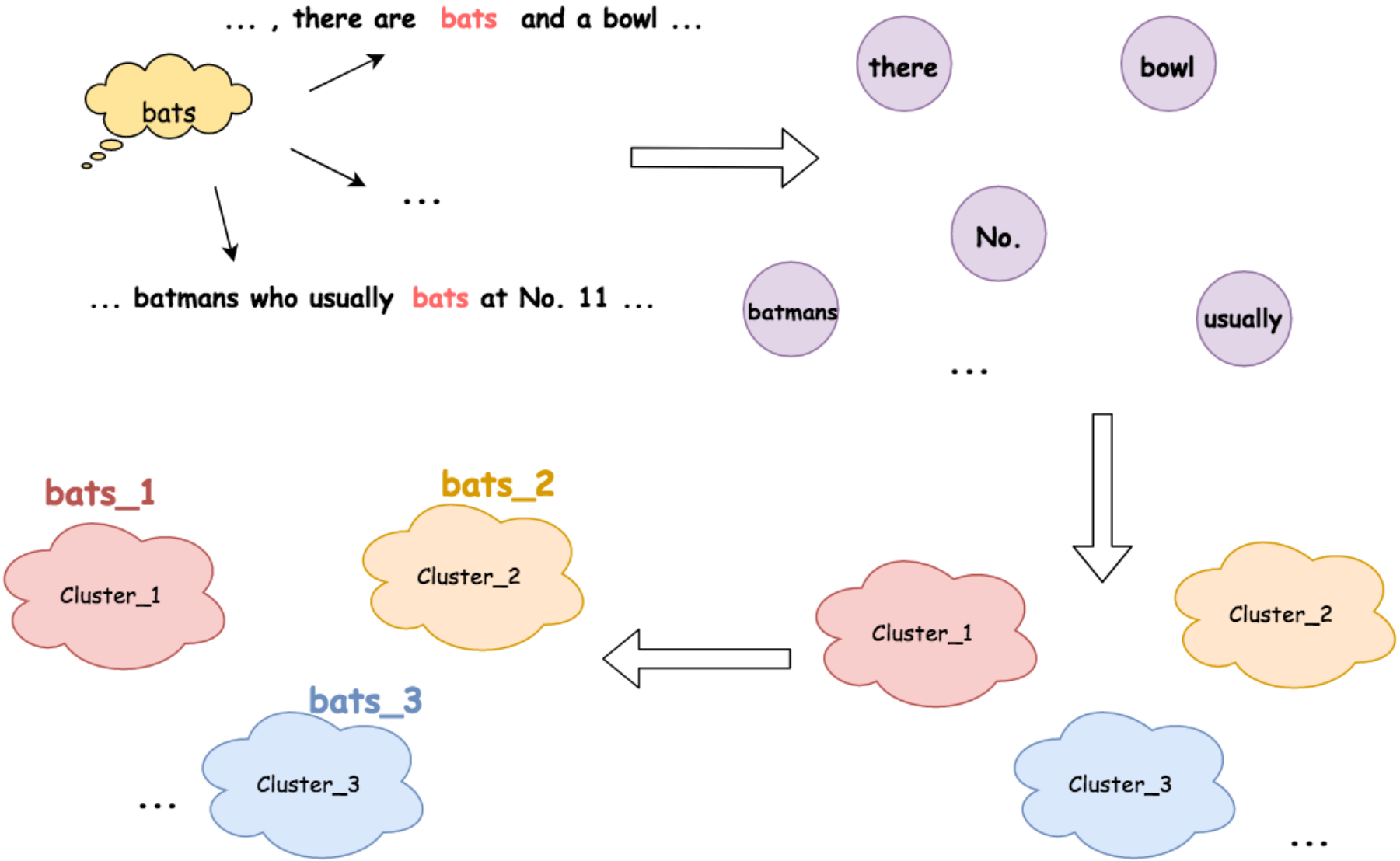
# Motivation

- Words have multiple meanings
- Single-prototype models represent only one representation for each word
  $\Rightarrow$ can not capture different meanings
- Representation of one of the meanings is influenced by all meanings of the word
- Using all contexts of a homonymous or polysemous word to build a single prototype
  $\Rightarrow$ none of the meanings is well represented

$\longrightarrow$ Multi-prototype model uses multiple representations to capture different senses and usages of a word. *(Reisinger and Mooney, 2010b)*

$\longrightarrow$ **Idea**: Learned single-prototype embeddings to represent each context window, then clustered to perform word sense discrimination *(Schütze, 1998)*

# Learning multiple prototypes



# Words similarity in multi-prototype model *(Reisinger and Mooney, 2010b)*

« AvgSimC corresponds to *soft cluster assignment*, weighting each similarity term in AvgSim by the likelihood of the word contexts appearing in their respective clusters»

$$AvgSimC(w, w') = \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K} p(c, w, i)\, p(c', w', j)\, d(\mu_i(w), \mu_j(w')) \tag{8}$$

- $K$: number of clusters
- $p(c, w, i)$: likelihood that word $w$ is in its cluster $i$ given context $c$
- $\mu_i(w)$: vector representing $i$-th cluster centroid of $w$
- $d(v, v')$: similarity between two vectors, any of the distance functions *(Curran 2004)*

$\Rightarrow$ Can be computed with or without context
(assuming uniform $p(c, w, i)$ over $i \longrightarrow AvgSim$)

# Setup

- Corpus: April 2010 snapshot of *Wikipedia* corpus *(Shaoul and Westbury, 2010)*, total 2 million articles and 990 million tokens
- Dictionary of 30,000 most frequent words in Wikipedia in lower case
- Preprocessing:
  - Map rare words not found in dictionary to an UNKNOWN token
- Hyperparameters:
  - 10-words window size of local context
  - 10 prototypes (for multi-prototype variants, $K = 10$)

# Qualitative Evaluations

| Center Word | C&W | Our Model |
|---|---|---|
| markets | firms, industries, stores | market, firms, businesses |
| American | Australian, Indian, Italian | U.S., Canadian, African |
| illegal | alleged, overseas, banned | harmful, prohibited, convicted |

Table 1: Nearest neighbors of words based on cosine similarity (C&W model - Model using single prototype approach)

Compared with results of *C&W* model *(Collobert and Weston, 2008)*

- Less constrained by syntax, singular and plural forms of a word are similar in meaning
- More semantic

# Qualitative Evaluations

| Center Word | Nearest Neighbors |
|---|---|
| bank_1 | corporation, insurance, company |
| bank_2 | shore, coast, direction |
| star_1 | movie, film, radio |
| star_2 | galaxy, planet, moon |
| cell_1 | telephone, smart, phone |
| cell_2 | pathology, molecular, physiology |
| left_1 | close, leave, live |
| left_2 | top, round, right |

Table 2: Nearest neighbors of words based learned by the model using the multi-prototype approach based on cosine similarity

The clustering can find different *meanings*, *usages* and *parts of speech* of the words

$\Rightarrow$ can group different contexts of a word in different groups

# WordSim-353 *(Finkelstein et al., 2001)*

| Model | Corpus | $\rho \times 100$ |
|---|---|---|
| Our Model-g | Wiki. | 22.8 |
| C&W | RCV1 | 29.5 |
| HLBL | RCV1 | 33.2 |
| C&W* | Wiki. | 49.8 |
| C&W | Wiki. | 55.3 |
| Our Model | Wiki. | 64.2 |
| Our Model* | Wiki. | 71.3 |
| Pruned *tf-idf* | Wiki. | 73.4 |
| ESA | Wiki. | 75 |
| Tiered Pruned *tf-idf* | Wiki. | 76.9 |

Table 3: Spearman's $\rho$ correlation on WordSim-353

# WordSim-353 *(Finkelstein et al., 2001)*

| Model | Corpus | $\rho \times 100$ |
|---|---|---|
| Our Model-g | Wiki. | 22.8 |
| C&W | RCV1 | 29.5 |
| HLBL | RCV1 | 33.2 |
| C&W* | Wiki. | 49.8 |
| C&W | Wiki. | 55.3 |
| Our Model | Wiki. | 64.2 |
| Our Model* | Wiki. | 71.3 |
| Pruned *tf-idf* | Wiki. | 73.4 |
| ESA | Wiki. | 75 |
| Tiered Pruned *tf-idf* | Wiki. | 76.9 |

Table 3: Spearman's $\rho$ correlation on WordSim-353

# WordSim-353 *(Finkelstein et al., 2001)*

| Model | Corpus | $\rho \times 100$ |
|---|---|---|
| Our Model-g | Wiki. | 22.8 |
| C&W | RCV1 | 29.5 |
| HLBL | RCV1 | 33.2 |
| C&W* | Wiki. | 49.8 |
| C&W | Wiki. | 55.3 |
| Our Model | Wiki. | 64.2 |
| Our Model* | Wiki. | 71.3 |
| Pruned *tf-idf* | Wiki. | 73.4 |
| ESA | Wiki. | 75 |
| Tiered Pruned *tf-idf* | Wiki. | 76.9 |

Table 3: Spearman's $\rho$ correlation on WordSim-353

# WordSim-353 *(Finkelstein et al., 2001)*

| Model | Corpus | $\rho \times 100$ |
|---|---|---|
| Our Model-g | Wiki. | 22.8 |
| C&W | RCV1 | 29.5 |
| HLBL | RCV1 | 33.2 |
| C&W* | Wiki. | 49.8 |
| C&W | Wiki. | 55.3 |
| Our Model | Wiki. | 64.2 |
| Our Model* | Wiki. | 71.3 |
| Pruned *tf-idf* | Wiki. | 73.4 |
| ESA | Wiki. | 75 |
| Tiered Pruned *tf-idf* | Wiki. | 76.9 |

Table 3: Spearman's $\rho$ correlation on WordSim-353

# WordSim-353 *(Finkelstein et al., 2001)*

- Higher correlation (**64.2**) than using either local (*C&W*: **55.3**) or global context (*Our Model-g*: **22.8**) alone
- Removing stop words improved correlation (*Our Model**: **71.3** $>$ **64.2** )
- Still lower than *state-of-the-art*-results (**71.3** $<$ **73.4** $<$ **75** $<$ **76.9**)

# Stanford Contextual Word Similarity (SCWS)

Problems with WordSim-353?

# Stanford Contextual Word Similarity (SCWS)

Problems with WordSim-353?

- tiger - tiger has a similarity of 10
- similarity has been assigned in isolation
- small dataset, only nouns, ...

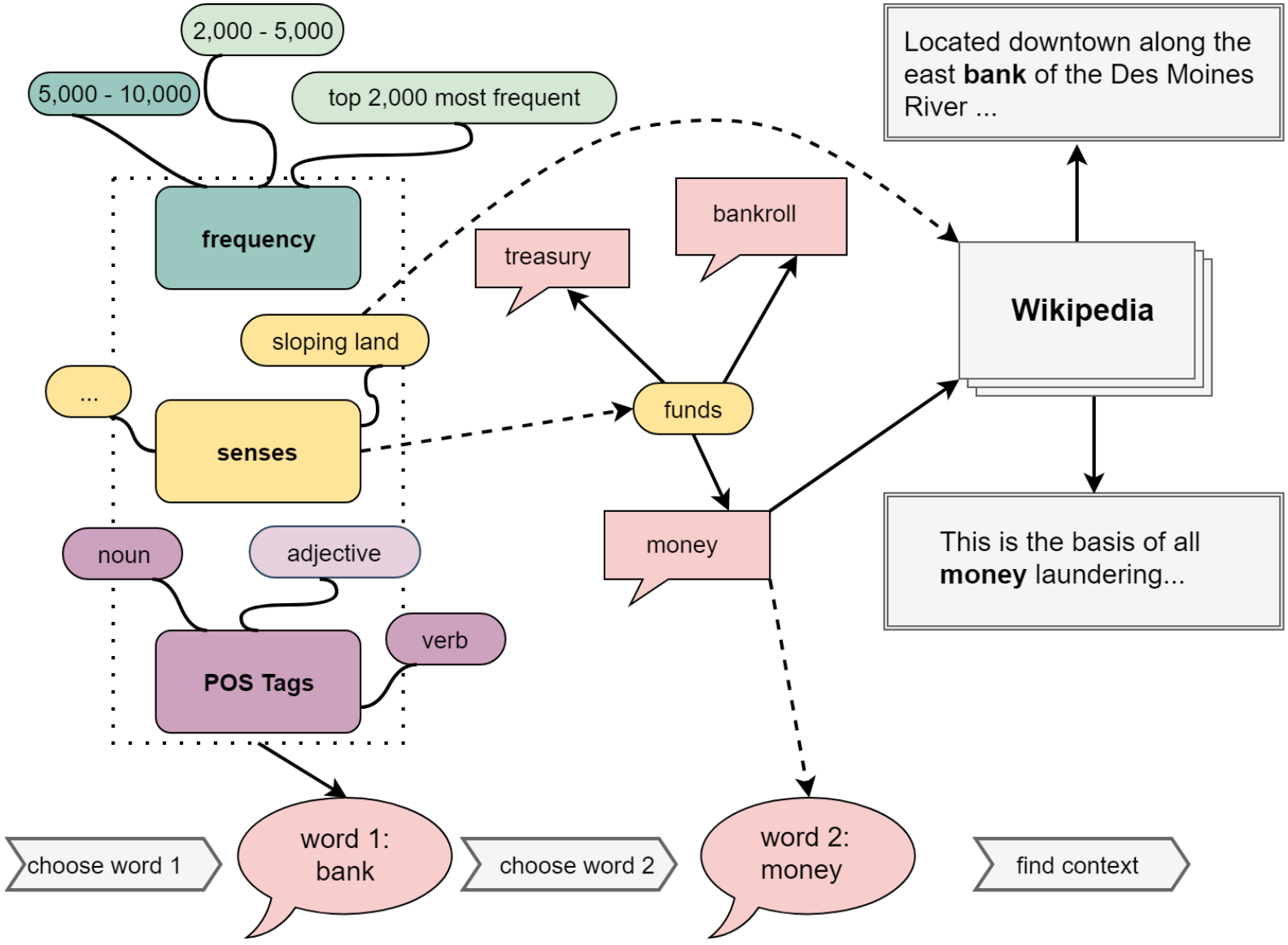
# Stanford Contextual Word Similarity (SCWS)

Problems with WordSim-353?

- tiger - tiger has a similarity of 10
- similarity has been assigned in isolation
- small dataset, only nouns, ...

| Word 1 | Word 2 |
|---|---|
| Located downtown along the east **bank** of the Des Moines River ... | This is the basis of all **money** laundering ... |
| ... and Andy 's getting ready to **pack** his bags and head up to Los Angeles ... | ... defends the house against another **pack** of zombies ... |

# Dataset Construction



# Evaluation on SCWS

| Model | $\rho \times 100$ |
|---|---|
| C&W-S | 57.0 |
| Our Model-S | 58.6 |
| Our Model-M AvgSim | 62.8 |
| Our Model-M AvgSimC | **65.7** |
| *tf-idf*-S | 26.3 |
| Pruned *tf-idf*-S | 62.5 |
| Pruned *tf-idf*-M AvgSim | 60.4 |
| Pruned *tf-idf*-M AvgSimC | 60.5 |

Table 4: Spearman's $\rho$ correlation on new dataset.

AvgSim: similarity from each prototype equally considered
AvgSimC: weighted similarity based on context

# Evaluation of model

- Outperformed C&W's model and *state-of-the-art*-results
- Multi-prototype model improved performance without using context
- context improves performance even further
- lower scores overall (task is harder)

# Evaluation on real-world NLU tasks

- 'Do Multi-Sense Embeddings Improve Natural Language Understanding?' (2015)
- Li & Jurafsky compare performance of multi-sense vectors on different tasks
- use their own model + SkipGram vectors as baseline
- we take a look at general trends and findings

# Results - excerpt

| | SkipGram 50d | L&J 50d | SG 100d | L&J + global context 100d | SG 300d |
|---|---|---|---|---|---|
| Named Entity Recognition (CoNLL-2003) | 0.852 | 0.854 | 0.867 | 0.871 | 0.882 |
| POS-Tagging (WSJ) | 0.925 | 0.938 | 0.940 | 0.952 | 0.954 |
| Sentence-level Sentiment Classification(IMDb, Pang et. al) | 0.750 | 0.750 | 0.768 | 0.763 | 0.774 |
| Semantic Relationship Classification (SemEval-2010) | 0.748 | 0.762 | 0.770 | 0.778 | 0.798 |
| Sentence Semantic Relatedness (SICK, LSTM Model) | 0.843 | 0.846 | 0.850 | 0.854 | 0.850 |

Accuracy as reported for different tasks in Li & Jurafsky. L&J denotes Li & Jurafsky's Expectation model, L&J + global context is the Expectation model that also takes global context into account.

# Results - excerpt

| | SkipGram 50d | L&J 50d | SG 100d | L&J + global context 100d | SG 300d |
|---|---|---|---|---|---|
| Named Entity Recognition (CoNLL-2003) | 0.852 | 0.854 | 0.867 | 0.871 | 0.882 |
| POS-Tagging (WSJ) | 0.925 | 0.938 | 0.940 | 0.952 | 0.954 |
| Sentence-level Sentiment Classification(IMDb, Pang et. al) | 0.750 | 0.750 | 0.768 | 0.763 | 0.774 |
| Semantic Relationship Classification (SemEval-2010) | 0.748 | 0.762 | 0.770 | 0.778 | 0.798 |
| Sentence Semantic Relatedness (SICK, LSTM Model) | 0.843 | 0.846 | 0.850 | 0.854 | 0.850 |

Accuracy as reported for different tasks in Li & Jurafsky. L&J denotes Li & Jurafsky's Expectation model, L&J + global context is the Expectation model that also takes global context into account.

# Results - excerpt

| | SkipGram 50d | L&J 50d | SG 100d | L&J + global context 100d | SG 300d |
|---|---|---|---|---|---|
| Named Entity Recognition (CoNLL-2003) | 0.852 | 0.854 | 0.867 | 0.871 | 0.882 |
| POS-Tagging (WSJ) | 0.925 | 0.938 | 0.940 | 0.952 | 0.954 |
| Sentence-level Sentiment Classification(IMDb, Pang et. al) | 0.750 | 0.750 | 0.768 | 0.763 | 0.774 |
| Semantic Relationship Classification (SemEval-2010) | 0.748 | 0.762 | 0.770 | 0.778 | 0.798 |
| Sentence Semantic Relatedness (SICK, LSTM Model) | 0.843 | 0.846 | 0.850 | 0.854 | 0.850 |

Accuracy as reported for different tasks in Li & Jurafsky. L&J denotes Li & Jurafsky's Expectation model, L&J + global context is the Expectation model that also takes global context into account.

# Results - excerpt

| | SkipGram 50d | L&J 50d | SG 100d | L&J + global context 100d | SG 300d |
|---|---|---|---|---|---|
| Named Entity Recognition (CoNLL-2003) | 0.852 | 0.854 | 0.867 | 0.871 | 0.882 |
| POS-Tagging (WSJ) | 0.925 | 0.938 | 0.940 | 0.952 | 0.954 |
| Sentence-level Sentiment Classification(IMDb, Pang et. al) | 0.750 | 0.750 | 0.768 | 0.763 | 0.774 |
| Semantic Relationship Classification (SemEval-2010) | 0.748 | 0.762 | 0.770 | 0.778 | 0.798 |
| Sentence Semantic Relatedness (SICK, LSTM Model) | 0.843 | 0.846 | 0.850 | 0.854 | 0.850 |

Accuracy as reported for different tasks in Li & Jurafsky. L&J denotes Li & Jurafsky's Expectation model, L&J + global context is the Expectation model that also takes global context into account.

# Results - excerpt

| | SkipGram 50d | L&J 50d | SG 100d | L&J + global context 100d | SG 300d |
|---|---|---|---|---|---|
| Named Entity Recognition (CoNLL-2003) | 0.852 | 0.854 | 0.867 | 0.871 | 0.882 |
| POS-Tagging (WSJ) | 0.925 | 0.938 | 0.940 | 0.952 | 0.954 |
| Sentence-level Sentiment Classification(IMDb, Pang et. al) | 0.750 | 0.750 | 0.768 | 0.763 | 0.774 |
| Semantic Relationship Classification (SemEval-2010) | 0.748 | 0.762 | 0.770 | 0.778 | 0.798 |
| Sentence Semantic Relatedness (SICK, LSTM Model) | 0.843 | 0.846 | 0.850 | 0.854 | 0.850 |

Accuracy as reported for different tasks in Li & Jurafsky. L&J denotes Li & Jurafsky's Expectation model, L&J + global context is the Expectation model that also takes global context into account.

# Results - excerpt

| | SkipGram 50d | L&J 50d | SG 100d | L&J + global context 100d | SG 300d |
|---|---|---|---|---|---|
| Named Entity Recognition (CoNLL-2003) | 0.852 | 0.854 | 0.867 | 0.871 | 0.882 |
| POS-Tagging (WSJ) | 0.925 | 0.938 | 0.940 | 0.952 | 0.954 |
| Sentence-level Sentiment Classification(IMDb, Pang et. al) | 0.750 | 0.750 | 0.768 | 0.763 | 0.774 |
| Semantic Relationship Classification (SemEval-2010) | 0.748 | 0.762 | 0.770 | 0.778 | 0.798 |
| Sentence Semantic Relatedness (SICK, LSTM Model) | 0.843 | 0.846 | 0.850 | 0.854 | 0.850 |

Accuracy as reported for different tasks in Li & Jurafsky. L&J denotes Li & Jurafsky's Expectation model, L&J + global context is the Expectation model that also takes global context into account.

# Trends

- global context always outperforms 50d SkipGram vectors
- ... however 100d vectors always outperform 50d vectors
- increasing dimensionality often equivalent to training with complex model
- many tasks do not profit from word sense disambiguation
  - Named Entity Recognition
  - Sentiment Analysis
- POS-Tagging and Semantic Relationship Classification improved (kind of)
- when using state-of-the-art models: no impact

# Conclusion

- sense disambiguation only helpful in very specific tasks
- similar or better results are often achieved with better models and/or higher dimensionality
- more sophisticated models have knowledge about multiple word senses (cf. lecture 'Uncovering information')

# Discussion

- questions?
- expectation vs reality?

# References I



Collobert et. al (2011)

Natural Language Processing (Almost) from Scratch


Collobert & Weston (2008)

A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning


Curran (2004)

From Distributional to Semantic Similarity. Technical Report


Dhillon & Modha (2001)

Concept Decompositions for Large Sparse Text Data using Clustering


Finkelstein et al. (2001)

Placing Search in Context: The Concept Revisited

# References II


Hendrickx et. al (2010)

SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals

Huang et. al (2012)

Improving Word Representations via Global Context and Multiple Word Prototypes

Li & Jurafsky (2015)

Do Multi-Sense Embeddings Improve Natural Language Understanding?

Pang et. al (2002)

Thumbs up? Sentiment Classification using Machine Learning Techniques

Reisinger & Mooney (2010b)

Multi-Prototype Vector-Space Models of Word Meaning

# References III


Shaoul and Westbury (2010)
The Wesbury Lab Wikipedia corpus


Schütze (1998)
Automatic Word Sense Discrimination