CS 6770 Natural Language Processing

Word Embeddings: Skip-gram and GloVe

Yangfeng Ji

Information and Language Processing Lab
Department of Computer Science
University of Virginia

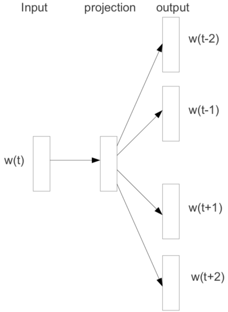
Overview

1. The Skip-gram Model
2. GloVe: Global Vectors for Word Representation
3. Further Discussion

The Skip-gram Model

The Skip-gram Model

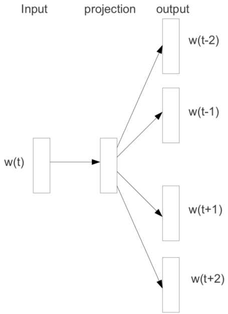
Instead of using matrix decomposition, a different strategy of learning word embeddings is using a word w_t to predict its surrounding words w_{t+i}



[Mikolov et al., 2013]

The Skip-gram Model

Instead of using matrix decomposition, a different strategy of learning word embeddings is using a word w_t to predict its surrounding words w_{t+i}



In probabilistic form, we need

$$P(w_{t+i} \mid w_t) = ? \tag{1}$$

[Mikolov et al., 2013]

Skip-gram

One way of finding a better word representation is to make sure it has the potential to predict its surrounding words

$$P(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})} \tag{2}$$

where $i \in \{-c, \ldots, -1, 1, \ldots, c\}$ and c is the window size.

Skip-gram

One way of finding a better word representation is to make sure it has the potential to predict its surrounding words

$$P(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{u}_{w_{t+i}}^\top \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^\top \boldsymbol{v}_{w_t})} \tag{2}$$

where $i \in \{-c, \ldots, -1, 1, \ldots, c\}$ and c is the window size.

- $t = 6, c = 2$

Skip-gram

One way of finding a better word representation is to make sure it has the potential to predict its surrounding words

$$P(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})} \tag{2}$$

where $i \in \{-c, \ldots, -1, 1, \ldots, c\}$ and c is the window size.

- $t = 6, c = 2$
- Usually, larger window size c gives better quality of word representations, but it also causes large computational complexity.

Skip-gram

One way of finding a better word representation is to make sure it has the potential to predict its surrounding words

$$P(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})} \tag{2}$$

where $i \in \{-c, \ldots, -1, 1, \ldots, c\}$ and c is the window size.

- $t = 6, c = 2$
- Usually, larger window size c gives better quality of word representations, but it also causes large computational complexity.
- Unlike LSA, the skip-gram model always considers local context.

Word Vectors vs. Context Vectors

Distinguish a word as target (input) and context (output):

$$p(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})} \tag{3}$$

The definition in equation 3 requires two sets of parameters for the same vocabulary

- $\boldsymbol{v}_w$: word vector (as input)
- $\boldsymbol{u}_w$: context vector (as output)

Word Vectors vs. Context Vectors

Distinguish a word as target (input) and context (output):

$$p(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})} \quad (3)$$

The definition in equation 3 requires two sets of parameters for the same vocabulary

- $\boldsymbol{v}_w$: word vector (as input)
- $\boldsymbol{u}_w$: context vector (as output)

Quiz

Why we need two vectors for a word?

Word Vectors vs. Context Vectors

Distinguish a word as target (input) and context (output):

$$p(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})} \tag{3}$$

The definition in equation 3 requires two sets of parameters for the same vocabulary

- $\boldsymbol{v}_w$: word vector (as input)
- $\boldsymbol{u}_w$: context vector (as output)

Quiz

Why we need two vectors for a word? Assume we only use one set of the parameter $\{\boldsymbol{v}_w\}$

$$p(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{v}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{v}_{w'}^{\top} \boldsymbol{v}_{w_t})} \tag{4}$$

Word Vectors vs. Context Vectors

Distinguish a word as target (input) and context (output):

$$p(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})} \tag{3}$$

The definition in equation 3 requires two sets of parameters for the same vocabulary

- $\boldsymbol{v}_w$: word vector (as input)
- $\boldsymbol{u}_w$: context vector (as output)

Quiz

Why we need two vectors for a word? Assume we only use one set of the parameter $\{\boldsymbol{v}_w\}$

$$p(w_{t+i} \mid w_t; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{v}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{v}_{w'}^{\top} \boldsymbol{v}_{w_t})} \tag{4}$$

A trivial solution that maximize the (log-)probability is $\boldsymbol{v}_{w_{t+i}} = \boldsymbol{v}_w$, which means all words will have the exactly same embedding.

Objective Function

The objective function of a skip-gram model is defined as

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq i\leq c; i\neq 0}\log p(w_{t+i} \mid w_t) \tag{5}$$

Objective Function

The objective function of a skip-gram model is defined as

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le i\le c; i\neq 0}\log p(w_{t+i}\mid w_t) \tag{5}$$

Each log probability is defined as

$$\log p(w_{t+i}\mid w_t) \quad = \quad \log\frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top}\boldsymbol{v}_{w_t})}{\sum_{w'\in\mathcal{V}}\exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_t})}$$

Objective Function

The objective function of a skip-gram model is defined as

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq i\leq c;i\neq 0}\log p(w_{t+i}\mid w_t) \tag{5}$$

Each log probability is defined as

$$\begin{aligned}\log p(w_{t+i}\mid w_t) &= \log\frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top}\boldsymbol{v}_{w_t})}{\sum_{w'\in\mathcal{V}}\exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_t})}\\ &= \boldsymbol{u}_{w_{t+i}}^{\top}\boldsymbol{v}_{w_t}-\log\sum_{w'\in\mathcal{V}}\exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_t})\end{aligned}$$

Objective Function

The objective function of a skip-gram model is defined as

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq i\leq c; i\neq 0}\log p(w_{t+i}\mid w_t) \tag{5}$$

Each log probability is defined as

$$\begin{aligned}\log p(w_{t+i}\mid w_t) &= \log\frac{\exp(\boldsymbol{u}_{w_{t+i}}^{\top}\boldsymbol{v}_{w_t})}{\sum_{w'\in\mathcal{V}}\exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_t})}\\ &= \boldsymbol{u}_{w_{t+i}}^{\top}\boldsymbol{v}_{w_t} - \log\sum_{w'\in\mathcal{V}}\exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_t})\end{aligned}$$

Essentially, this is learning a classifier over a huge number of classes. In practice, the vocab size could be 10K, 50K or even bigger, the normalization of prediction probability is the major bottleneck.

Negative Sampling

Review what have discussed so far

- The ultimate goal is learning word representations instead of a classifier
- The normalization of prediction probability is computationally expensive

Negative Sampling

Review what have discussed so far

- The ultimate goal is learning word representations instead of a classifier
- The normalization of prediction probability is computationally expensive

To reduce the computational complexity, we can replace

$$\log p(w_{t+i} \mid w_t) = \boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t} - \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})$$

with the following function as objective

$$\log \sigma(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t}) - \sum_{i=1}^{k} \log \sigma(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})\Big|_{w' \sim p_n(w)} \tag{6}$$

where k is the number of negative samples and $\sigma(\cdot)$ is the Sigmoid function (the one used for binary classification in lecture 02)

Basic Training Procedure

Example with $t = 6$, $i = 1$, and $k = 3$

```
... finding a better word representation ...
```

w_6	w_7	negative samples
better	word	larger
		cause
		window

Basic Training Procedure

Example with $t = 6$, $i = 1$, and $k = 3$

```
... finding a better word representation ...
```

w_6	w_7	negative samples
better	word	larger
		cause
		window

For a given word w_t and i

1. Treat its neighboring context word w_{t+i} as positive example
2. Randomly sample k other words from the vocab as negative examples
3. Optimize Equation 6 to update both $\boldsymbol{v}_{\cdot}$ and $\boldsymbol{u}_{\cdot}$

Two Factors in Negative Sampling

There are two factors that can affect the model performance [Mikolov et al., 2013]

$$\log \sigma(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t}) - \sum_{i=1}^{k} \log \sigma(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})\Big|_{w' \sim p_n(w)} \tag{7}$$

Two Factors in Negative Sampling

There are two factors that can affect the model performance [Mikolov et al., 2013]

$$\log \sigma(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t}) - \sum_{i=1}^{k} \log \sigma(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})\Big|_{w' \sim p_n(w)} \tag{7}$$

- The size of negative samples k
  - $5 \leq k \leq 20$ works better for small datasets
  - $2 \leq k \leq 5$ is enough for large datasets

Two Factors in Negative Sampling

There are two factors that can affect the model performance [Mikolov et al., 2013]

$$\log \sigma(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t}) - \sum_{i=1}^{k} \log \sigma(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t})\Big|_{w' \sim p_n(w)} \tag{7}$$

- The size of negative samples k
  - $5 \leq k \leq 20$ works better for small datasets
  - $2 \leq k \leq 5$ is enough for large datasets
- Noisy distribution $p_n(w)$
  - $p_n(w) \propto \text{unigram-distribution}(w)^{\frac{3}{4}}$

Examples

- Context window size: 3
- Word embedding dimension: 50
- Epochs of training: 3

natural	embeddings
processing	contextualized
nlp	embedding
nl	representations
language	vectors
understanding	elmo
nlu	static
nlg	word
fundamental	polyglot

Online Demo

GloVe: Global Vectors for Word Representation

Glove

The motivation of GloVe [Pennington et al., 2014] is to find a balance between the methods based on

- global matrix factorization (e.g., LSA) and
- local context windows (e.g., Skip-gram).

Word-to-word Co-occurrence Matrix

- Define $\mathbf{X}$ with $X_{i,j}$ denotes the frequency of word j appears in the context of word i

$$\mathbf{X} = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ X_{i,1} & \cdots & X_{i,j-1} & X_{i,j} & X_{i,j+1} & \cdots & X_{i,V} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \tag{8}$$

Each row corresponds one target word, each column corresponds one context word.

Word-to-word Co-occurrence Matrix

- Define $\mathbf{X}$ with $X_{i,j}$ denotes the frequency of word j appears in the context of word i

$$\mathbf{X} = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ X_{i,1} & \dots & X_{i,j-1} & X_{i,j} & X_{i,j+1} & \dots & X_{i,V} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{8}$$

Each row corresponds one target word, each column corresponds one context word.

- Empirical probability estimation of w_j given w_i

$$Q(w_j \mid w_i) = \frac{X_{ij}}{X_i} \tag{9}$$

where $X_i = \sum_j X_{i,j}$

Probability Estimation via Word Embeddings

Another way to estimate the probability of w_j given w_i is

$$P(w_j \mid w_i) = \frac{\exp(\boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^\top \boldsymbol{v}_{w_i})} \tag{10}$$

with $\boldsymbol{u}_\cdot$ and $\boldsymbol{v}_\cdot$ are two sets of parameters (embeddings) associated with words, similar to the Skip-gram model.

GloVe

The basic idea is to learn $\{\boldsymbol{v}_\cdot\}$ and $\{\boldsymbol{u}_\cdot\}$, such that

$$Q(w_j \mid w_i) \approx P(w_j \mid w_i) \tag{11}$$

or

$$\log Q(w_j \mid w_i) \approx \log P(w_j \mid w_i) \tag{12}$$

GloVe

The basic idea is to learn $\{\boldsymbol{v}_\cdot\}$ and $\{\boldsymbol{u}_\cdot\}$, such that

$$Q(w_j \mid w_i) \approx P(w_j \mid w_i) \tag{11}$$

or

$$\log Q(w_j \mid w_i) \approx \log P(w_j \mid w_i) \tag{12}$$

More specific

$$\log(X_{ij}) - \log(X_i) \approx \boldsymbol{u}_{w_j}^{\top} \boldsymbol{v}_{w_i} - \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_i}) \tag{13}$$

GloVe (II)

Starting point:

$$\log(X_{ij}) - \log(X_i) \approx \boldsymbol{u}_{w_j}^{\top}\boldsymbol{v}_{w_i} - \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_i}) \tag{14}$$

GloVe (II)

Starting point:

$$\log(X_{ij}) - \log(X_i) \approx \boldsymbol{u}_{w_j}^{\top}\boldsymbol{v}_{w_i} - \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_i}) \tag{14}$$

In order to find the best approximation, we could formulate this as a optimization problem

$$\left\{ \log(X_{ij}) - \log(X_i) - \boldsymbol{u}_{w_j}^{\top}\boldsymbol{v}_{w_i} + \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_i}) \right\}^2 \tag{15}$$

GloVe (II)

Starting point:

$$\log(X_{ij}) - \log(X_i) \approx \boldsymbol{u}_{w_j}^{\mathsf{T}} \boldsymbol{v}_{w_i} - \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\mathsf{T}} \boldsymbol{v}_{w_i}) \tag{14}$$

In order to find the best approximation, we could formulate this as a optimization problem

$$\left\{ \log(X_{ij}) - \log(X_i) - \boldsymbol{u}_{w_j}^{\mathsf{T}} \boldsymbol{v}_{w_i} + \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\mathsf{T}} \boldsymbol{v}_{w_i}) \right\}^2 \tag{15}$$

If we only consider the unnormalized version of P and Q, it can be further simplified as (Eq. 16 in [Pennington et al., 2014])

$$\left\{ \log(X_{ij}) - \boldsymbol{u}_{w_j}^{\mathsf{T}} \boldsymbol{v}_{w_i} \right\}^2 \tag{16}$$

Objective Function

The overall objective function is defined as

$$\sum_{w_i} \sum_{w_j} (\log(X_{ij}) - \boldsymbol{u}_{w_j}^{\mathsf{T}} \boldsymbol{v}_{w_i})^2 \tag{17}$$

Objective Function

The overall objective function is defined as

$$\sum_{w_i} \sum_{w_j} (\log(X_{ij}) - \boldsymbol{u}_{w_j}^{\top} \boldsymbol{v}_{w_i})^2 \tag{17}$$

The objective function is further refined by discouraging high-frequency words as

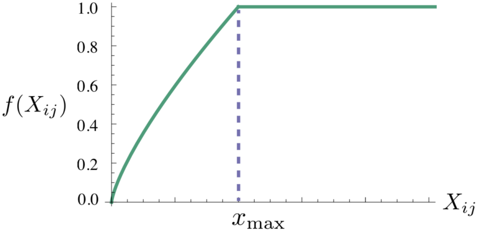
$$\sum_{w_i} \sum_{w_j} f(X_{ij})(\log(X_{ij}) - \boldsymbol{u}_{w_j}^{\top} \boldsymbol{v}_{w_i})^2 \tag{18}$$

Down-weighting

Weighting function:

$$f(x) = \begin{cases} \left(\frac{x}{x_{\max}}\right)^a & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases} \quad (19)$$

where $a = 3/4$.



Further Discussion

Gender Bias

$$
\begin{aligned}
v_{\texttt{man}} - v_{\texttt{woman}} &\approx v_{\texttt{computer programmer}} - v_{\texttt{homemaker}} && (20) \\
v_{\texttt{father}} - v_{\texttt{mother}} &\approx v_{\texttt{doctor}} - v_{\texttt{nurse}} && (21)
\end{aligned}
$$

[Bolukbasi et al., 2016]

Example

[Bolukbasi et al., 2016]

Problem

Word embeddings like this not only reflect such stereotypes but also amplify them

A Solution

Three steps [Bolukbasi et al., 2016]

1. find gender neutral words with biases in the original embeddings;
2. identify the gender-specific space V and its orthogonal complement $V^{\perp}$
3. project embeddings of the gender neutral words to the subspace $V^{\perp}$

Question

Can we have an interpretability of each dimension?

Solution: post-processing on word embeddings

- reconstructing with sparsity constraint [Faruqui et al., 2015]
- rotating word embedding space using factor analysis [Park et al., 2017]

Reconstruction with Sparsity

Interpretability is *derived* from the sparsity constraint as

$$\underset{\mathbf{D},\mathbf{A}}{\operatorname{argmin}} \sum_{i=1}^{V} \|\boldsymbol{x}_i - \mathbf{D}\boldsymbol{a}_i\|_2^2 + \lambda \|\boldsymbol{a}_i\|_1 + \tau \|\mathbf{D}\|_2^2 \tag{22}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{a}_i$ are the original and sparse embeddings of word i, $\mathbf{D}$ is the transformation matrix.

Example

X	combat, guard, honor, bow, trim, naval 'll, could, faced, lacking, seriously, scored see, n't, recommended, depending, part due, positive, equal, focus, respect, better sergeant, comments, critics, she, videos
A	fracture, breathing, wound, tissue, relief relationships, connections, identity, relations files, bills, titles, collections, poems, songs naval, industrial, technological, marine stadium, belt, championship, toll, ride, coach

Figure: Top-ranked words per-dimension before and after reconstruction. Each line shows words from a different dimension.

Problem

- Word embeddings from either Word2vec or GloVe encode not just semantic information

Problem

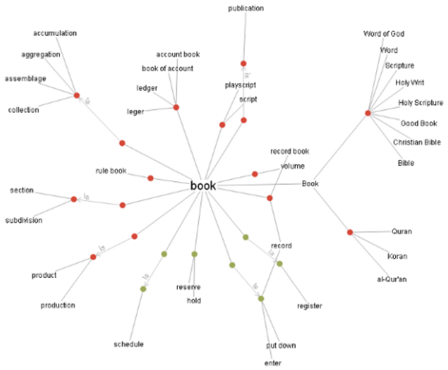
- Word embeddings from either Word2vec or GloVe encode not just semantic information
- In some applications, we want to emphasize one particular aspect of linguistic information
  - Semantic information [Faruqui et al., 2014, Mrksic et al., 2016]
  - Discourse information [Ji and Eisenstein, 2014]

Problem

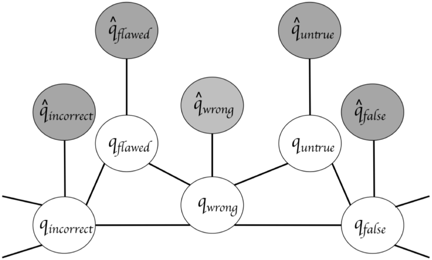
- Word embeddings from either Word2vec or GloVe encode not just semantic information
- In some applications, we want to emphasize one particular aspect of linguistic information
  - Semantic information [Faruqui et al., 2014, Mrksic et al., 2016]
  - Discourse information [Ji and Eisenstein, 2014]
- Solutions
  - fine-tuning word embeddings with certain constraints [Faruqui et al., 2014, Mrksic et al., 2016]
  - learning from supervision information [Ji and Eisenstein, 2014]

Retrofitting with WordNet [Miller, 1995]

- $\Omega = (V, E)$ be a semantic graph over words, where V is the node set with each element as a word, and E is the edge set with each edge representing a semantic relation between two words.

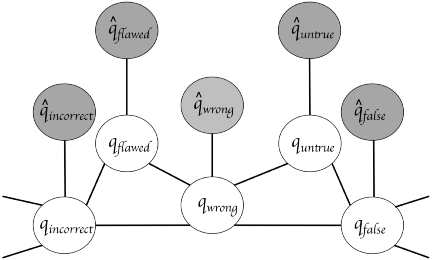


Retrofitting (II)



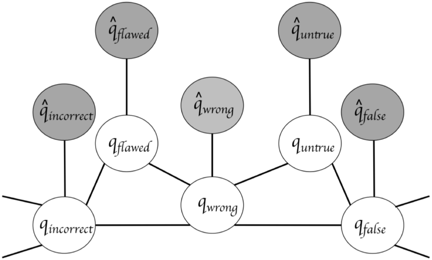
- The goal is to learn word embeddings $\{\boldsymbol{q}\}$ such that $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ are close enough if $(i, j) \in E$.

Retrofitting (II)



- The goal is to learn word embeddings $\{\boldsymbol{q}\}$ such that $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ are close enough if $(i, j) \in E$.
- In addition, $\{\boldsymbol{q}\}$ should also satisfy the constraint from original word embeddings, such that $\boldsymbol{q}_i$ and $\boldsymbol{q}_i$ are close enough for every word in $\mathcal{V}$.

Retrofitting (II)



- The goal is to learn word embeddings $\{\boldsymbol{q}\}$ such that $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ are close enough if $(i, j) \in E$.
- In addition, $\{\boldsymbol{q}\}$ should also satisfy the constraint from original word embeddings, such that $\boldsymbol{q}_i$ and $\boldsymbol{q}_i$ are close enough for every word in $\mathcal{V}$.

$$\Psi(\tilde{\mathbf{Q}}) = \sum_{i=1}^{|\mathcal{V}|} \left[\alpha_i \|\boldsymbol{q}_i - \hat{\boldsymbol{q}}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|\boldsymbol{q}_i - \boldsymbol{q}_j\|^2 \right] \tag{23}$$

Counter-fitting

Inject antonymy and synonymy constraints into word embedding space to improve the embeddings' capability for judging semantic similarity

	east	**expensive**	**British**
Before	west	pricey	American
	north	cheaper	Australian
	south	costly	Britain
	southeast	overpriced	European
	northeast	inexpensive	England
After	eastward	costly	Brits
	eastern	pricy	London
	easterly	overpriced	BBC
	-	pricey	UK
	-	afford	Britain

Table 1: Nearest neighbours for target words using GloVe vectors before and after counter-fitting

[Mrksic et al., 2016]

Learning from Supervision Signal

Word embeddings learned from unsupervised methods may not be optimal for a particular task, and may not capture the desired linguistic information.

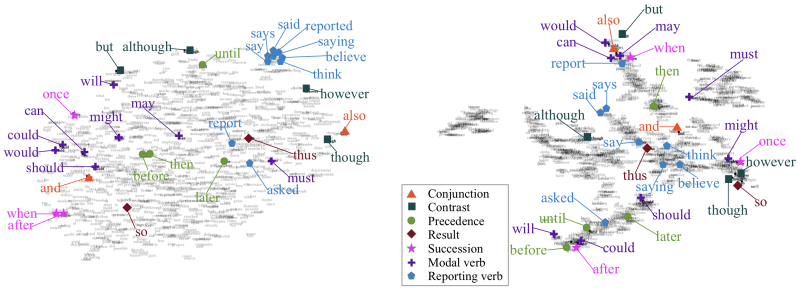


Figure: (Left) Word embeddings learned with supervision signal; (Right) Unsupervised word embeddings.

Reference

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016).
Man is to computer programmer as woman is to homemaker? debiasing word embeddings.
In *Advances in Neural Information Processing Systems*, pages 4349–4357.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2014).
Retrofitting word vectors to semantic lexicons.
arXiv preprint arXiv:1411.4166.

Faruqui, M., Tsvetkov, Y., Yogatama, D., Dyer, C., and Smith, N. (2015).
Sparse overcomplete word vector representations.
arXiv preprint arXiv:1506.02004.

Ji, Y. and Eisenstein, J. (2014).
Representation learning for text-level discourse parsing.
In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 13–24.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013).
Distributed representations of words and phrases and their compositionality.
In *Advances in neural information processing systems*, pages 3111–3119.

Miller, G. A. (1995).
Wordnet: a lexical database for english.
Communications of the ACM, 38(11):39–41.

Mrksic, N., Seaghdha, D. O., Thomson, B., Gasic, M., Rojas-Barahona, L., Su, P.-H., Vandyke, D., Wen, T.-H., and Young, S. (2016).
Counter-fitting word vectors to linguistic constraints.

Park, S., Bak, J., and Oh, A. (2017).
Rotated word vector representations and their interpretability.
In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 401–411.

Pennington, J., Socher, R., and Manning, C. (2014).
Glove: Global vectors for word representation.