Subword-level Word Vector Representations and their Interpretability

Sungjoon Park

171219 Naver TechTalk

Contents

- **Rotated Word Vector Representations and their Interpretability**

  In Proceedings of the Conference on Empirical Method of Natural Language Processing (EMNLP) 2017

- **Subword-level Word Vector Representations for Korean Language**

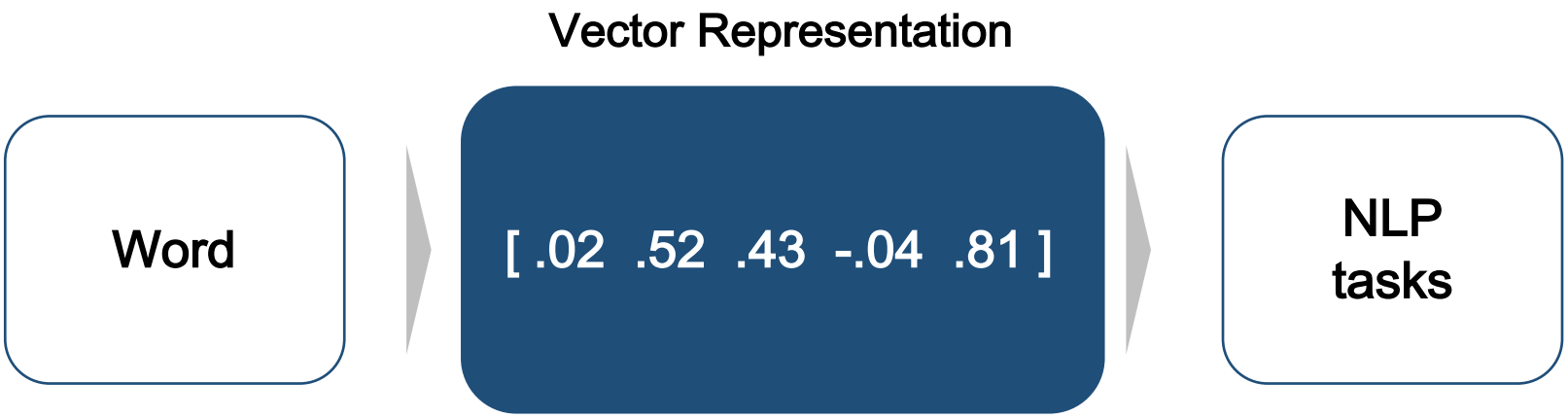
  (Work in Progress)

Rotated Word Vector Representations and their Interpretability

Sungjoon Park, Jinyeong Bak, Alice Oh

In Proceedings of the Conference on Empirical Method of Natural Language Processing (EMNLP) 2017

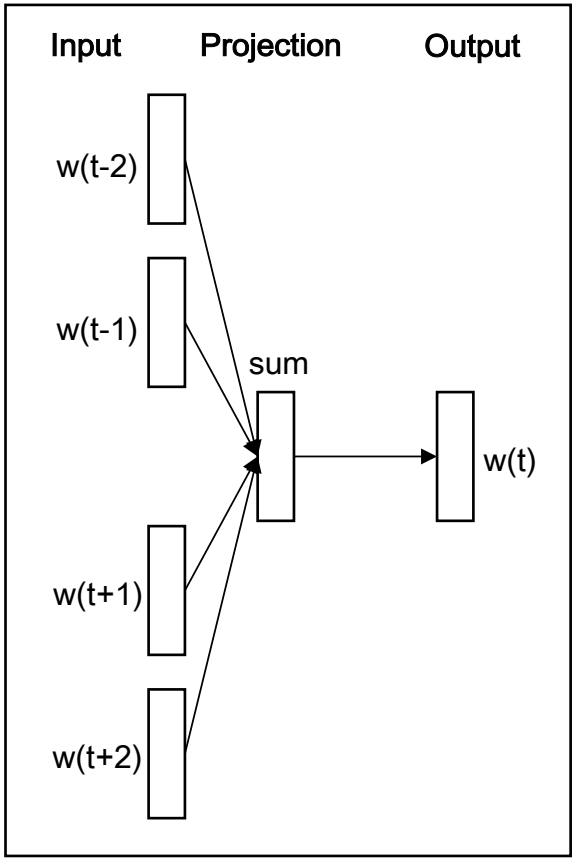
Introduction



- Word2vec (Mikolov, 2013)
- Glove (Pennington, et al., 2014)

Distributed Representation of Words

CBOW (Mikolov et al., 2013)



Objective Given a sequence of training words $w_{1\ldots t}$

$$\frac{1}{V}\sum_{V}\sum_{-c\leq j\leq c, j\neq 0}\log p\left(w_t\middle|w_{t+j}\right)$$

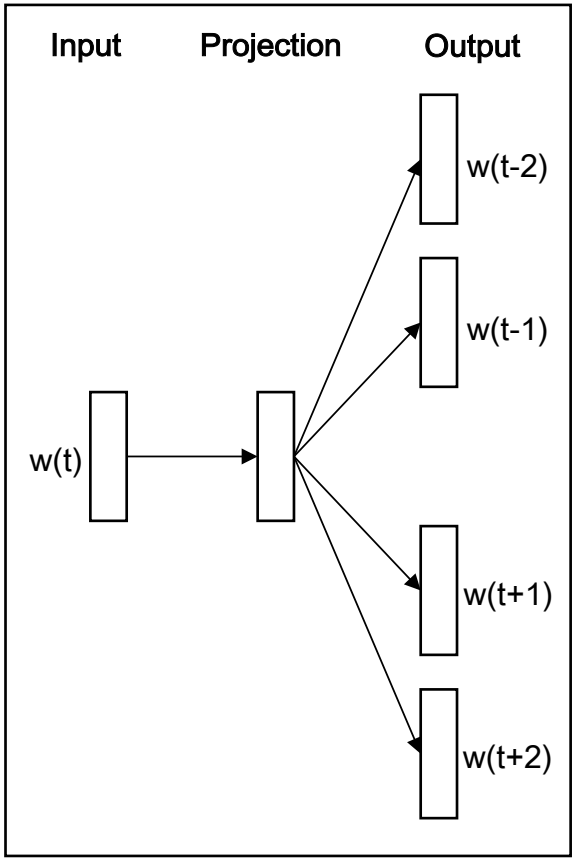
where $p(w_O|w_I)$ is a softmax:

$$p(w_O|w_I)=\frac{\exp(u_{w_O}^T v_{w_I})}{\sum_w \exp(u_w^T v_{w_I})}$$

computed by hierarchical softmax or negative sampling

Distributed Representation of Words

Skip-Gram (Mikolov et al., 2013)



Objective Given a sequence of training words $w_{1\ldots t}$

$$\frac{1}{V}\sum_{V}\sum_{-c\leq j\leq c, j\neq 0}\log p\left(w_{t+j}\middle|w_t\right)$$

where $p(w_O|w_I)$ is a softmax:

$$p(w_O|w_I)=\frac{\exp(u_{w_O}^T v_{w_I})}{\sum_w \exp(u_w^T v_{w_I})}$$

computed by hierarchical softmax or negative sampling

Distributed Representation of Words

Glove (Pennington et al., 2014)

Motivation Use global information (co-occurrence over corpus) while learning word vectors

Objective Dot product between two word vectors should be equal to
log of the words' probability of co-occurrence (with given context words)

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

$$F(w_i, w_j, w_k) = \frac{P_{ik}}{P_{jk}}$$

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^t \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

Introduction

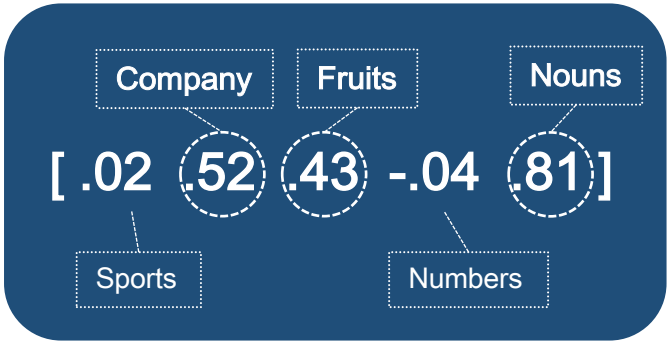
Vector Representation

Not clear what it means

Introduction

Interpretable Vector Representation

"Apple"



Sparsity

(Faruqui et al., 2015)

- Understanding semantic / syntactic compositionality of words
- Increasing efficiency of storage
- Reducing complexity of higher-level models

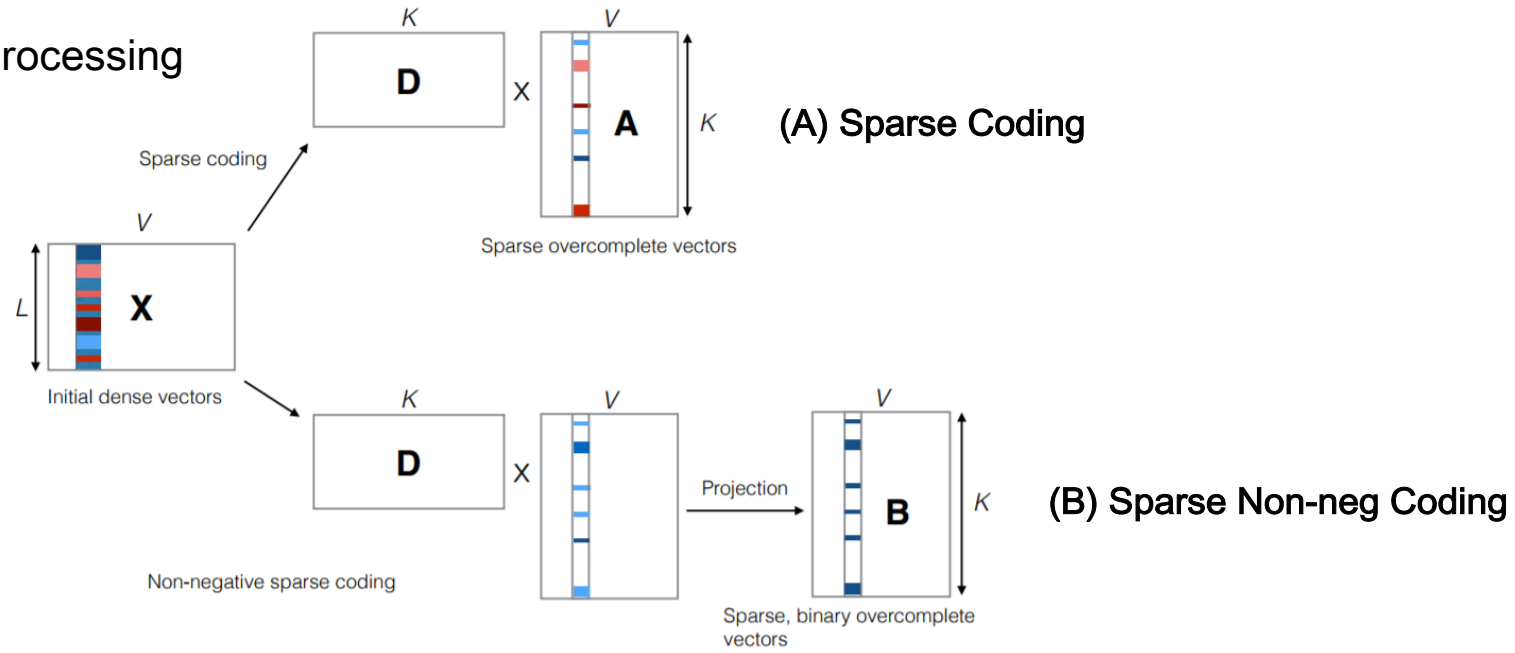
Related Work

Sparse L1 Regularized Word Vectors (Sun et al., IJCAI 2016)

- Stand-alone model
- Objective $\mathcal{L}_{s-cbow} = \mathcal{L}_{cbow} - \lambda \sum_{w \in W} \|\vec{w}\|_1$

Sparse Overcomplete Word Vectors (Faruqui et al., ACL 2015)

- Post-processing



Related Work

Sparse Overcomplete Representations (Faruqui et al., ACL 2015)

(A) Sparse Coding Minimizing reconstruction error,
with $l1$ regularizer $\Omega(A)$ and dictionary of basis vectors D.

$$\arg\min_{D,A} \sum_{i=1}^{V} \|x_i - Da_i\|^2 + \lambda\Omega(a_i) + \tau\|D\|_2^2$$

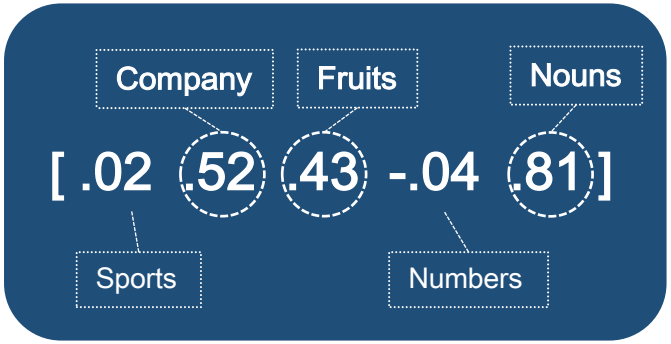
(B) Sparse Non-negative Coding Adding non-negativity constraint to (A)

$$\arg\min_{D,A} \sum_{i=1}^{V} \|x_i - Da_i\|^2 + \lambda\Omega(a_i) + \tau\|D\|_2^2 \ , D \in \mathbb{R}_{\geq 0}^{K \times V}, A \in \mathbb{R}_{\geq 0}^{K \times V}$$

- Trained by ADAGrad, resulting higher interpretability while preserving expressive performance
- X10 of number of dimensions tends to work well

Idea

Interpretable Vector Representation



Rotate Dimensions

(as a post-processing method)

Idea

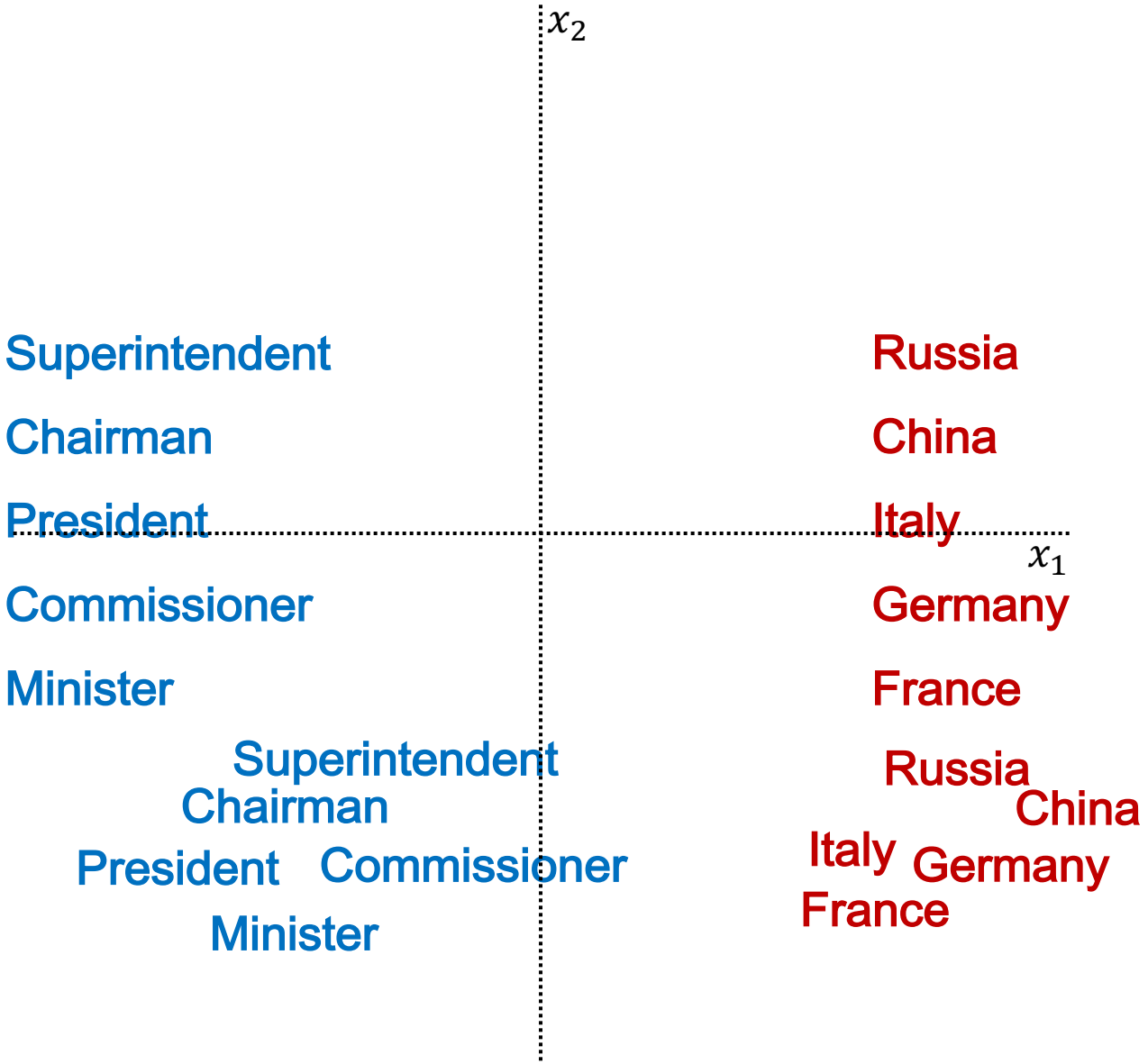
Position

- Superintendent
- Chairman
- President
- Commissioner
- Minister

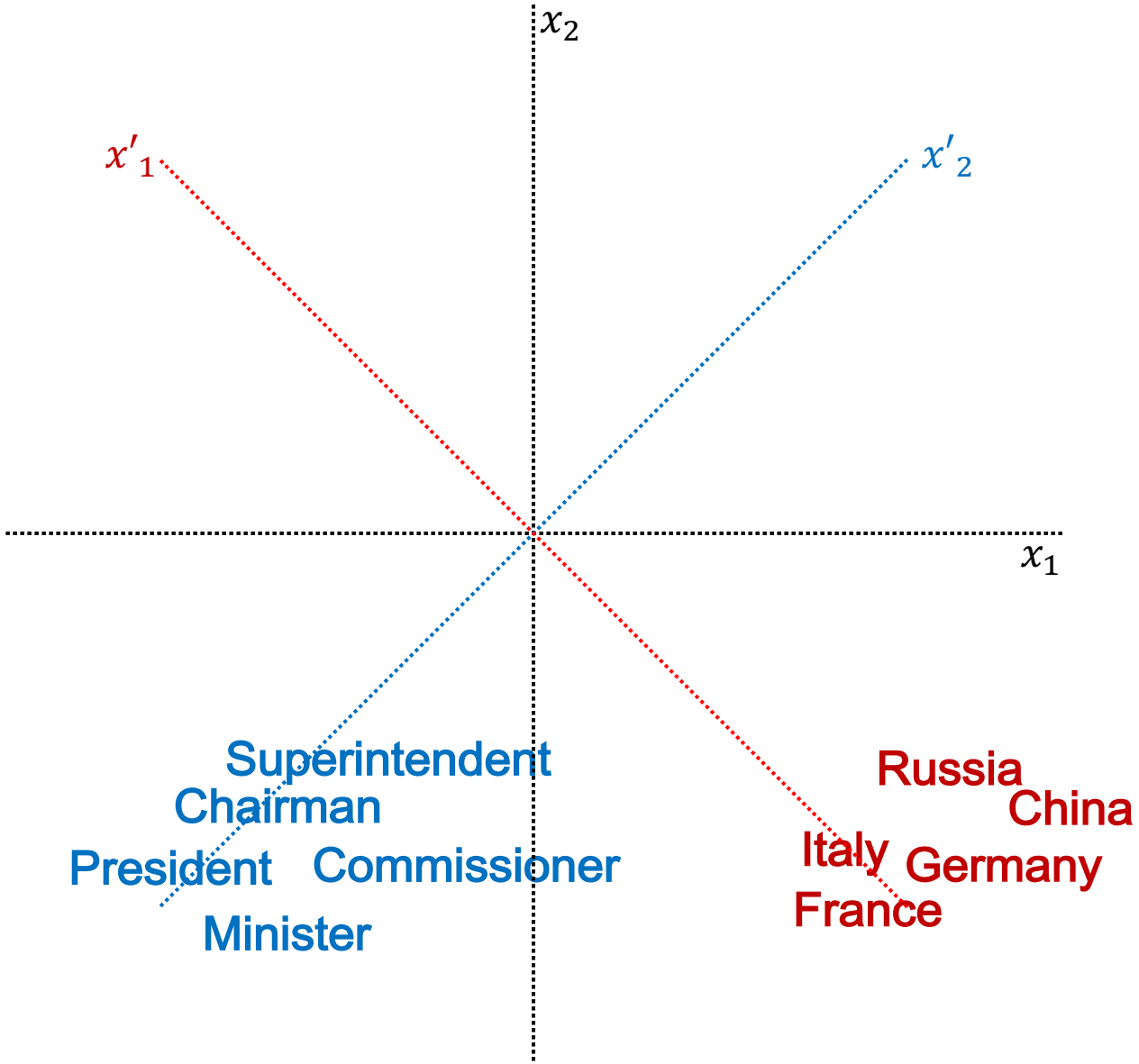
Nation

- Russia
- China
- Italy
- Germany
- France

Idea


x_2
Superintendent
Chairman
President
Commissioner
Minister
Russia
China
Italy
x_1
Germany
France
Superintendent
Chairman
President
Commissioner
Minister
Russia
China
Italy
Germany
France

Idea


x_2
x'_1
x'_2
x_1
Superintendent
Chairman
President
Commissioner
Minister
Russia
China
Italy
Germany
France

Factor Rotation

Exploratory Factor Analysis:

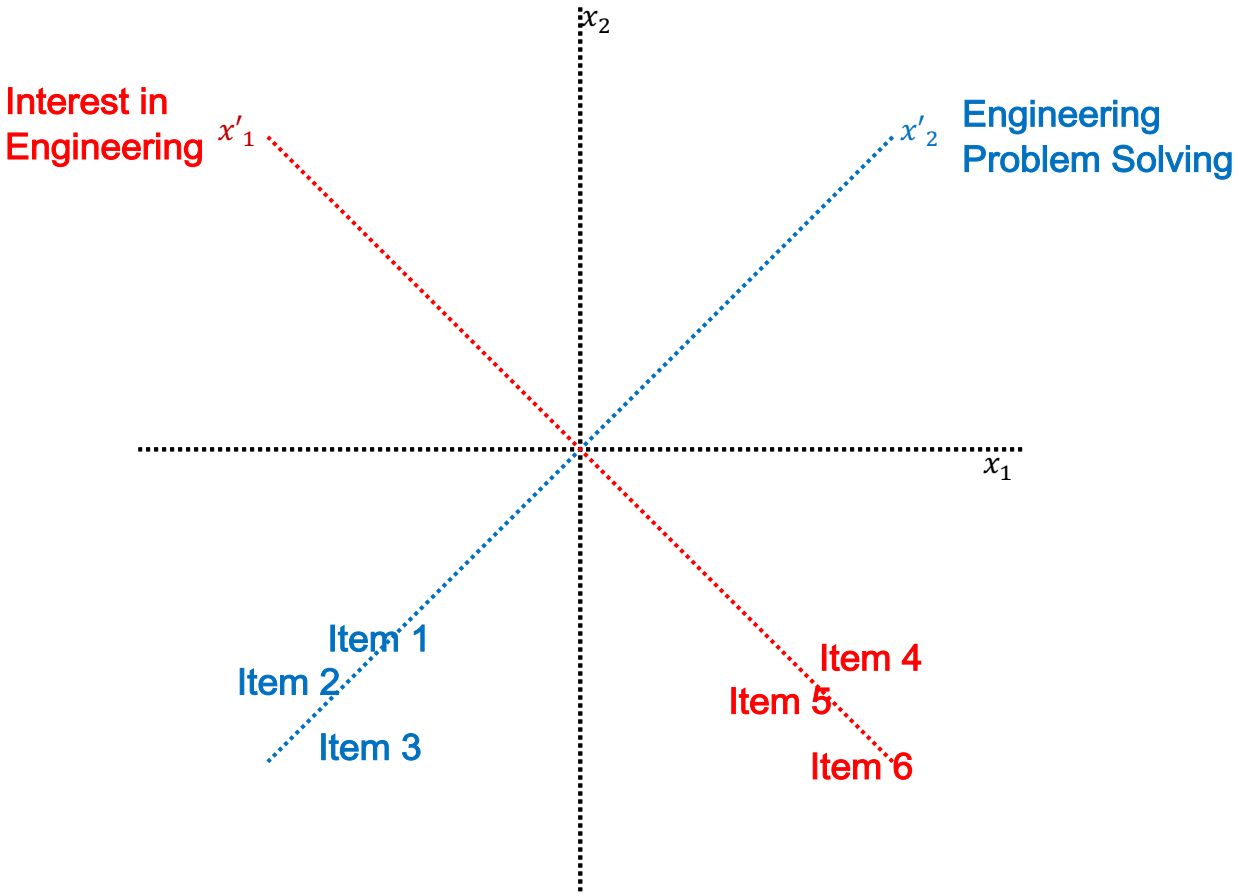
Embed the items to latent factor space by using factor analysis

Engineering Problem Solving	How well did you feel prepared for: (1) Defining what the problem really is (2) Thinking up potential solutions to the problem (3) Detailing how to implement the solution to the problem
Interest in Engineering	(1) I find many topics in engineering to be interesting (2) Solving engineering problems is interesting to me (3) Engineering fascinates me

Factor Rotation

Exploratory Factor Analysis:

Embed the items to latent factor space by using factor analysis



Method

How to: Crawford-Ferguson Rotation Family

- To Compute Λ: $\Lambda = AT$

  - Satisfying: $T'T = I$ *Orthogonal* $\quad diag(T^{-1}T^{-1\prime}) = I$ *Oblique*

  - Minimize: $f(\lambda) = (1-\kappa)\sum_{i=1}^{p}\sum_{j=1}^{m}\sum_{l\neq j,l=1}^{m}\lambda_{ij}^2\lambda_{il}^2 + \kappa\sum_{j=1}^{m}\sum_{i=1}^{p}\sum_{l\neq i,l=1}^{m}\lambda_{ij}^2\lambda_{lj}^2$

    Row complexity $\quad$ Column complexity

  - ✓ κ : weighting parameter

Quartimax	Varimax	Parsimax	Factor Parsimony
0	1/p	m-1/p+m-2	1

Method

Experimental Settings

- Training

English Wikipedia

- 5.3M articles
- 83M sentences
- 1,676 tokens

[.02 .52 .43 -.04 .81]

2 Vector Representations

- Word2Vec, Glove
- 306491 words
- 300 dimensions

[.21 .32 .63 -.04 .31]
[.47 .12 .42 .34 .04]
[.12 .54 .49 .88 .01]
[.54 .32 .63 -.04 .54]

16 Rotated Representations

- For each kappa (4)
- For each Embedding (2)
- For each constraint (2)

- Implementation
  - Algorithm : Gradient Projection (Jennrich. 2001)
  - Github: https://github.com/SungjoonPark/factor_rotation (TensorFlow)

Interpretability

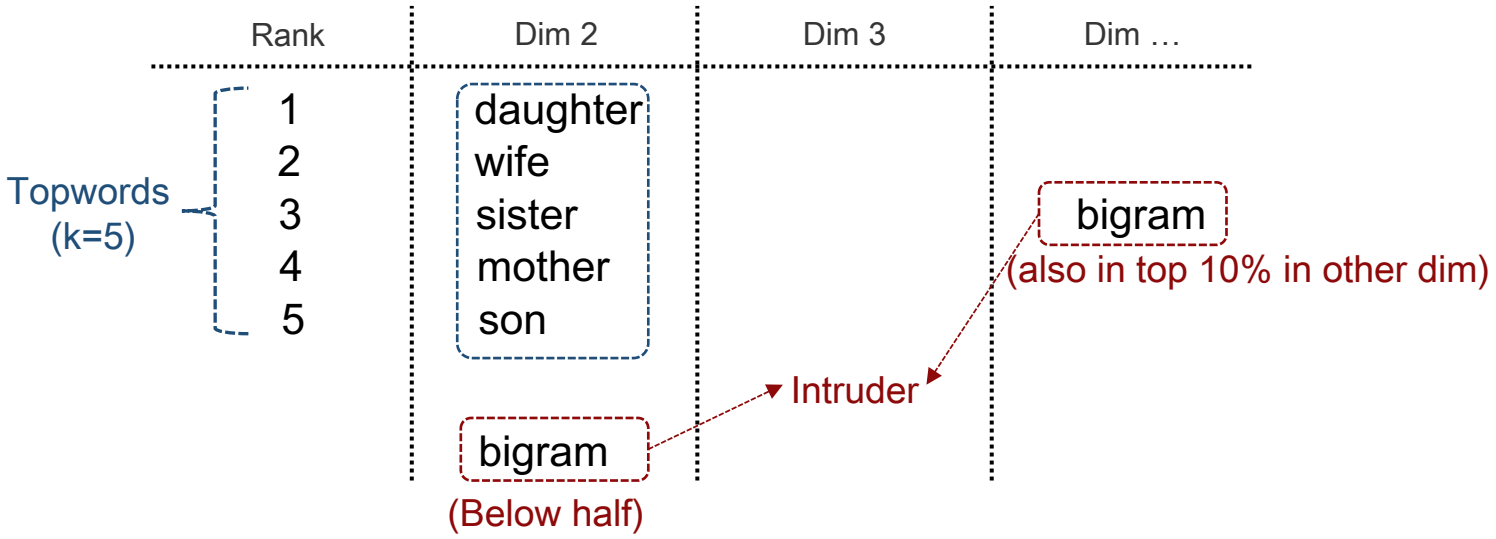
Task : Word Intrusion

To measure semantic coherence of words:

Intruder!

{‘daughter’, ‘wife’, ‘sister’, ‘mother’, ‘son’, ‘bigram’}

- Choosing an intruder (for that dimension):



Interpretability

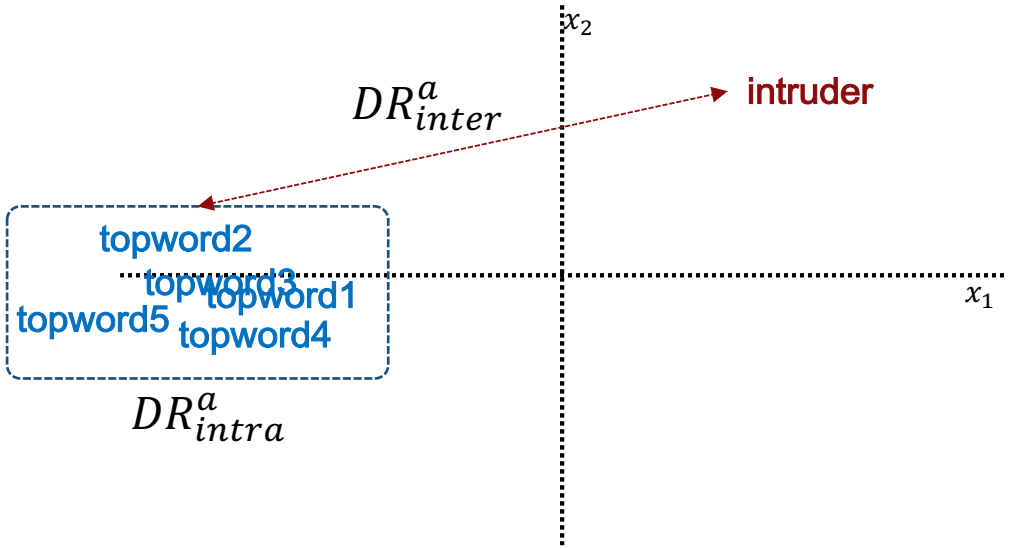
Measure: Distance Ratio

$$DR_{overall} = \frac{1}{d} \frac{\sum_{a=1}^{d} D_{inter}^{a}}{\sum_{a=1}^{d} D_{intra}^{a}}$$, where

$$DR_{intra}^{a} = \frac{\Sigma_{w_i} \Sigma_{w_j} dist(w_i, w_j)}{k(k-1)}$$

$$DR_{inter}^{a} = \frac{\Sigma_{w_i} dist(w_i, w_{intruder})}{k}$$

- Example:



Interpretability

Results

Quantitative Results

Distance Ratio	SG	Glove
Original	1.258	1.095
SOV	1.089	1.05
SOV (non-neg)	1.081	1.074
Quartimax (orthogonal)	1.479	1.248
Varimax (orthogonal)	1.477	1.289
Parsimax (orthogonal)	1.596	1.261
FacParsim (orthogonal)	1.3	1.102
Quartimax (oblique)	1.385	1.225
Varimax (oblique)	1.398	1.222
Parsimax (oblique)	1.386	1.174
FacParsim (oblique)	1.145	1.081

Qualitative Examples

Skip-Gram

householder, asked, indicted, there, ethnic
score, two, best, three, four
mining, footballer, population, laps, settled
density, census, fourier, editor, photos
money, toured, season, announced, banned

Rotated Skip-Gram

twitter, facebook, youtube, myspace, internet
receptors, receptor, neurons, apoptosis, neuronal
pennsylvania,ohio,maryland,philadelphia,illinois
paintings, portraits, painting, drawings, painter
that, which, when, where, but

Expressive Performance

Tasks

- Word Similarity (SimLex-999)

  Compare to human similarity evaluation of word pairs

- Semantic/Syntactic Analogies

  Predict D in {A : B = C : D} analogy problems

- Sentiment Analysis

  Predict Pos/Neg sentiment of input sentence, by averaging vectors

- Question Classification

  Classify question categories of TREC dataset

- Topic Classification

  Classify news topics of 20 newsgroups

- NP bracketing

  Classifying noun phrases in terms of bracketing: (A B) C or A (B C)

Expressive Performance

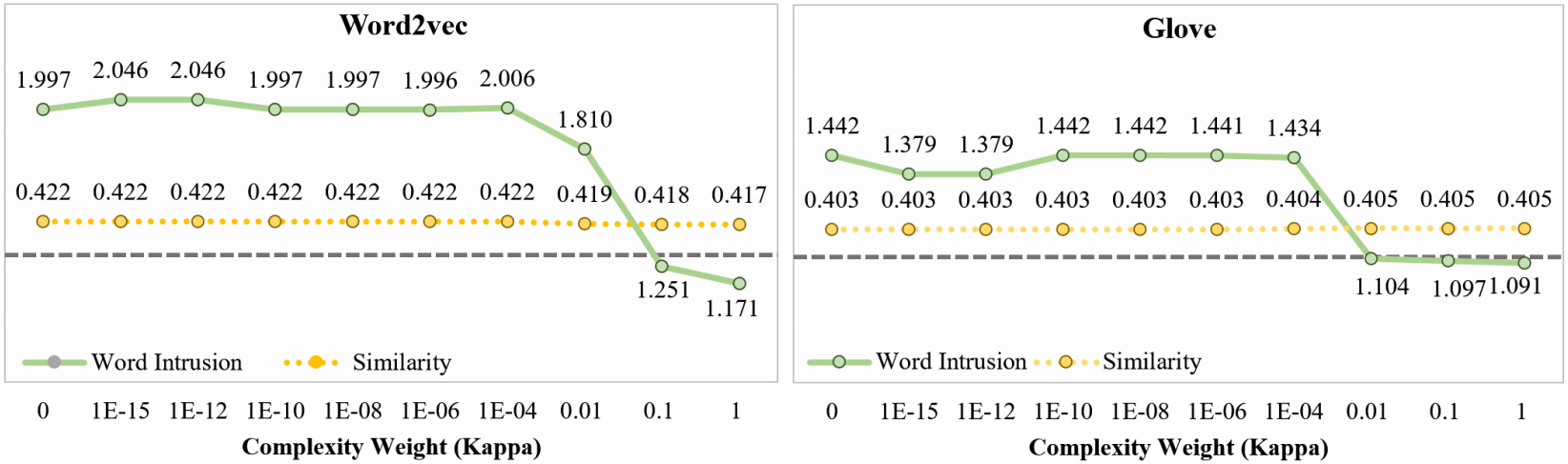
Results

	# dims	Simil.	Analg. (sem)	Analg. (syn)	Sent.	Ques.	Topics (Sp.)	NP brckt.
Skip-Gram	300	.374	.668	**.652**	.741	.920	**.960**	.812
SOV	3000	.390	.640	.594	.751	.910	.955	**.836**
SOV (non-neg)	3000	.384	.566	.480	**.761**	.918	**.960**	.829
Quartimax (orthogonal)	300	.374	.668	**.652**	.744	.922	.956	.822
Varimax (orthogonal)	300	.374	.668	**.652**	.744	.922	.956	.822
Parsimax (orthogonal)	300	.374	.668	**.652**	.744	.922	.956	.819
FacParsim (orthogonal)	300	.374	.668	**.652**	.744	.922	.956	.822
Quartimax (oblique)	300	**.422**	**.673**	.624	.755	**.932**	.955	.820
Varimax (oblique)	300	**.422**	**.673**	.624	.755	**.932**	.955	.820
Parsimax (oblique)	300	.421	.671	.623	.752	**.932**	.956	.826
FacParsim (oblique)	300	.417	.660	.620	.751	.928	.952	.820

- Preserves cosine distance (similarity) between word vectors
- Preserves expressive power of original word vectors

Weighting Complexities

Effect of kappa



- Large weights on column complexity may degrade interpretability
- Widely used criterion show effective improvement on interpretability

Discussion

Conclusion

- Rotation algorithms can improve interpretability of word vectors
- Can be widely applied to any pre-trained vectors, with preserving expressive performance

Future Work

- Useful in lexical semantics to explore compositionality of words
- Rotation algorithms can be applied to any layer in the neural networks for interpretability

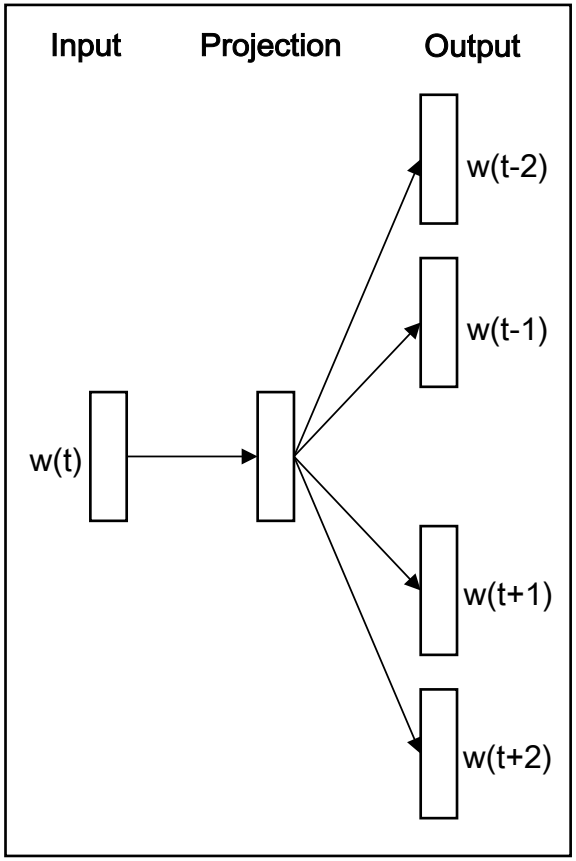
Subword-level Word Vector Representations for Korean Language

Sungjoon Park, Jeongmin Byun, Sion Baek, Yongseok Cho, Alice Oh

(Work in Progress)

Subword-level Word Vector Representations

SISG (Subword Information Skip-Gram, FastText) (Bojanowski et al., TACL 2017)



Objective Given a sequence of training words $w_{1\dots t}$

$$\frac{1}{V}\sum_{V}\sum_{-c\le j\le c, j\neq 0}\log p\left(w_{t+j}\middle|w_t\right)$$

where $p(w_O|w_I)$ is a softmax:

$$p(w_O|w_I)=\frac{\exp(u_{w_O}^T v_{w_I})}{\sum_w \exp(u_w^T v_{w_I})},\qquad s(w,c)=u_{w_O}^T v_{w_I}$$

computed by negative sampling.

Given a word w and denote $G_w=\{1,\dots,G\}$ the set of n-grams in w

$$s(w,c)=\sum_{g\in G_w} z_g^T v_c$$

That is, word vector $u_{w_O}^T$ is replaced to the sum of n-grams vectors

Subword-level Word Vector Representations

SISG (Subword Information Skip-Gram, FastText) (Bojanowski et al., TACL 2017)

- Word-level: Word2vec (Skip-Gram/CBOW)

  안녕하세요: 안녕하세요

- Character-level: FastText (3-6gram)

  <안녕하세요> : <안녕하, 안녕하, 녕하세, 하세요, 세요>

- Phoneme-level: FastText (3-6gram)

  <안녕하세요> : <ㅇㅏㄴㄴㅕㅇㅎㅏㅅㅔㅇㅛ>

Preliminary Results

SISG (Subword Information Skip-Gram, FastText) (Bojanowski et al., TACL 2017)

- Korean word similarity (WS-353): SISG outperforms SG in Korean as well

Hyperparams		Word-level		Syllable-level		Phoneme-level	
# dims	# neg. samples	type	cor	ngrams	cor	ngrams	cor
500	5	sg	0.542	36	0.552	**36**	**0.602**
500	15	sg	0.547	36	0.575	**36**	**0.607**
500	5	cbow	0.531	24	0.577	24	0.590
500	15	cbow	0.541	24	0.560	24	0.587
300	5	sg	**0.565**	36	0.564	**36**	**0.623**
300	15	sg	0.529	36	0.572	36	0.592
300	5	cbow	0.528	**24**	**0.600**	24	0.595
300	15	cbow	0.535	24	0.598	24	0.593
100	5	sg	0.522	36	0.554	36	0.585
100	15	sg	0.533	36	0.553	36	0.569
100	5	cbow	0.506	24	0.560	24	0.573
100	15	cbow	0.515	24	0.582	24	0.572
AVG			**0.533**		**0.571**		**0.591**

Thank you.