CS 6501 Natural Language Processing

Word Embeddings

Yangfeng Ji

October 24, 2019

Department of Computer Science
University of Virginia

Overview

1. Word Embeddings: Skip-gram
2. Word Embedding: GloVe
3. Evaluation Methods
4. Problems

Word Embeddings: Skip-gram

Objective Function

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

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

- $\log p(w_{t+i} \mid w_t) = \boldsymbol{u}_{w_{t+i}}^{\top}\boldsymbol{v}_{w_t} - \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_t})$
- In practice, the vocab size could be 10K, 50K or even bigger, the computation of the log-sum-exp is prohibitively expensive

Negative Sampling

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

$$\log \sigma(\boldsymbol{u}_{w_{t+i}}^\top \boldsymbol{v}_{w_t}) - \sum_{i=1}^{k} E_{w' \sim p_n(w)} \Big[\log \sigma(\boldsymbol{u}_{w'}^\top \boldsymbol{v}_{w_t}) \Big] \tag{2}$$

where k is the number of negative samples

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 2 to update both $\boldsymbol{v}_{\cdot}$ and $\boldsymbol{u}_{\cdot}$

Two Factors in Negative Sampling

$$\log \sigma(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t}) - \sum_{i=1}^{k} E_{w' \sim p_n(w)} \Big[\log \sigma(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t}) \Big] \tag{3}$$

Two Factors in Negative Sampling

$$\log \sigma(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t}) - \sum_{i=1}^{k} E_{w' \sim p_n(w)} \left[\log \sigma(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t}) \right] \quad (3)$$

Two factors [Mikolov et al., 2013a]

- $k = ?$
  - $5 \leq k \leq 20$ works better for small datasets
  - $2 \leq k \leq 5$ is enough for large datasets

Two Factors in Negative Sampling

$$\log \sigma(\boldsymbol{u}_{w_{t+i}}^{\top} \boldsymbol{v}_{w_t}) - \sum_{i=1}^{k} E_{w' \sim p_n(w)} \Big[\log \sigma(\boldsymbol{u}_{w'}^{\top} \boldsymbol{v}_{w_t}) \Big] \quad (3)$$

Two factors [Mikolov et al., 2013a]

- 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: Words and their Neighbors

The same Yalp dataset, with $k = 50$

yummy	horrible
delicious	terrible
tasty	poor
delish	awful
yum	*customer*
incredible	exceptional
superb	bad
phenomenal	astonished
fantastic	*pleasant*
disappoint	*happier*
awesome	*zero*

Word Embedding: GloVe

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

$$\mathbf{X} = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ X_{i,1} & \dots & X_{i,j-1} & X_{i,j} & X_{i,j+1} & \dots & X_{i,V} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{4}$$

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

$$\mathbf{X} = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ X_{i,1} & \ldots & X_{i,j-1} & X_{i,j} & X_{i,j+1} & \ldots & X_{i,V} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \tag{4}$$

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

- Empirical probability estimation of w_j given w_i

$$Q(w_j \mid w_i) = \frac{X_{ij}}{X_i} \tag{5}$$

  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

$$P(w_j \mid w_i) = \frac{\exp(\boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i})}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^\top \boldsymbol{v}_{w_i})} \tag{6}$$

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

$$Q(w_j \mid w_i) \approx P(w_j \mid w_i) \tag{7}$$

or

$$\log Q(w_j \mid w_i) \approx \log P(w_j \mid w_i) \tag{8}$$

GloVe

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

$$Q(w_j \mid w_i) \approx P(w_j \mid w_i) \tag{7}$$

or

$$\log Q(w_j \mid w_i) \approx \log P(w_j \mid w_i) \tag{8}$$

More specific

$$\log(X_{ij}) - \log(X_i) \approx \boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i} - \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^\top \boldsymbol{v}_{w_i}) \tag{9}$$

GloVe (II)

Starting point:

$$\log(X_{ij}) - \log(X_i) \approx \boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i} - \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^\top \boldsymbol{v}_{w_i}) \tag{10}$$

GloVe (II)

Starting point:

$$\log(X_{ij}) - \log(X_i) \approx \boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i} - \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^\top \boldsymbol{v}_{w_i}) \tag{10}$$

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

$$\left\{ \log(X_{ij}) - \log(X_i) - \boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i} + \log \sum_{w' \in \mathcal{V}} \exp(\boldsymbol{u}_{w'}^\top \boldsymbol{v}_{w_i}) \right\}^2 \tag{11}$$

GloVe (II)

Starting point:

$$\log(X_{ij}) - \log(X_i) \approx \boldsymbol{u}_{w_j}^{\top}\boldsymbol{v}_{w_i} - \log\sum_{w'\in\mathcal{V}}\exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_i}) \tag{10}$$

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

$$\left\{\log(X_{ij}) - \log(X_i) - \boldsymbol{u}_{w_j}^{\top}\boldsymbol{v}_{w_i} + \log\sum_{w'\in\mathcal{V}}\exp(\boldsymbol{u}_{w'}^{\top}\boldsymbol{v}_{w_i})\right\}^2 \tag{11}$$

It can be further simplified as (Eq. 16 in [Pennington et al., 2014])

$$\left\{\log(X_{ij}) - \boldsymbol{u}_{w_j}^{\top}\boldsymbol{v}_{w_i}\right\}^2 \tag{12}$$

if we only consider the unnormalized version of P and Q.

Objective Function

The overall objective function is defined as

$$\sum_{w_i} \sum_{w_j} (\log(X_{ij}) - \boldsymbol{u}_{w_j}^{\top} \boldsymbol{v}_{w_i})^2 \tag{13}$$

Objective Function

The overall objective function is defined as

$$\sum_{w_i} \sum_{w_j} (\log(X_{ij}) - \boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i})^2 \tag{13}$$

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

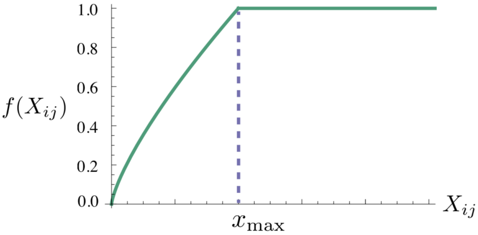
$$\sum_{w_i} \sum_{w_j} f(X_{ij})(\log(X_{ij}) - \boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i})^2 \tag{14}$$

Down-weighting

Weighting function:

$$f(x) = \begin{cases} (\frac{x}{x_{\max}})^{a} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

where $a = 3/4$.



Skip-gram as Implicit Matrix Factorization

[Levy and Goldberg, 2014] shows that skip-gram with negative sampling can be viewed as an implicit matrix factorization over a word-word co-occurrence matrix weighted by point-wise mutual information (PMI).

$$\boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i} \approx \mathrm{PMI}(w_i, w_j) - \log k \tag{16}$$

where $\mathrm{PMI}(w_i, w_j)$ is the mutual information of $P(w_i)$ and $P(w_j)$ with *a given window size* and k is the number of negative samples.

Skip-gram as Implicit Matrix Factorization (II)

The definition of $\mathrm{PMI}(w_i, w_j)$ is

$$\mathrm{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)} = \log P(w_j \mid w_i) - \log P(w_j) \qquad (17)$$

Skip-gram as Implicit Matrix Factorization (II)

The definition of $\mathrm{PMI}(w_i, w_j)$ is

$$\mathrm{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)} = \log P(w_j \mid w_i) - \log P(w_j) \tag{17}$$

Combine 16 and 17, we have

$$\begin{aligned} \boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i} &\approx \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)} - \log k \\ &= \log P(w_j \mid w_i) - \log P(w_j) - \log k \\ &= \log(X_{ij}) - \log(X_i) - \log(X_j) + \log D - \log k \end{aligned} \tag{18}$$

Similar to Eq. 8 in [Pennington et al., 2014].

Essentially,

A unified framework

$$\boldsymbol{u}_{w_j}^{\top}\boldsymbol{v}_{w_i} \approx \log(X_{ij}) + g(\mathbf{X}) \tag{19}$$

Essentially,

A unified framework

$$\boldsymbol{u}_{w_j}^\top \boldsymbol{v}_{w_i} \approx \log(X_{ij}) + g(\mathbf{X}) \tag{19}$$

Which one matters?

- $g(\mathbf{X})$, or
- Implicit/explicit optimization, or
- Other tricks (down-sampling, hyper-parameters, etc.)

Evaluation Methods

Overview

- Intrinsic Evaluation
  - Word similarity
  - Word analogy
  - Word intrusion
- Extrinsic Evaluation

Word Similarity

Let w_i and w_j be two words, and $\boldsymbol{v}_{w_i}$ and $\boldsymbol{v}_{w_j}$ be the corresponding word embeddings, word similarity can be obtained by computing their cosine similarity between $\boldsymbol{v}_{w_i}$ and $\boldsymbol{v}_{w_j}$ as

$$\cos(\boldsymbol{v}_{w_i}, \boldsymbol{v}_{w_j}) = \frac{\boldsymbol{v}_{w_i}^\top, \boldsymbol{v}_{w_j}}{\|\boldsymbol{v}_{w_i}\|_2 \cdot \|\boldsymbol{v}_{w_j}\|_2} \tag{20}$$

Examples

Word_1	Word_2	Similarity score [0,10]
love	sex	6.77
stock	jaguar	0.92
money	cash	9.15
development	issue	3.97
lad	brother	4.46

Figure: Sample word pairs along with their human similarity judgment from WS-353 [Faruqui et al., 2016].

Datasets

Available word similarity datasets

Dataset	Word pairs	Reference
RG	65	Rubenstein and Goodenough (1965)
MC	30	Miller and Charles (1991)
WS-353	353	Finkelstein et al. (2002)
YP-130	130	Yang and Powers (2006)
MTurk-287	287	Radinsky et al. (2011)
MTurk-771	771	Halawi et al. (2012)
MEN	3000	Bruni et al. (2012)
RW	2034	Luong et al. (2013)
Verb	144	Baker et al. (2014)
SimLex	999	Hill et al. (2014)

Figure: Word similarity datasets [Faruqui et al., 2016].

Word Similarity

the basis for other intrinsic evaluations

Word Analogy

- It is sometimes referred as *linguistic regularity* [Mikolov et al., 2013b]
- The basic setup

$$w_a : w_b = w_c :?$$

  where $w_{a,b,c}$ are words and w_a, w_b are related under a certain linguistic relation
- Calculation: $(\boldsymbol{v}_{w_a} - \boldsymbol{v}_{w_b})^\top (\boldsymbol{v}_{w_c} - \boldsymbol{v}_{w_d})$

Word Analogy

- It is sometimes referred as *linguistic regularity* [Mikolov et al., 2013b]
- The basic setup

$$w_a : w_b = w_c :?$$

  where $w_{a,b,c}$ are words and w_a, w_b are related under a certain linguistic relation
- Calculation: $(\boldsymbol{v}_{w_a} - \boldsymbol{v}_{w_b})^\top (\boldsymbol{v}_{w_c} - \boldsymbol{v}_{w_d})$
- Example
  - Semantic `love` : `like`
  - Syntactic `quick` : `quickly`
  - Gender `king` : `man`
  - Others `Beijing` : `China`

Word Analogy: Examples

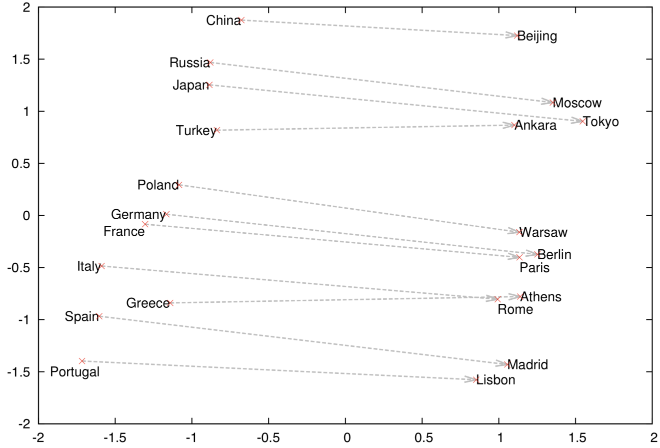


Figure: Word analogy examples.

Word intrusion

From [Faruqui et al., 2014]

`naval, industrial, technological, marine, identity`

- constructed from word embeddings
- evaluated by human annotators

Extrinsic Evaluation

- Implicit assumption: there is a consistent, global ranking of word embedding quality, and that higher quality embeddings will necessarily improve results on *any* downstream task.
- Unfortunately, this assumption does not hold in general [Schnabel et al., 2015].
- Examples
  - empirical results show that it may not be able give much help to syntactic parsing [Andreas and Klein, 2014]
  - adding surface-form features always help ([Ji and Eisenstein, 2014a] and many other works)

Problems

Gender Bias

$$v_{\texttt{man}} - v_{\texttt{woman}} \approx v_{\texttt{computer programmer}} - v_{\texttt{homemaker}} \quad (21)$$

$$v_{\texttt{father}} - v_{\texttt{mother}} \approx v_{\texttt{doctor}} - v_{\texttt{nurse}} \quad (22)$$

[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 $V^{\perp}$

Example

tote
reading
records
clip
commit
game
browsing
sites
seconds
slow
arrival
tactical
crafts
credits
trimester
tanning
user
parts
drop reel
firepower
ultrasound
busy
hoped
command
housing
caused
ill
scrimmage
modeling
beautiful
oils
self
gel
looks
zeal
builder
drafted
sewing
dress
dance
steals
effect
trips
brilliant
genius
flirt
nuclear
yard
pageant
earrings
divorce
firms
cocky
journeyman
nurses
seeking
ties
guru
salon
tearful
cold
buddy
cow
voters
youth
rule
sassy
breasts
pearls
vases
iv
regional
firmly
buddies
burly
homemaker
dancer
friend
babe
lamb
folks
priest
mate
beard
mommy
witch
witches
dads
boys
cousin
chap
boyhood
actresses
gals
wives
lad
fiance
sons son
queen
girlfriends
girlfriend
brothers
sisters
grandmother
wife
daddy
nephew
ladies
fiancee
daughters
biased
he
okay

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

$$\underset{\mathbf{D},\mathbf{A}}{\operatorname{argmin}} \sum_{i=1}^{V} \|\boldsymbol{x}_i - \mathbf{D}\boldsymbol{a}_i\|_2^2 + \lambda \|\boldsymbol{a}_i\|_1 + \tau \|\mathbf{D}\|_2^2 \tag{23}$$

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]
  - Discourse information [Ji and Eisenstein, 2014b]

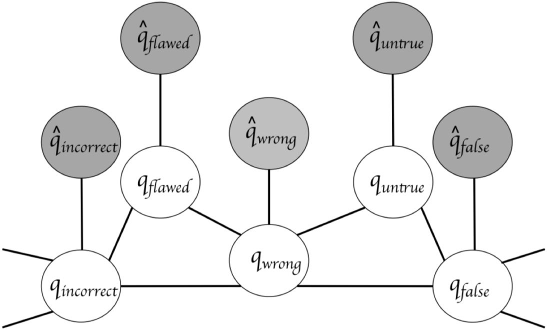
Problem

- 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]
  - Discourse information [Ji and Eisenstein, 2014b]
- Solutions
  - retrofitting word embeddings [Faruqui et al., 2014]
  - learning from supervision information [Ji and Eisenstein, 2014b]

Retrofitting

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

$$\Psi(\tilde{\mathbf{V}}) = \sum_{i=1}^{|\mathcal{V}|} \left[\alpha_i \|\boldsymbol{v}_i - \tilde{\boldsymbol{v}}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|\tilde{\boldsymbol{v}}_i - \tilde{\boldsymbol{v}}_j\|^2\right] \tag{24}$$

Learning from Supervision Signal

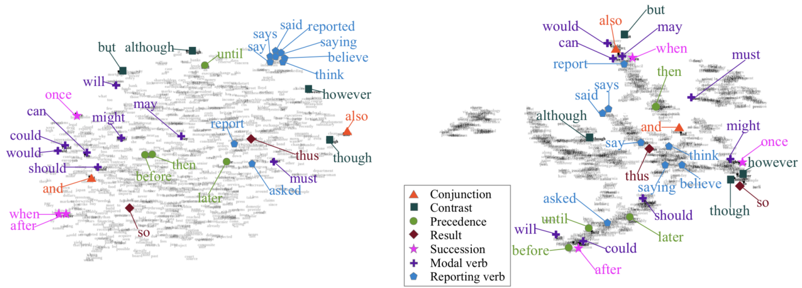


Figure: (Left) Word embeddings learned with supervision signal; (Right) Unsupervised word embeddings.

Reference

Andreas, J. and Klein, D. (2014).
How much do word embeddings encode about syntax?
In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 822–827.

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., Rastogi, P., and Dyer, C. (2016).
Problems with evaluation of word embeddings using word similarity tasks.
arXiv preprint arXiv:1605.02276.

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. (2014a).
One vector is not enough: Entity-augmented distributional semantics for discourse relations.
arXiv preprint arXiv:1411.6699.

Ji, Y. and Eisenstein, J. (2014b).
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.

Levy, O. and Goldberg, Y. (2014).
Neural word embedding as implicit matrix factorization.
In *Advances in neural information processing systems*, pages 2177–2185.