25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

Enhancing Domain Word Embedding via Latent Semantic Imputation

Shibo Yao, Dantong Yu, Keli Xiao

Outline

Word Embedding and issues.

How to fuse different data sources in word embedding?

Problem Definition.

Model and Analysis.

Empirical Study.

Conclusions.

Word Representation Learning

knowledge discovery and data mining

$$v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \qquad v_i \in \mathbb{R}^n$$

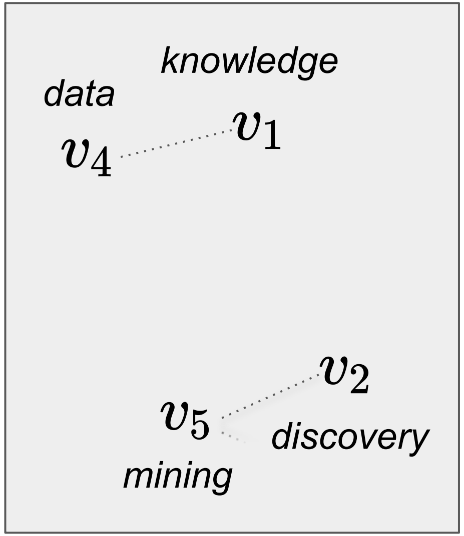
Unsupervised / Self-supervised Learning on Corpora

Word2Vec

Glove

FastText

Matrix Factorization



Issues

Limited-size corpus

Low-frequency domain words

Out-of-vocabulary words

How to fuse different data sources into word embedding?

Example in Chemistry and Medicine

ethanol v_1

calcium carbonate v_2

potassium v_3

$\vdots$

	State of mass	organic ?	metal ?	. .
ethanol	liquid	1	0	
calcium carbonate	solid	0	0	
potassium	solid	0	1	
$\vdots$				

Semantic Space $\mathbb{R}^S$

Domain Data $\mathbb{R}^D$

Example in Financial Text

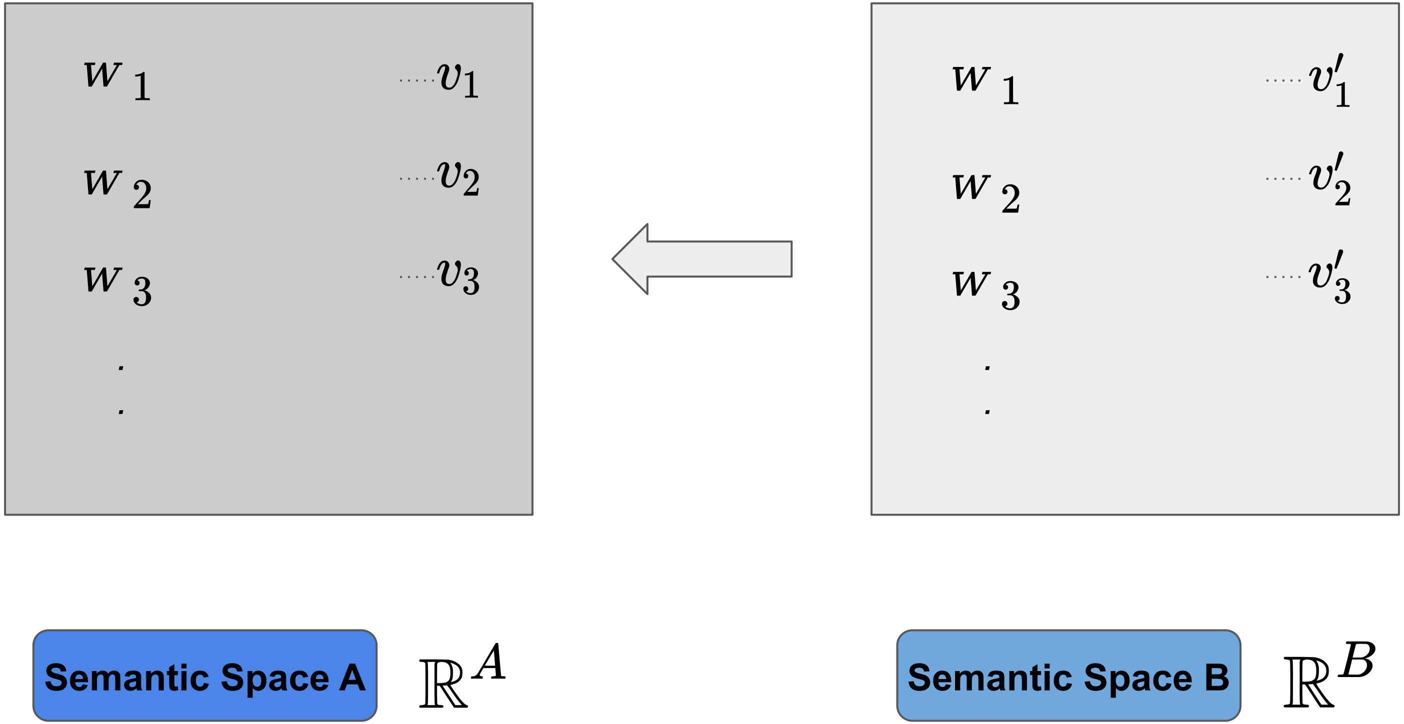
nvidia	v_1
qualcomm	v_2
google	v_3
. .	

	Return of day1	**Return of day2**	. .	**Return of dayD**
nvidia	0.23%	-1.21%	. .	2.49%
qualcomm	0.34%	-0.98%	. .	3.07%
google	-0.11%	-0.55%	. .	1.86%
. .				

Semantic Space $\mathbb{R}^S$

Domain Data $\mathbb{R}^D$

Two Semantic Spaces



Contributions

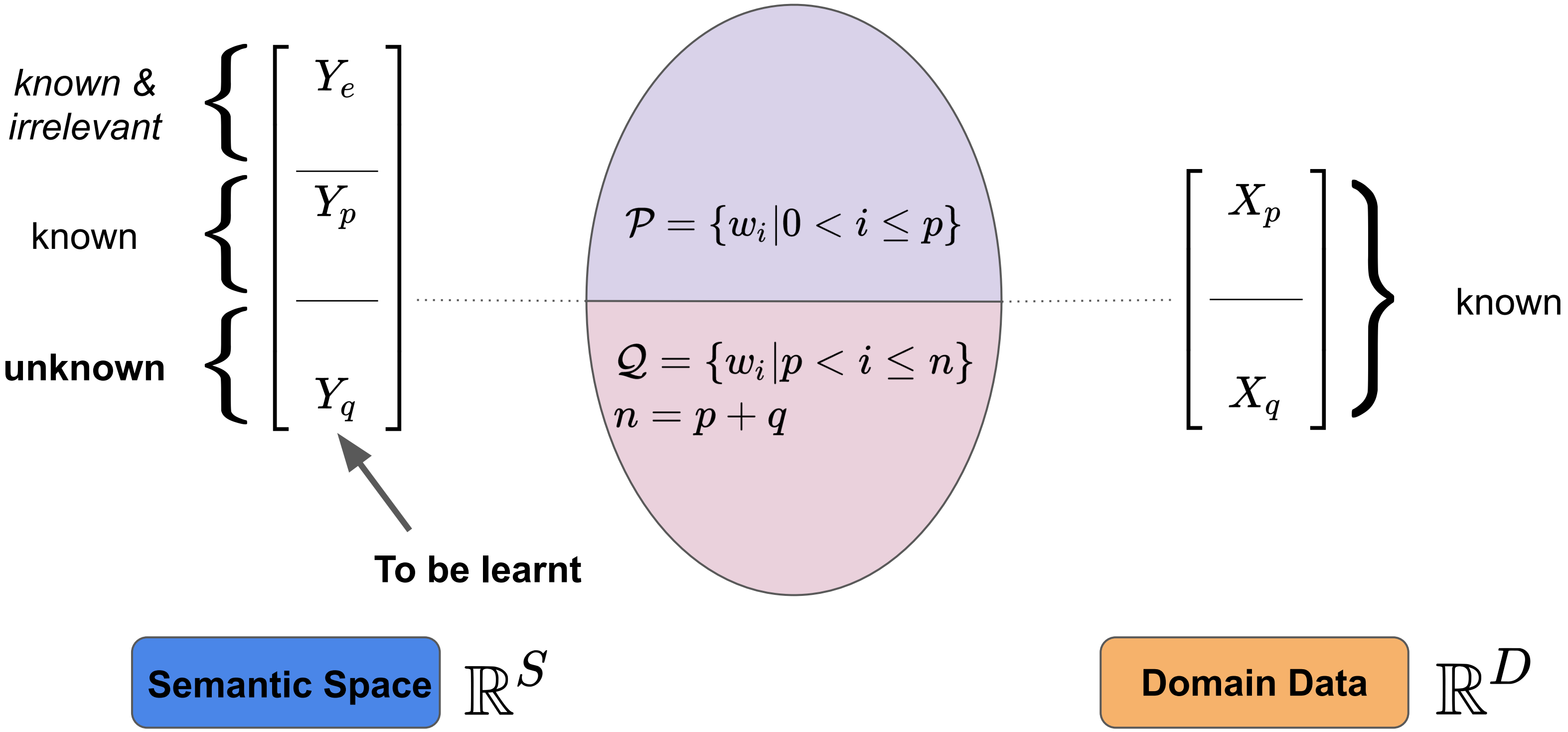
Formalize the problem of combining embeddings in different spaces.

Propose *Latent Semantic Imputation* to solve it.

Prove the deterministic convergence of *LSI*.

Conduct experiments to support our arguments.

Problem Definition



Assumptions

The **local** geometric structures of the data points in $\mathbb{R}^S$ and $\mathbb{R}^D$ are the same.

ST Roweis, LK Saul. 2000 Locally Linear Embedding

$$\begin{bmatrix} Y_p \\ \text{——} \\ Y_q \end{bmatrix} \xleftarrow{\mathcal{G} = (V, E)} \begin{bmatrix} X_p \\ \text{——} \\ X_q \end{bmatrix}$$

To be learnt

Each data point can be represented by a linear combination of its ***one-hop in-neighbors*** in the graph.

The Model

1. Given ***X***, build a ***MST-kNN* Graph** (***0-1 Adjacency Matrix***) based on Euclidean distance

2. Obtain a ***Weighted Adjacency Matrix***

$$\begin{aligned} \underset{\mathbf{w}_i}{\operatorname{argmin}} \quad & \|\mathbf{x}_i - \sum_j w_{ij}\mathbf{x}_j\|^2 \\ \text{s.t.} \quad & \sum_j w_{ij} = 1, \\ & j \in \{k|(v_k, v_i) \in E\}, \\ & w_{ij} \geq 0 \end{aligned}$$

$$W = \begin{bmatrix} W_p \\ W_q \end{bmatrix} = \left[\begin{array}{c|c} W_{pp} & W_{pq} \\ \hline W_{qp} & W_{qq} \end{array}\right]$$

$$\rightarrow \left[\begin{array}{c|c} I_p & 0 \\ \hline W_{qp} & W_{qq} \end{array}\right]$$

3. Power Iteration to solve ***Y***

$$Y^{(t+1)} = WY^{(t)}$$

Latent Semantic Imputation guarantees ***Deterministic Convergence***.

$$\lim_{t\to\infty} \begin{bmatrix} Y_p^{(t)} \\ Y_q^{(t)} \end{bmatrix} = \lim_{t\to\infty} W^t \begin{bmatrix} Y_p^{(0)} \\ Y_q^{(0)} \end{bmatrix}$$

A High-level Proof of Convergence

W is a ***Random Walk*** Matrix.

$$\forall i, \sum_j w_{ij} = 1$$

$$\forall i \forall j, w_{ij} \geq 0$$

W does not have any eigenvalue of -1.

$$\forall i, -1 < \lambda_i \leq 1$$

Each dimension of *Y* will converge to a linear combination of *W*'s ***dominant eigenvectors***.

$$\lim_{t\to\infty} W^t \vec{b} = \lim_{t\to\infty} W^t \sum_i c_i \vec{v}_i$$

$$= \lim_{t\to\infty} \sum_i c_i W^t \vec{v}_i$$

$$= \lim_{t\to\infty} \sum_i c_i \lambda_i^t \vec{v}_i = \sum_k c_k \vec{v}_k$$

A High-level Proof of Deterministic Convergence

$$W = \left[\begin{array}{c|c} I_p & 0 \\ \hline W_{qp} & W_{qq} \end{array}\right]$$

$$\lim_{t\to\infty} Y_q^{(t)} = \lim_{t\to\infty} W_{qq}^t Y_q^{(0)} + \lim_{t\to\infty} [\sum_{i=0}^{t-1} W_{qq}^t] W_{qp} Y_p$$

Xiaojin Zhu, Zoubin Ghahramani. 2002 Label Propagation

W_{qq} is a ***Substochastic*** Matrix.

$$\exists i, r_i < 1$$

In *MST-kNNG*, for every node in V_q there exists a path from V_p to it.

W_{qq} is a ***Convergent*** Matrix.

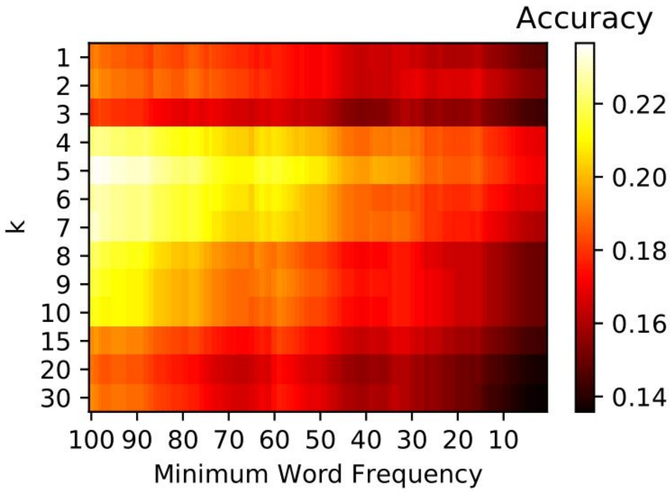
$$\forall i, r_i^{(t)} < 1 \qquad \lim_{t\to\infty} W_{qq}^t = 0$$

Empirical Study

Argument: low-frequency words are associated with unreliable embedding vectors.

Task 1: Classification on Embedding Vectors

Textual Data: ~50k Wiki articles about S&P 500 companies.
(word2vec, self-trained with Tensorflow example code)



Empirical Study

Argument: fusing a different data source can enhance word embedding.

Task 1: Classification on Embedding Vectors

Domain Data: S&P 500 company historical daily stock returns.
Pretrained embeddings: word2vec, fastText, Glove, self-trained.

E \ k	2	5	8	10	15	20	30
self	0.154	0.170	0.150	0.150	0.144	0.138	0.135
self(hf)	0.180	0.190	0.172	0.167	0.157	0.157	0.157
self(hf)+aff	0.556	0.472	0.396	0.359	0.302	0.261	0.187
Google	0.220	0.297	0.271	0.305	0.280	0.280	0.186
Google+aff	0.838	0.803	0.784	0.768	0.725	0.678	0.626
Glove	0.417	0.466	0.490	0.500	0.500	0.505	0.451
Glove+aff	0.832	0.766	0.690	0.653	0.606	0.542	0.405
fast	0.443	0.496	0.527	0.500	0.511	0.470	0.447
fast+aff	0.811	0.749	0.713	0.684	0.641	0.608	0.595

Empirical Study

Argument: Enhanced domain word embedding can facilitate downstream LM.

Task 2: Language modeling (LSTM, Tensorflow example code)

Textual Data: ~50k financial news headlines about ~4k companies retrieved from *WRDS*.
Domain Data: ~4k company historical daily stock return.
Pretrained embeddings: word2vec, fastText, Glove, self-trained.

Embedding	Test PP	%decrease
self	13.093	
self+Google	12.742	2.75
self+fastText	12.477	4.94
self+Glove	12.646	3.54
Google	12.431	5.33
fastText	12.215	7.19
Glove	12.218	7.16
self+aff	11.883	10.18
Google+aff	11.646	12.42
fastText+aff	11.638	12.51
Glove+aff	11.510	13.76

Conclusions

We can leverage different data sources to enhance word representation.

LSI is proposed to combine entity representations defined in different spaces.

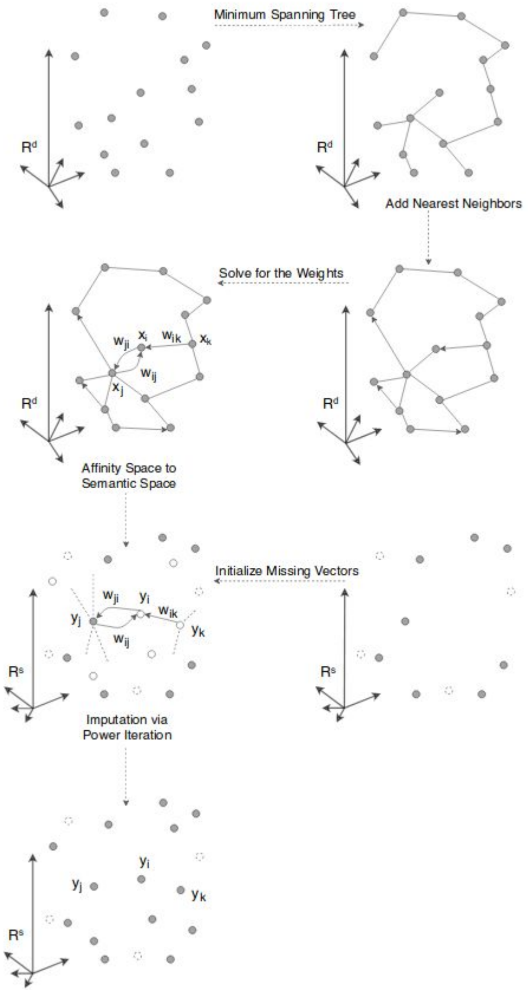
LSI guarantees deterministic convergence and has few hyperparameters.

Empirical study results support our arguments.

Acknowledgement

Yiannis Koutis, New Jersey Institute of Technology

Xiangmin (Jim) Jiao, Stony Brook University


Minimum Spanning Tree
R^d
R^d
Add Nearest Neighbors
Solve for the Weights
w_{ji}
x_i
w_{ik}
x_k
w_{ij}
x_j
R^d
R^d
Affinity Space to Semantic Space
Initialize Missing Vectors
w_{ji}
y_i
w_{ik}
y_j
y_k
w_{ij}
R^s
R^s
Imputation via Power Iteration
y_i
y_j
y_k
R^s