PTE : Predictive Text Embedding through Large-scale Heterogeneous Text Networks

Pramod Srinivasan

CS591txt - Text Mining Seminar

University of Illinois, Urbana-Champaign

April 8, 2016

Outline

Motivation

Introduction
- The Data Sparseness Problem

Related Work
- Overview
- Information Network Embedding

Predictive Text Embedding
- The Text Representation
- Heterogeneous Text Network Embedding
- Bipartite Network Embedding
- Text Embedding

Experiments
- Statistics of the Dataset
- Results
- Results

Analysis

Conclusion

Motivation

- Learning a meaningful and effective representation of text

Motivation

- Learning a meaningful and effective representation of text
- Critical pre-cursor to several machine learning tasks

Motivation

– Learning a meaningful and effective representation of text

– Critical pre-cursor to several machine learning tasks

– PTE's Goals
  – Capture Semantic Relatedness between words

Motivation

- Learning a meaningful and effective representation of text
- Critical pre-cursor to several machine learning tasks

- PTE's Goals
  - Capture Semantic Relatedness between words
  - Avoid issues such as data sparsity, polysemy and synonymy

Motivation

– Learning a meaningful and effective representation of text
– Critical pre-cursor to several machine learning tasks

– PTE's Goals
  – Capture Semantic Relatedness between words
  – Avoid issues such as data sparsity, polysemy and synonymy

Motivation

Introduction
- The Data Sparseness Problem

Related Work
- Overview
- Information Network Embedding

Predictive Text Embedding
- The Text Representation
- Heterogeneous Text Network Embedding
- Bipartite Network Embedding
- Text Embedding

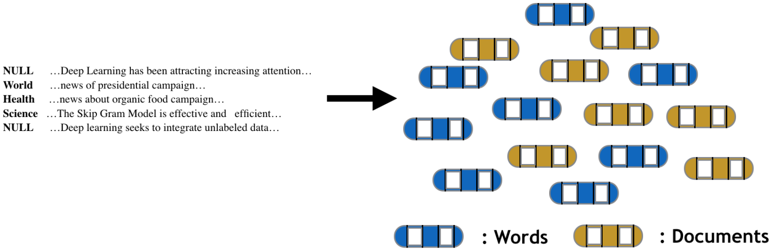
Experiments
- Statistics of the Dataset
- Results
- Results

Analysis

Conclusion

Solving the Problem of Data Sparseness($|V|$) : Text Embedding

- Representing words and documents in low-dimensional space
- Words and documents with similar meanings are embedded closely to each other



Motivation

Introduction
The Data Sparseness Problem

Related Work
Overview
Information Network Embedding

Predictive Text Embedding
The Text Representation
Heterogeneous Text Network Embedding
Bipartite Network Embedding
Text Embedding

Experiments
Statistics of the Dataset
Results
Results

Analysis

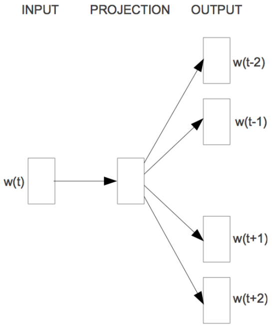
Conclusion

Unsupervised Text Embedding

- CBOW (Mikolov et al. 2013)
- Skip-Gram (Mikolov et al. 2013)
- Paragraph Vector (Le et al. 2014)

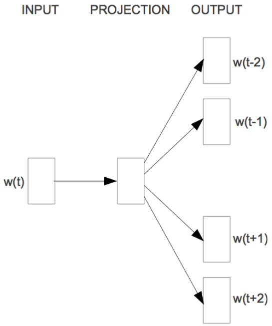
Unsupervised Text Embedding

- CBOW (Mikolov et al. 2013)
- Skip-Gram (Mikolov et al. 2013)
- Paragraph Vector (Le et al. 2014)



Unsupervised Text Embedding

- CBOW (Mikolov et al. 2013)
- Skip-Gram (Mikolov et al. 2013)
- Paragraph Vector (Le et al. 2014)

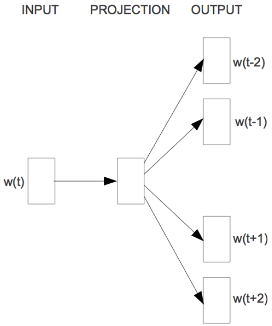


Pros

- Scalable, yet simple model
- Insensitive parameters
- Potential to leverage a large amount of unlabeled data, embeddings are general for different tasks

Unsupervised Text Embedding

- CBOW (Mikolov et al. 2013)
- Skip-Gram (Mikolov et al. 2013)
- Paragraph Vector (Le et al. 2014)



Pros

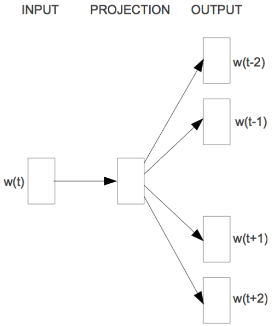
- Scalable, yet simple model
- Insensitive parameters
- Potential to leverage a large amount of unlabeled data, embeddings are general for different tasks

Cons

- Fully unsupervised, not tuned for specific tasks

Unsupervised Text Embedding

- CBOW (Mikolov et al. 2013)
- Skip-Gram (Mikolov et al. 2013)
- Paragraph Vector (Le et al. 2014)



Pros

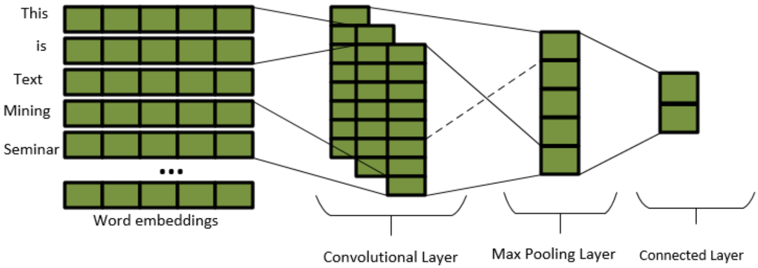
- Scalable, yet simple model
- Insensitive parameters
- Potential to leverage a large amount of unlabeled data, embeddings are general for different tasks

Cons

- Fully unsupervised, not tuned for specific tasks

(Deep) Neural Networks

- Recurrent Neural Networks (Mikolov et al. 2010)
- Recursive Neural Networks (Socher et al. 2012)
- Convolutional Neural Network (Kim et al. 2014)



Supervised Learning Model

- Recurrent Neural Networks (Mikolov et al. 2010)
- Recursive Neural Networks (Socher et al. 2012)
- Convolutional Neural Network (Kim et al. 2014)

Pros

- State-of-the-art performance on specific tasks

Cons

- Computationally expensive
- Require a large number of labeled data, hard to leverage unlabeled data
- *Very* sensitive to parameters, difficult to tune
- Potential to leverage a large amount of unlabeled data, embeddings are general for different tasks

Information Network Embedding

- Embedding one instance of some mathematical structure contained within another instance.
- Words that are used together with many similar words are likely to have similar meanings.

Information Network Embedding

– Embedding one instance of some mathematical structure contained within another instance.
– Words that are used together with many similar words are likely to have similar meanings.

The Ideal Embedding Model

- Reduced Parameter Space

The Ideal Embedding Model

- Reduced Parameter Space
- Improve generalization

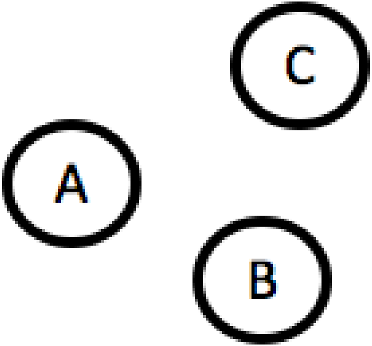
The Ideal Embedding Model

- Reduced Parameter Space
- Improve generalization
- Transfer or share knowledge between entities

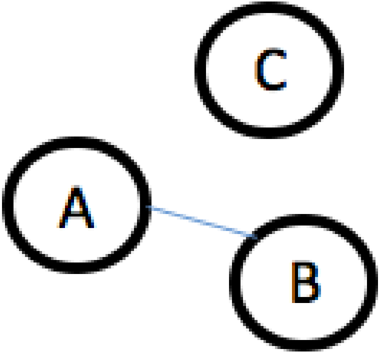
The Ideal Embedding Model

- Reduced Parameter Space
- Improve generalization
- Transfer or share knowledge between entities

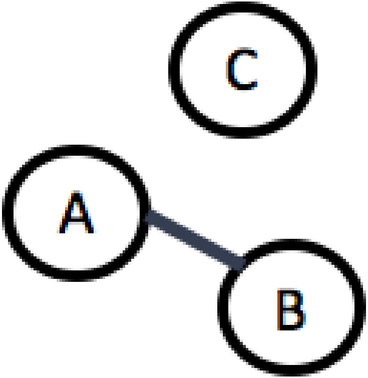
LINE : Intuition



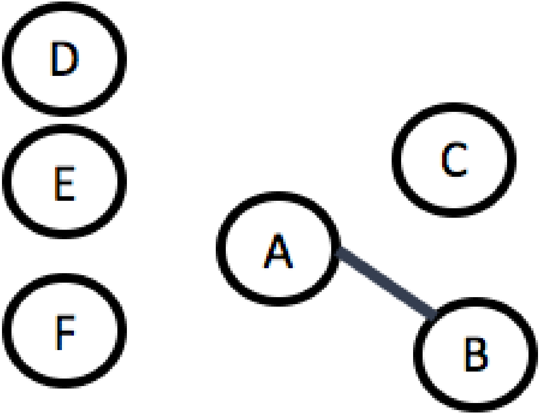
LINE : Intuition(contd)



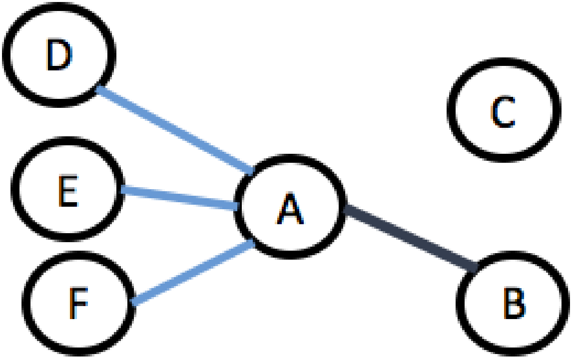
LINE : Intuition(contd)



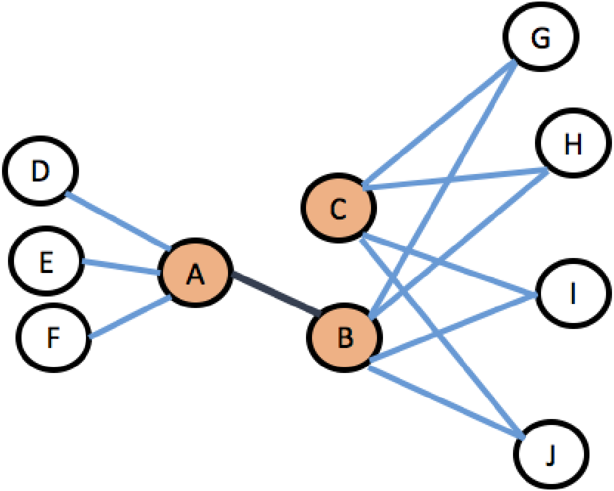
LINE : Intuition(contd)



LINE : Intuition(contd)



LINE : Intuition(contd)



Large-scale Information Network Embedding

– LINE extends the embedding idea to general information networks, more specifically, it transfers the vertices in a graph to vectors.

Large-scale Information Network Embedding

- LINE extends the embedding idea to general information networks, more specifically, it transfers the vertices in a graph to vectors.
- Preserves the *first-order* and *second-order* proximity between the vertices

Large-scale Information Network Embedding

- LINE extends the embedding idea to general information networks, more specifically, it transfers the vertices in a graph to vectors.
- Preserves the *first-order* and *second-order* proximity between the vertices
- **First-order proximity** : Observed Link between vertices

Large-scale Information Network Embedding

- LINE extends the embedding idea to general information networks, more specifically, it transfers the vertices in a graph to vectors.
- Preserves the *first-order* and *second-order* proximity between the vertices
- **First-order proximity** : Observed Link between vertices
- **Second-order proximity** : Proximity between their neighborhood structures

Large-scale Information Network Embedding

- LINE extends the embedding idea to general information networks, more specifically, it transfers the vertices in a graph to vectors.
- Preserves the *first-order* and *second-order* proximity between the vertices
- **First-order proximity** : Observed Link between vertices
- **Second-order proximity** : Proximity between their neighborhood structures

Motivation

Introduction
- The Data Sparseness Problem

Related Work
- Overview
- Information Network Embedding

Predictive Text Embedding
- The Text Representation
- Heterogeneous Text Network Embedding
- Bipartite Network Embedding
- Text Embedding

Experiments
- Statistics of the Dataset
- Results
- Results

Analysis

Conclusion

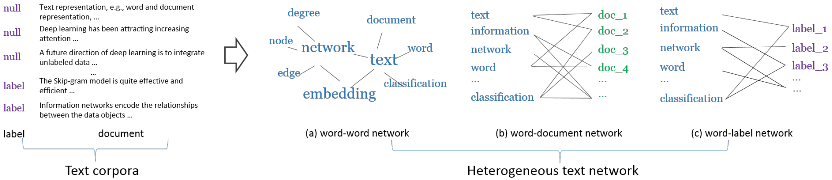
Predictive Text Embedding(PTE)

- Adapt the advantages of unsupervised text embedding approaches but naturally utilize the **labeled** data for specific tasks
- How to uniformly represent unsupervised and supervised information?

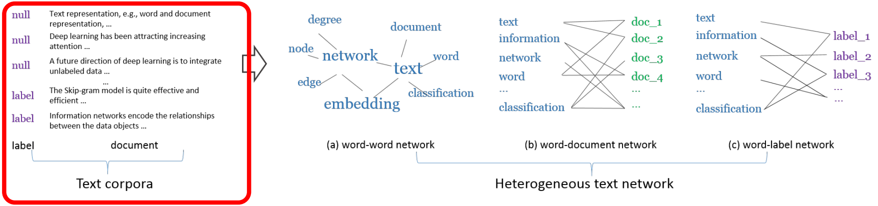
Predictive Text Embedding(PTE)

- Adapt the advantages of unsupervised text embedding approaches but naturally utilize the **labeled** data for specific tasks
- How to uniformly represent unsupervised and supervised information?
  - Heterogeneous Text Network
- Different Levels of Word Occurrences : *Word-Word Network, Word-Document Network, Word-Label Network*

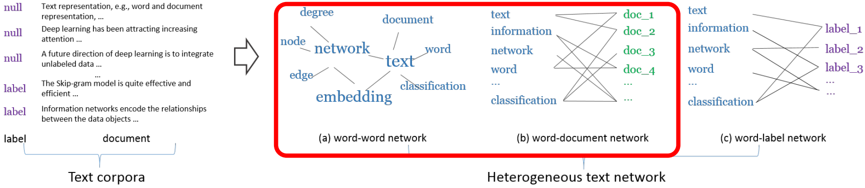
Converting Text Corpora



Partially Labeled Text Corpora

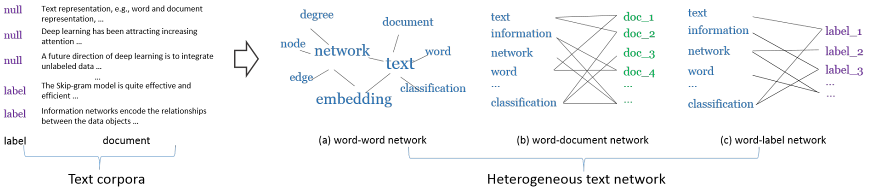


Word-Word and Word-Document Network



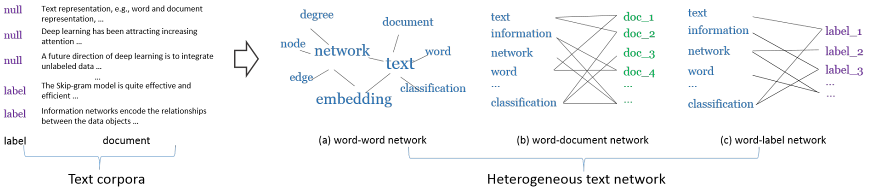
– Both word-document and word-word networks encode unsupervised information

Word-Word and Word-Document Network



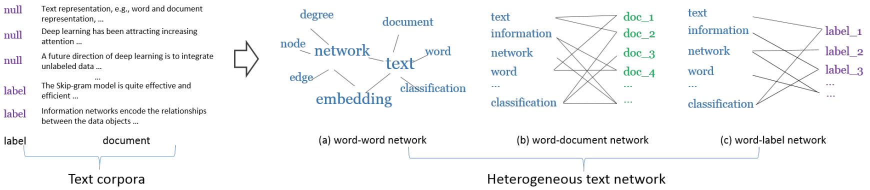
– Both word-document and word-word networks encode unsupervised information

Word-Word and Word-Document Network



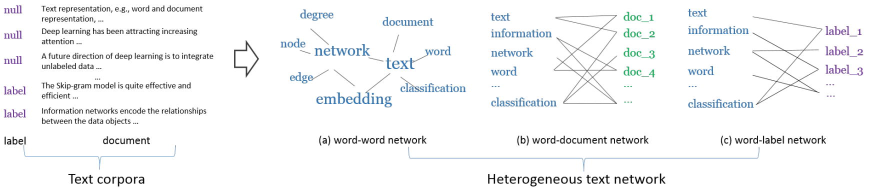
– Both word-document and word-word networks encode unsupervised information
– Word-Document $\equiv$ Topic Models, LDA

Word-Word and Word-Document Network



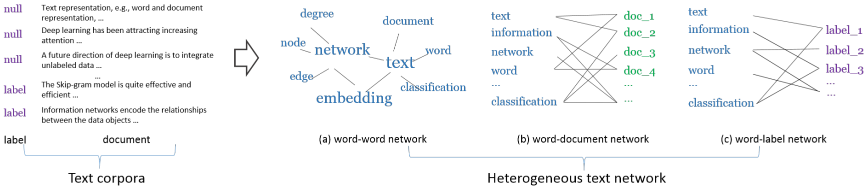
- Both word-document and word-word networks encode unsupervised information
- Word-Document $\equiv$ Topic Models, LDA
- Word-Word $\equiv$ Skip-Gram

Word-Word and Word-Document Network



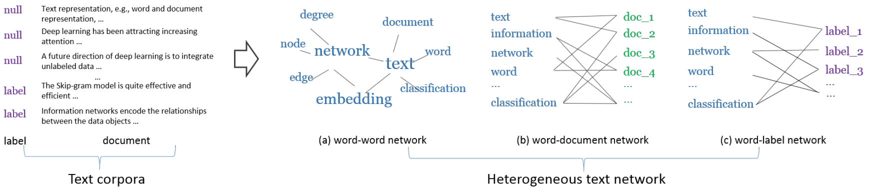
- Both word-document and word-word networks encode unsupervised information
- Word-Document $\equiv$ Topic Models, LDA
- Word-Word $\equiv$ Skip-Gram
- The weight between the word and document is its frequency in the document.

Word-Word and Word-Document Network



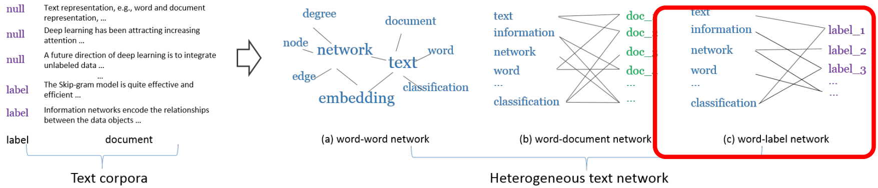
- Both word-document and word-word networks encode unsupervised information
- Word-Document $\equiv$ Topic Models, LDA
- Word-Word $\equiv$ Skip-Gram
- The weight between the word and document is its frequency in the document.
- The weight between two words is the number of times the two words co-occur in a *given window size*

Word-Word and Word-Document Network



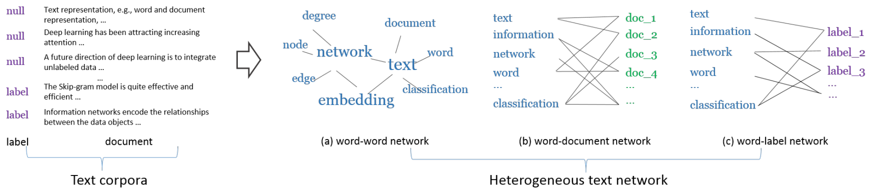
- Both word-document and word-word networks encode unsupervised information
- Word-Document $\equiv$ Topic Models, LDA
- Word-Word $\equiv$ Skip-Gram
- The weight between the word and document is its frequency in the document.
- The weight between two words is the number of times the two words co-occur in a *given window size*

Word Label Network



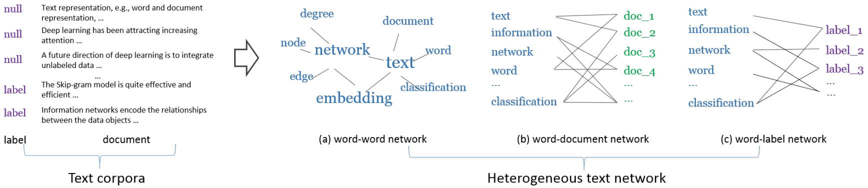
– Both word-document and word-word networks encode unsupervised information

Word Label Network



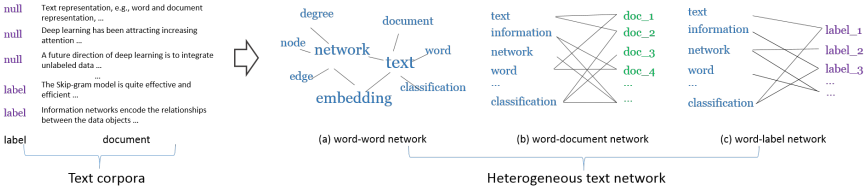
– Both word-document and word-word networks encode unsupervised information

Word Label Network



- Both word-document and word-word networks encode unsupervised information
- Word-Label Network encodes the supervised information

Word Label Network



- Both word-document and word-word networks encode unsupervised information
- Word-Label Network encodes the supervised information

Heterogeneous Text Network

Heterogeneous Text Network

- Three Bipartite Networks : Word-word(word-context), word-document and word-label network
- Encodes different levels of word co-occurency

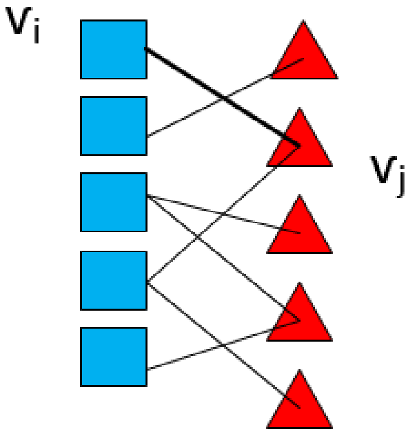
Heterogeneous Text Network

- Three Bipartite Networks : Word-word(word-context), word-document and word-label network
- Encodes different levels of word co-occurency
- Contains both `supervised` and `unsupervised` information

Heterogeneous Text Network

- Three Bipartite Networks : Word-word(word-context), word-document and word-label network
- Encodes different levels of word co-occurency
- Contains both `supervised` and `unsupervised` information
- Embedding a Heterogeneous Text Network, we obtain a very robust and optimized word embeddings for a specific task.

Bipartite Network Embedding



What is the ideal proximity measure?
Hint : It's either **First Order** or **Second Order**

Bipartite Network Embedding(contd.)

What is the ideal proximity measure?
Hint : It's either **First Order** or **Second Order**

Bipartite Network Embedding : Second-order proximity

– For each edge (v_i, v_j), define a conditional probability:

$$p_2(v_j|v_i) = \frac{e^{\vec{u}_j'^T \cdot \vec{u}_i}}{\sum_{k=1}^{|V|} e^{\vec{u}_k'^T \cdot \vec{u}_i}} \tag{1}$$

$$O_2 = -\sum_{(i,j)\in E} w_{ij} \log p_2(v_j|v_i). \tag{2}$$

Heterogeneous Text Network Embedding

Heterogeneous Text Network Embedding

– Jointly embed three bipartite networks

Heterogeneous Text Network Embedding

- Jointly embed three bipartite networks
- Objective Function :

$$O_{pte} = O_{ww} + O_{wd} + O_{wl}, \tag{3}$$

where

$$O_{ww} = - \sum_{(i,j) \in E_{ww}} w_{ij} \log p(v_i | v_j) \tag{4}$$

$$O_{wd} = - \sum_{(i,j) \in E_{wd}} w_{ij} \log p(v_i | d_j) \tag{5}$$

$$O_{wl} = - \sum_{(i,j) \in E_{wl}} w_{ij} \log p(v_i | l_j) \tag{6}$$

Optimization

Two different ways of optimization : Depends on when **labeled data(word-label network)** are utilized.

- Joint Training
  - Train the unlabeled data and the labeled data simultaneously
- Pre-training + Fine-Tuning
  - Jointly train the G_{ww} and G_{wd} networks
  - Fine tuning the word embeddings with the word-label network

Learning Word Representations

- *Robust* and *Optimized* word Embeddings for *specific tasks*
  - Containing different levels of word co-occurences.
  - Encoding both supervised and unsupervised data
- Given an arbitrary textpiece $d = w_1 w_2 ... w_n$
- For every w_i, the text embedding is given by $\vec{u}_i$.
- The vector representation of the embedding can be computed as :

$$\vec{d} = \frac{1}{n} \sum_{i}^{n} \vec{u_i} \tag{7}$$

Motivation

Introduction
The Data Sparseness Problem

Related Work
Overview
Information Network Embedding

Predictive Text Embedding
The Text Representation
Heterogeneous Text Network Embedding
Bipartite Network Embedding
Text Embedding

Experiments
Statistics of the Dataset
Results
Results

Analysis

Conclusion

Evaluating Effectiveness of PTE

Task : Text Classification

- Embeddings as Features
- Classifier : Logistic Regression

Compared Algorithms

- **BOW** : Classical "bag-of-words" representation

Evaluating Effectiveness of PTE

Task : Text Classification

- Embeddings as Features
- Classifier : Logistic Regression

Compared Algorithms

- **BOW** : Classical "bag-of-words" representation
- **Unsupervised Text Embedding**
  - **Skip-Gram**
  - **Paragraph Vector(PV)**
  - **LINE**, applied to text networks

Evaluating Effectiveness of PTE

Task : Text Classification

- Embeddings as Features
- Classifier : Logistic Regression

Compared Algorithms

- **BOW** : Classical "bag-of-words" representation
- **Unsupervised Text Embedding**
  - **Skip-Gram**
  - **Paragraph Vector(PV)**
  - **LINE**, applied to text networks
- **Predictive Text Embedding : incorporates labels a.k.a Supervised Learning**

Evaluating Effectiveness of PTE

Task : Text Classification

- Embeddings as Features
- Classifier : Logistic Regression

Compared Algorithms

- **BOW** : Classical "bag-of-words" representation
- **Unsupervised Text Embedding**
  - **Skip-Gram**
  - **Paragraph Vector(PV)**
  - **LINE**, applied to text networks
- **Predictive Text Embedding : incorporates labels a.k.a Supervised Learning**
  - **CNN**
  - **PTE**

Datasets

	Long Documents							Short Documents		
Name	20NG	WIKI	IMDB	CORPORATE	ECONOMICS	GOVERNMENT	MARKET	DBLP	MR	TWITTER
Train	11,314	1,911,617*	25,000	245,650	77,242	138,990	132,040	61,479	7,108	800,000
Test	7,532	21,000	25,000	122,827	38,623	69,496	66,020	20,000	3,554	400,000
—V—	89,039	913,881	71,381	141,740	65,254	139,960	64,049	22,270	17,376	405,994
Doc. length	305.77	672.56	231.65	102.23	145.10	169.07	119.83	9.51	22.02	14.36
#classes	20	7	2	18	10	23	4	6	2	2

*In the WIKI data set, only 42,000 documents are labeled.

Results of Text Classification Long Documents : Unsupervised

		20NG		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	80.88	79.30	79.95	80.03	86.54	86.54
Unsupervised Embedding	Skip-gram	70.62	68.99	75.80	75.77	85.34	85.34
	PVDBOW	75.13	73.48	76.68	76.75	86.76	86.76
	PVDM	61.03	56.46	72.96	72.76	82.33	82.33
	LINE(G_{ww})	72.78	70.95	77.72	77.72	86.16	86.16
	LINE(G_{wd})	79.73	78.40	80.14	80.13	89.14	89.14
	LINE($G_{ww} + G_{wd}$)	78.74	77.39	79.91	79.94	89.07	89.07

Results of Text Classification Long Documents : Unsupervised

		20NG		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	80.88	79.30	79.95	80.03	86.54	86.54
Unsupervised Embedding	Skip-gram	70.62	68.99	75.80	75.77	85.34	85.34
	PVDBOW	75.13	73.48	76.68	76.75	86.76	86.76
	PVDM	61.03	56.46	72.96	72.76	82.33	82.33
	LINE(G_{ww})	72.78	70.95	77.72	77.72	86.16	86.16
	LINE(G_{wd})	79.73	78.40	80.14	80.13	89.14	89.14
	LINE($G_{ww} + G_{wd}$)	78.74	77.39	79.91	79.94	89.07	89.07

– Local context-level word co-occurences : LINE(G_{ww}) $>$ Skip-gram

Results of Text Classification Long Documents : Unsupervised

		20NG		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	80.88	79.30	79.95	80.03	86.54	86.54
Unsupervised Embedding	Skip-gram	70.62	68.99	75.80	75.77	85.34	85.34
	PVDBOW	75.13	73.48	76.68	76.75	86.76	86.76
	PVDM	61.03	56.46	72.96	72.76	82.33	82.33
	LINE(G_{ww})	72.78	70.95	77.72	77.72	86.16	86.16
	LINE(G_{wd})	79.73	78.40	80.14	80.13	89.14	89.14
	LINE($G_{ww} + G_{wd}$)	78.74	77.39	79.91	79.94	89.07	89.07

- Local context-level word co-occurences : LINE(G_{ww}) $>$ Skip-gram
- Document-Level word co-occurences : LINE(G_{wd}) $>$ PV

Results on Long Documents : Predictive

		20NG		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	80.88	79.30	79.95	80.03	86.54	86.54
Predictive Embedding	CNN	78.85	78.29	79.72	79.77	86.15	86.15
	CNN(pretrain)	80.15	79.43	79.25	79.32	89.00	89.00
	PTE(G_{wl})	82.70	81.97	79.00	79.02	85.98	85.98
	PTE($G_{ww} + G_{wl}$)	83.90	83.11	81.65	81.62	89.14	89.14
	PTE($G_{wd} + G_{wl}$)	**84.39**	**83.64**	82.29	82.27	89.76	89.76
	PTE(pretrain)	82.86	82.12	79.18	79.21	86.28	86.28
	PTE(joint)	84.20	83.39	**82.51**	**82.49**	**89.80**	**89.80**

Results on Long Documents : Predictive

		20NG		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	80.88	79.30	79.95	80.03	86.54	86.54
Predictive Embedding	CNN	78.85	78.29	79.72	79.77	86.15	86.15
	CNN(pretrain)	80.15	79.43	79.25	79.32	89.00	89.00
	PTE(G_{wl})	82.70	81.97	79.00	79.02	85.98	85.98
	PTE($G_{ww} + G_{wl}$)	83.90	83.11	81.65	81.62	89.14	89.14
	PTE($G_{wd} + G_{wl}$)	**84.39**	**83.64**	82.29	82.27	89.76	89.76
	PTE(pretrain)	82.86	82.12	79.18	79.21	86.28	86.28
	PTE(joint)	84.20	83.39	**82.51**	**82.49**	**89.80**	**89.80**

– PTE(joint) $>$ PTE(pretrain)

Results on Long Documents : Predictive

		20NG		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	80.88	79.30	79.95	80.03	86.54	86.54
Predictive Embedding	CNN	78.85	78.29	79.72	79.77	86.15	86.15
	CNN(pretrain)	80.15	79.43	79.25	79.32	89.00	89.00
	PTE(G_{wl})	82.70	81.97	79.00	79.02	85.98	85.98
	PTE($G_{ww} + G_{wl}$)	83.90	83.11	81.65	81.62	89.14	89.14
	PTE($G_{wd} + G_{wl}$)	**84.39**	**83.64**	82.29	82.27	89.76	89.76
	PTE(pretrain)	82.86	82.12	79.18	79.21	86.28	86.28
	PTE(joint)	84.20	83.39	**82.51**	**82.49**	**89.80**	**89.80**

- PTE(joint) $>$ PTE(pretrain)
- PTE(joint) $>$ PTE(G_{wl})

Results on Long Documents : Predictive

		20NG		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	80.88	79.30	79.95	80.03	86.54	86.54
Predictive Embedding	CNN	78.85	78.29	79.72	79.77	86.15	86.15
	CNN(pretrain)	80.15	79.43	79.25	79.32	89.00	89.00
	PTE(G_{wl})	82.70	81.97	79.00	79.02	85.98	85.98
	PTE($G_{ww} + G_{wl}$)	83.90	83.11	81.65	81.62	89.14	89.14
	PTE($G_{wd} + G_{wl}$)	**84.39**	**83.64**	82.29	82.27	89.76	89.76
	PTE(pretrain)	82.86	82.12	79.18	79.21	86.28	86.28
	PTE(joint)	84.20	83.39	**82.51**	**82.49**	**89.80**	**89.80**

- PTE(joint) $>$ PTE(pretrain)
- PTE(joint) $>$ PTE(G_{wl})
- PTE(joint) $>$ CNN/CNN(pretrain)

Results on Long Documents : Predictive

		20NG		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	80.88	79.30	79.95	80.03	86.54	86.54
Predictive Embedding	CNN	78.85	78.29	79.72	79.77	86.15	86.15
	CNN(pretrain)	80.15	79.43	79.25	79.32	89.00	89.00
	PTE(G_{wl})	82.70	81.97	79.00	79.02	85.98	85.98
	PTE($G_{ww} + G_{wl}$)	83.90	83.11	81.65	81.62	89.14	89.14
	PTE($G_{wd} + G_{wl}$)	**84.39**	**83.64**	82.29	82.27	89.76	89.76
	PTE(pretrain)	82.86	82.12	79.18	79.21	86.28	86.28
	PTE(joint)	84.20	83.39	**82.51**	**82.49**	**89.80**	**89.80**

- PTE(joint) $>$ PTE(pretrain)
- PTE(joint) $>$ PTE(G_{wl})
- PTE(joint) $>$ CNN/CNN(pretrain)

Results on Short Documents : Unsupervised

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	75.28	71.59	71.90	71.90	75.27	75.27
Unsupervised Embedding	Skip-gram	73.08	68.92	67.05	67.05	73.02	73.00
	PVDBOW	67.19	62.46	67.78	67.78	71.29	71.18
	PVDM	37.11	34.38	58.22	58.17	70.75	70.73
	LINE(G_{ww})	73.98	69.92	71.07	71.06	73.19	73.18
	LINE(G_{wd})	71.50	67.23	69.25	69.24	73.19	73.19
	LINE($G_{ww} + G_{wd}$)	74.22	70.12	71.13	71.12	73.84	73.84

Results on Short Documents : Unsupervised

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	75.28	71.59	71.90	71.90	75.27	75.27
Unsupervised Embedding	Skip-gram	73.08	68.92	67.05	67.05	73.02	73.00
	PVDBOW	67.19	62.46	67.78	67.78	71.29	71.18
	PVDM	37.11	34.38	58.22	58.17	70.75	70.73
	LINE(G_{ww})	73.98	69.92	71.07	71.06	73.19	73.18
	LINE(G_{wd})	71.50	67.23	69.25	69.24	73.19	73.19
	LINE($G_{ww} + G_{wd}$)	74.22	70.12	71.13	71.12	73.84	73.84

– Local context-level word co-occurences : LINE(G_{ww}) $>$ Skip-gram

Results on Short Documents : Unsupervised

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	75.28	71.59	71.90	71.90	75.27	75.27
Unsupervised Embedding	Skip-gram	73.08	68.92	67.05	67.05	73.02	73.00
	PVDBOW	67.19	62.46	67.78	67.78	71.29	71.18
	PVDM	37.11	34.38	58.22	58.17	70.75	70.73
	LINE(G_{ww})	73.98	69.92	71.07	71.06	73.19	73.18
	LINE(G_{wd})	71.50	67.23	69.25	69.24	73.19	73.19
	LINE($G_{ww} + G_{wd}$)	74.22	70.12	71.13	71.12	73.84	73.84

- Local context-level word co-occurences : LINE(G_{ww}) $>$ Skip-gram
- Document-Level word co-occurences
  - LINE(G_{wd}) $>$ PV
  - LINE($G_{ww} + G_{wd}$) $>$ LINE($G_w w$) $>$ LINE($G_w d$)

Results on Short Documents : Predictive

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	75.28	71.59	71.90	71.90	75.27	75.27
Predictive Embedding	CNN	76.16	73.08	72.71	72.69	**75.97**	**75.96**
	CNN(pretrain)	75.39	72.28	68.96	68.87	75.92	75.92
	PTE(G_{wl})	76.45	72.74	73.44	73.42	73.92	73.91
	PTE($G_{ww} + G_{wl}$)	76.80	73.28	72.93	72.92	74.93	74.92
	PTE($G_{wd} + G_{wl}$)	**77.46**	**74.03**	73.13	73.11	75.61	75.61
	PTE(pretrain)	76.53	72.94	73.27	73.24	73.79	73.79
	PTE(joint)	77.15	73.61	**73.58**	**73.57**	75.21	75.21

Results on Short Documents : Predictive

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	75.28	71.59	71.90	71.90	75.27	75.27
Predictive Embedding	CNN	76.16	73.08	72.71	72.69	**75.97**	**75.96**
	CNN(pretrain)	75.39	72.28	68.96	68.87	75.92	75.92
	PTE(G_{wl})	76.45	72.74	73.44	73.42	73.92	73.91
	PTE($G_{ww} + G_{wl}$)	76.80	73.28	72.93	72.92	74.93	74.92
	PTE($G_{wd} + G_{wl}$)	**77.46**	**74.03**	73.13	73.11	75.61	75.61
	PTE(pretrain)	76.53	72.94	73.27	73.24	73.79	73.79
	PTE(joint)	77.15	73.61	**73.58**	**73.57**	75.21	75.21

Results on Short Documents : Predictive

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	75.28	71.59	71.90	71.90	75.27	75.27
Predictive Embedding	CNN	76.16	73.08	72.71	72.69	**75.97**	**75.96**
	CNN(pretrain)	75.39	72.28	68.96	68.87	75.92	75.92
	PTE(G_{wl})	76.45	72.74	73.44	73.42	73.92	73.91
	PTE($G_{ww} + G_{wl}$)	76.80	73.28	72.93	72.92	74.93	74.92
	PTE($G_{wd} + G_{wl}$)	**77.46**	**74.03**	73.13	73.11	75.61	75.61
	PTE(pretrain)	76.53	72.94	73.27	73.24	73.79	73.79
	PTE(joint)	77.15	73.61	**73.58**	**73.57**	75.21	75.21

– PTE(joint) $>$ PTE(pretrain)

Results on Short Documents : Predictive

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	75.28	71.59	71.90	71.90	75.27	75.27
Predictive Embedding	CNN	76.16	73.08	72.71	72.69	**75.97**	**75.96**
	CNN(pretrain)	75.39	72.28	68.96	68.87	75.92	75.92
	PTE(G_{wl})	76.45	72.74	73.44	73.42	73.92	73.91
	PTE($G_{ww} + G_{wl}$)	76.80	73.28	72.93	72.92	74.93	74.92
	PTE($G_{wd} + G_{wl}$)	**77.46**	**74.03**	73.13	73.11	75.61	75.61
	PTE(pretrain)	76.53	72.94	73.27	73.24	73.79	73.79
	PTE(joint)	77.15	73.61	**73.58**	**73.57**	75.21	75.21

– PTE(joint) $>$ PTE(pretrain)

– PTE(joint) $>$ PTE(G_{wl})

Results on Short Documents : Predictive

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	75.28	71.59	71.90	71.90	75.27	75.27
Predictive Embedding	CNN	76.16	73.08	72.71	72.69	**75.97**	**75.96**
	CNN(pretrain)	75.39	72.28	68.96	68.87	75.92	75.92
	PTE(G_{wl})	76.45	72.74	73.44	73.42	73.92	73.91
	PTE($G_{ww} + G_{wl}$)	76.80	73.28	72.93	72.92	74.93	74.92
	PTE($G_{wd} + G_{wl}$)	**77.46**	**74.03**	73.13	73.11	75.61	75.61
	PTE(pretrain)	76.53	72.94	73.27	73.24	73.79	73.79
	PTE(joint)	77.15	73.61	**73.58**	**73.57**	75.21	75.21

- PTE(joint) $>$ PTE(pretrain)
- PTE(joint) $>$ PTE(G_{wl})
- PTE(joint) $>$ CNN/CNN(pretrain)

Results on Short Documents : Predictive

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word	BOW	75.28	71.59	71.90	71.90	75.27	75.27
Predictive Embedding	CNN	76.16	73.08	72.71	72.69	**75.97**	**75.96**
	CNN(pretrain)	75.39	72.28	68.96	68.87	75.92	75.92
	PTE(G_{wl})	76.45	72.74	73.44	73.42	73.92	73.91
	PTE($G_{ww} + G_{wl}$)	76.80	73.28	72.93	72.92	74.93	74.92
	PTE($G_{wd} + G_{wl}$)	**77.46**	**74.03**	73.13	73.11	75.61	75.61
	PTE(pretrain)	76.53	72.94	73.27	73.24	73.79	73.79
	PTE(joint)	77.15	73.61	**73.58**	**73.57**	75.21	75.21

– PTE(joint) $>$ PTE(pretrain)

– PTE(joint) $>$ PTE(G_{wl})

– PTE(joint) $>$ CNN/CNN(pretrain)

Motivation

Introduction
- The Data Sparseness Problem

Related Work
- Overview
- Information Network Embedding

Predictive Text Embedding
- The Text Representation
- Heterogeneous Text Network Embedding
- Bipartite Network Embedding
- Text Embedding

Experiments
- Statistics of the Dataset
- Results
- Results

Analysis

Conclusion

Unsupervised Embedding

Unsupervised Embedding

- Long Documents
  - Documemt-level word co-occurences are more useful than local Context-Level word co-occurences.
  - *No improvement observed* when these two co-occurences are combined.

Unsupervised Embedding

- Long Documents
  - Documemt-level word co-occurences are more useful than local Context-Level word co-occurences.
  - *No improvement observed* when these two co-occurences are combined.
- Short Documents
  - Local context-level word co-occurences are more useful than document-Level word co-occurences.
  - Combination further improves the embedding.

Summary

Summary

- Predictive Text Embedding
  - Adapt the advantages of unsupervised text embedding approaches
  - Naturally incorporate the labeled data
- Encode unsupervised and supervised information through Large-scale heterogeneous information networks
- Outperform or comparable to sophisticated methods such as CNN
  - Outperform CNN on long documents
  - Comparable to CNN on short documents

Takeaway

Predictive Text Embedding

Given a large collection of text data with unlabeled and *labeled* information, PTE aims to learn *low-dimensional* representations of words by **embedding** the **heterogeneous** representations of words by embedding the heterogeneous text network constructed from the collection into a low dimensional vector space.

Thank You !