Word2vec Word Sense Disambiguation with Clustering

Gerome Yoo

Department of Statistics, Sunkyunkwan University

2019

INDEX

Word Embedding

Word Embedding

Collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers.

- *"Word Embedding"* from Wikipedia

$\Rightarrow$ **Simply, way of mapping vectors to words.**

INTRODUCTION

Word Embedding

- "Sparse representation" and "Dense Representation" are two categories of word embedding.
- The concept of word embedding itself only allow only one vector representation for one letter representation.
- Polysemy is problem of one word with multiple word senses.

Sparse Representation

$$\text{cat} \Rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \text{dog} \Rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \text{kitty} \Rightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

One-hot encoding vector

- Most simple way to convert word into a vector
  (1) Score the Words. Then,
  (2) Element of vector which stands for word becomes 1.
  (3) Element elsewhere become Zero.
- If there are N-word, vector will be N-dimensional with only one element with value of 1.

Sparse Representation

$$\text{cat} \Rightarrow \begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix}, \quad \text{dog} \Rightarrow \begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix}, \quad \text{kitty} \Rightarrow \begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix}$$

Drawback of One-hot encoding vector

- There is no difference between relationship of "cat and dog" and "cat and kitty"
- There supposed to be relation between "cat and kitty" but there is not, since **"Orthogonal"**.
- It can be concluded that, this embedding can not reflect relationship between words.

Dense Representations

Dense Representation

$$\text{cat} \Rightarrow \begin{bmatrix} 0.7 \\ -0.3 \\ -0.5 \\ 0.42 \\ 0.73 \end{bmatrix}, \quad \text{dog} \Rightarrow \begin{bmatrix} 0.65 \\ 0.4 \\ 0.5 \\ 0.3 \\ 0.71 \end{bmatrix}, \quad \text{kitty} \Rightarrow \begin{bmatrix} 0.7 \\ -0.34 \\ -0.45 \\ 0.4 \\ 0.34 \end{bmatrix}$$

- Represent the word in projection of N-dimension, where N is decided arbitrarily.
- "Cat and Kitty", in sparse, there are no relation. In dense, the distance between vectors will be very close.
- What exact attributes a dimension means can not be known.

WORD2VEC

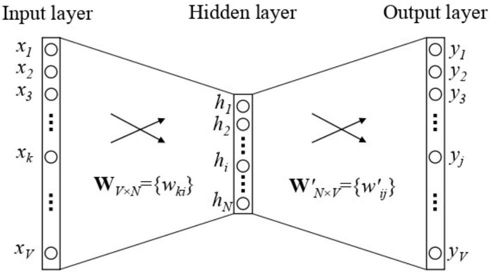

Figure 1: Undercomplete Autoencoder

- Word embedding can learn compressed information in lower dimension.
- One-hot encoding vector is used in input layer make possible to specify a column of the hidden layer as the word embedding.

WORD2VEC

Result of Word2Vec embedding

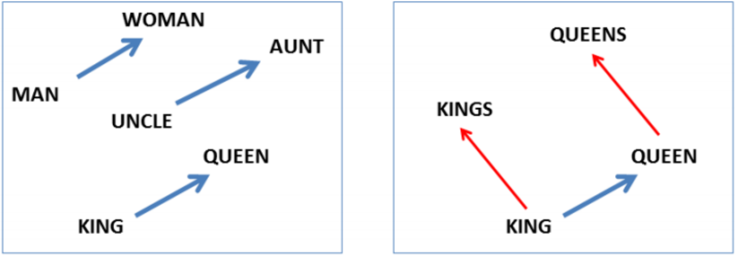

Figure 2: Word2Vec embeddings

- Korean word2vec

WORD2VEC

word2vec - 2 Models

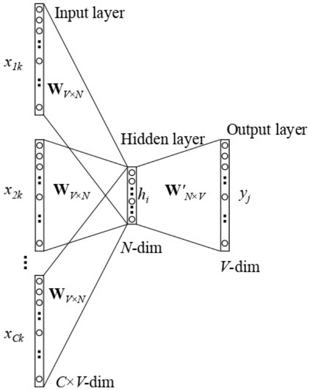

Figure 3: CBOW

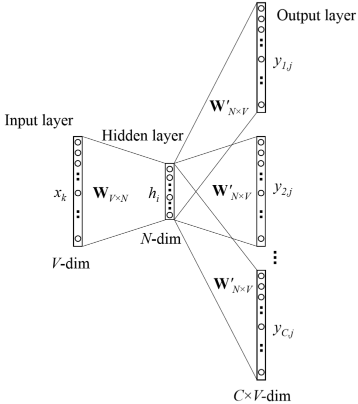

Figure 4: Skip-Gram

Word2Vec - Skip Gram

Skip-Gram

- CBOW puts target word at output layer and context words at input.
- Different from CBOW, target word set at the input layer and context words at output.
- Intuitively, SG sets target word to learn the context words.
- SG model generally shows better results at training the word embedding than CBOW.

MSSG MODEL

What is Word Sense?

- Word Sense is one of the meaning of a word.
- When obtaining information, word meaning need to be decided.
- Deciding the meaning of the word is called

Word Sense Disambiguation.

Example

- It is common practice in nuclear power **plants** to...
- ...**plants** to promote growth and increase yields.

MSSG MODEL

MSSG model structure

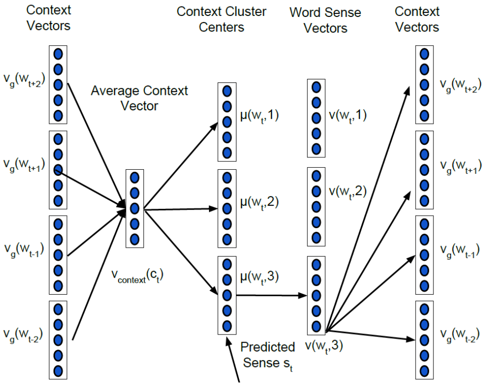

Figure 5: Architecture of MSSG model

MSSG MODEL

Capturing Multiple senses

- Creating Sense Vector

$$V_{sense}(C_t) = \frac{1}{2 \times R_t} \sum_{c \in c_t} V_g(C)$$

- C : the target word
- t : the position of target word in sentence
- R_t : size of window
- C_t : $\{W_{t-R_t}, ..., W_{t-1}, W_{t+1}, ..., W_{t+R_t}\}$
- W_i : i-th word in a sentence

MSSG MODEL

Modified Version for Sliding Window

- Creating Sense Vector

$$V_{sense}(C_t) = \frac{1}{\text{Cardinality}(C_t)} \sum_{c \in c_t} V_g(c)$$

- Context should not be arbitrarily truncated to match window size.
- Original equation does not reflect this property.

MSSG MODEL

Sliding window

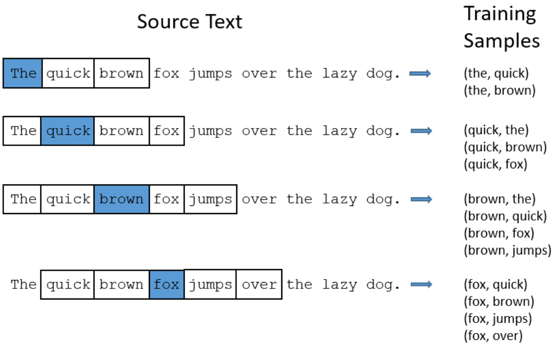

Figure 6: Basic Concepts of sliding window from word2vec tutorial by Chris McCormick

MSSG MODEL

Standard K-means

- Find $\mu_1, \cdots \mu_k$ for centroids of K-clusters by minimizing
- Minimizer

$$E = \frac{1}{N} \sum_{i=1}^{N} ||x_i - \mu_{k(i)}||^2$$

- N : total number of vectors
- Index of the closest cluster centroid to x

$$k(x) = \underset{k \in \{1, \cdots, k\}}{argmin} ||x - \mu_{k(x)}||$$

MSSG MODEL

k-means with cosine similarity

- Normalize each vectors to have unit length.
- Maximize

$$L = \sum_{i=1}^{N} x_i^T \mu_{k(i)}$$

- $\{\mu_1, \cdots \mu_k\}$: a set of unit-length centroid vectors
- Index of the closest cluster centroid to x

$$k(x) = \underset{k}{argmax}\, x^T \mu_{k(x)}$$

PROPOSAL

Weighting Context

- The vector of a term is weighted by frequency according to appearance frequency among the whole corpus at Skip-Gram.
- "...power plant near the forest"
  1 tree context and 1 power context
- "...plant habitat near nuclear facility"
  1 tree context and 2 power context
- Though, second context is tree-wise, it will be disambiguated as power

PROPOSAL

Weighting Context

- Weighting Context can be breakthrough.
- Sense vector mainly focus on the single specific term.
- Building sense vector for a specific term in a document should be different from building global vector for a term in the corpus.
- Assumption that more related word should appears closer in sentences.

Proposal

Weighting Context

- Ordinary

1	1	1	1	1	Target	1	1	1	1	1

Figure 7: Ordinary Weighting

- Proposal

1	2	3	4	5	Target	5	4	3	2	1

Figure 8: Proposed Weighting

- Precise Expression

$$weight_j = d + 1 - |j|$$
$$\text{where } j = -R_t, -R_t + 1, \cdots, -1, 1, \cdots, R_t - 1, R_t$$

- Various combinations of weighting method can be applied other than this.

Proposal

Weighting Context

- Equation Update to

$$\frac{1}{\sum_j weight_j} \sum_j weight_j \times V_g(W_{t-j})$$

$$\text{where } C_t = \{W_{t-R_t}, ..., W_{t-1}, W_{t+1}, ..., W_{t+R_t}\}$$

$$\text{and } j = -R_t, -R_t + 1, \cdots, -1, 1, \cdots, R_t - 1, R_t.$$

Clustering

- Other than K-means, hierarchical clustering will be compared.

Proposal

Distance

- A and B are two vectors of elements where A_i and B_i are each components of vector A and B.
- Euclidean Distance

$$d(A,B) = ||A - B||_2 = \sqrt{\sum_i (A_i - B_i)^2}$$

- Cosine Distance

$$d(A,B) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_i A_i B_i}{\sqrt{\sum_i A_i^2}\sqrt{\sum_i B_i^2}}$$

Proposal

Linkage

- Single Linkage

$$min\{d(a,b) : a \in A, b \in B\}$$

- Average Linkage

$$\frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} d(a,b)$$

- Complete Linkage

$$max\{d(a,b) : a \in A, b \in B\}$$

SIMULATION

Simulation Study

Detect Word Sense in word 'plant'

Data Introduction

- 884 Abstracts from Journal of Statistical Software(JSS).
  4 abstracts containing tree-wise 'plant'
- 10 abstracts from power-wise papers containing 'plant'.
- 10 abstracts from tree-wise papers containing 'plant'.

Simulation

Corpus

- Word plant Occurrence : 70
- Stemming by Porter Stemmer.

Skip-Gram Training Parameter

- Dimension : 100 $\sim$ 500 by 50
- Window : 5
- Min count of word : 1, 2, 3
- No StopWords Deletion
- 500 Iteration

Simulation

Corpus after Preprocess

Minword	Corpus Total	Corpus Retain	Corpus Ratio	Vocabulary Total	Vocabulary Retain	Vocabulary Ratio
Min 1	118816	118816	100%	6803	6803	100%
Min 2	118816	115762	97%	6803	3749	55%
Min 3	118816	113784	95%	6803	2760	40%
Min 5	118816	110835	93%	6803	1900	27%

Table 1: Corpus Vocabulary comparison with Minword

- Minword means that condition of minimum word frequency required to be remained in the model.
- Set to 2, 45% of the vocabularies is gone though the total word frequency decreased by 3%.
- Abstracts use lots of unique words so proper consideration of this property is needed at analyzing the results.

Word Vector

Vectorization Result

	0	1	2	3	4	5	6	7	8	9	...	190
the	-0.005274	-0.004881	0.008978	0.016738	-0.196532	0.039860	-0.008283	0.057658	-0.013314	0.038370	...	-0.056950
of	-0.046892	0.078688	0.009536	0.001345	-0.150550	0.070457	0.033300	-0.055566	0.050849	-0.038141	...	0.091164
and	-0.013463	-0.099499	0.057102	0.047343	-0.056065	0.034342	-0.037420	-0.044739	-0.029994	0.022349	...	-0.111621
a	0.088975	-0.026309	-0.010853	-0.052011	0.015842	0.121120	-0.012029	-0.061442	0.029275	0.061125	...	-0.051282
to	-0.001451	-0.096425	-0.050938	0.123212	-0.034227	0.015846	0.039572	-0.032654	0.065712	0.057349	...	0.020379
in	-0.010377	-0.155409	0.080253	-0.034712	-0.024648	0.038298	0.070546	0.062922	0.076265	0.131673	...	-0.002128
for	-0.014069	0.072710	-0.002113	0.022438	-0.108717	0.017320	-0.026740	-0.092279	-0.001959	0.007090	...	-0.031778
model	0.059081	-0.002837	0.087326	-0.148797	0.010424	0.093209	0.036014	0.058492	0.017576	-0.011537	...	-0.053278
is	0.032415	0.009137	0.043433	-0.025170	-0.133684	0.002041	0.062189	-0.048085	0.012151	-0.076218	...	-0.024664
packag	-0.011378	0.042784	-0.015387	0.050004	-0.100221	-0.027605	-0.023230	0.132628	0.022325	0.110128	...	-0.016424
use	-0.003510	-0.051181	-0.003584	-0.038074	0.051430	-0.022697	-0.086584	0.010690	0.005365	0.065578	...	-0.098157
data	-0.062955	-0.016199	0.081558	-0.051680	-0.094418	0.056436	0.189726	0.036928	0.093235	0.057148	...	-0.073167
are	0.027333	0.062075	0.013693	-0.072706	-0.026380	0.030791	0.026584	0.014936	-0.112236	-0.132439	...	0.001078
r	0.041546	0.049575	0.045000	-0.110312	-0.037867	-0.004644	-0.033628	0.128619	0.076034	0.028486	...	-0.036084
thi	0.026243	0.016053	-0.007376	-0.104401	-0.066868	-0.063286	0.082070	-0.009953	0.043020	0.083790	...	-0.034547
with	-0.014180	0.031629	0.002237	0.018051	-0.054588	0.161907	0.020293	-0.024866	0.027719	0.140517	...	-0.001839
as	0.025814	0.055456	-0.024011	-0.091078	-0.028687	0.026749	-0.116749	-0.094028	0.105277	0.074279	...	-0.150292
that	-0.027353	0.024537	-0.016085	-0.046037	-0.092417	-0.011636	-0.014884	-0.007575	0.047328	0.003559	...	-0.000196
be	-0.022672	-0.043258	-0.018999	-0.070277	0.031477	0.011959	0.043153	-0.062431	0.070694	0.014364	...	0.029787
we	-0.024113	0.105714	-0.049783	0.054866	0.026242	-0.036484	-0.077901	-0.047014	0.039281	0.000504	...	-0.028556
on	-0.114511	-0.080015	-0.007271	0.042869	-0.021789	0.075222	0.048453	-0.074878	0.097548	-0.045232	...	-0.046449
method	0.035439	0.085446	0.031535	0.065097	-0.137242	0.034622	0.024615	0.044898	0.030258	0.057919	...	-0.003838
an	0.073803	-0.031584	0.087269	-0.025697	-0.098544	-0.081930	-0.012254	0.011796	0.027309	0.066877	...	0.014081

Figure 9: Resulted Word Vector

Analysis

Clustering Result Minword 1

Minword 1	k-means Clustering			
Distance	Euclidean		Cosine	
Dim	Ordinary	Proposal	Ordinary	Proposal
100	75.71%	**84.29%**	78.57%	**82.86%**
150	82.86%	**85.71%**	80.00%	**81.43%**
200	82.86%	**84.29%**	78.57%	**81.43%**
250	82.86%	**85.71%**	80.00%	**82.86%**
300	84.29%	84.29%	80.00%	**82.86%**
350	85.71%	84.29%	80.00%	**82.86%**
400	82.86%	**84.29%**	80.00%	**82.86%**
450	84.29%	84.29%	80.00%	**82.86%**
500	84.51%	81.43%	80.00%	**82.86%**

Table 2: k-means Clustering Accuracy with Minword 1

Analysis

Clustering Result Minword 1

Minword 1	Euclidean Hierarchical Clustering					
Linkage	Single		Average		Complete	
Dim	Ordinary	Proposal	Ordinary	Proposal	Ordinary	Proposal
100	52.86%	**52.86**%	61.43%	61.43%	64.29%	**65.71**%
150	52.86%	52.86%	54.29%	52.86%	71.43%	**74.29**%
200	54.29%	52.86%	54.29%	54.29%	62.86%	**71.43**%
250	54.29%	52.86%	54.29%	52.86%	61.43%	**70.00**%
300	54.29%	52.86%	54.29%	54.29%	67.14%	67.14%
350	52.86%	**54.29**%	54.29%	54.29%	60.00%	**72.86**%
400	52.86%	52.86%	54.29%	54.29%	60.00%	**70.00**%
450	54.29%	54.29%	54.29%	54.29%	67.14%	**71.43**%
500	52.86%	**54.29**%	54.29%	54.29%	72.46%	71.43%

Table 3: Euclidean Hierarchical Clustering Accuracy with Minword 1

Analysis

Clustering Result Minword 1

Minword 1	Cosine Hierarchical Clustering					
Linkage	Single		Average		Complete	
Dim	Ordinary	Proposal	Ordinary	Proposal	Ordinary	Proposal
100	52.86%	52.86%	52.86%	**57.14**%	60.00%	**61.43**%
150	51.43%	51.43%	51.43%	**57.14**%	78.57%	62.86%
200	52.86%	52.86%	51.43%	**58.57**%	75.71%	**87.14**%
250	51.43%	51.43%	51.43%	51.43%	81.43%	54.29%
300	51.43%	51.43%	54.29%	51.43%	71.60%	**87.14**%
350	51.43%	**52.86**%	54.29%	51.43%	87.14%	71.43%
400	51.43%	51.43%	54.29%	51.43%	80.00%	**90.00**%
450	51.43%	**52.86**%	54.29%	51.43%	62.86%	**88.57**%
500	51.43%	**52.86**%	54.29%	51.43%	51.43%	**90.00**%

Table 4: Cosine Hierarchical Clustering Accuracy with Minword 1

Analysis

minword 1

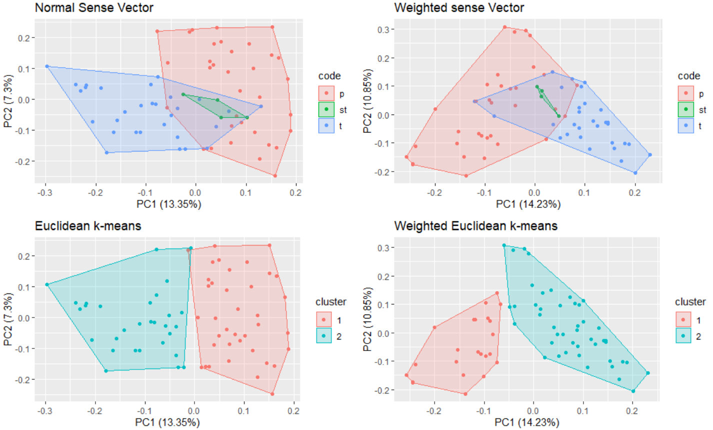

Figure 10: Clustering Visualization with Minword 1, dim 100

ANALYSIS

minword 1

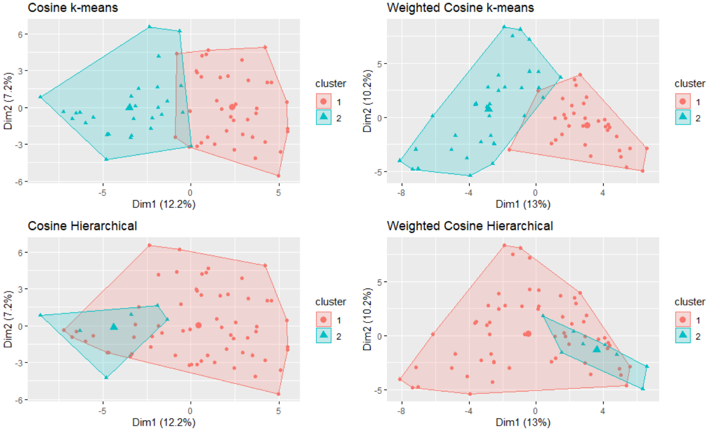

Figure 11: Clustering Visualization with Minword 1, dim 100

Analysis

minword 1

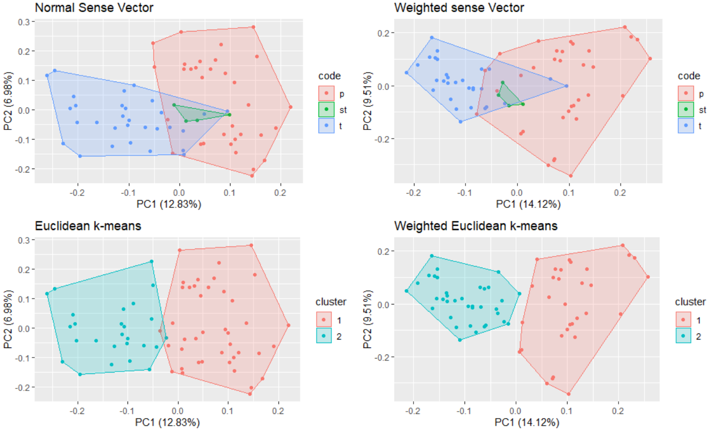

Figure 12: Clustering Visualization with Minword 1, dim 300

Analysis

minword 1

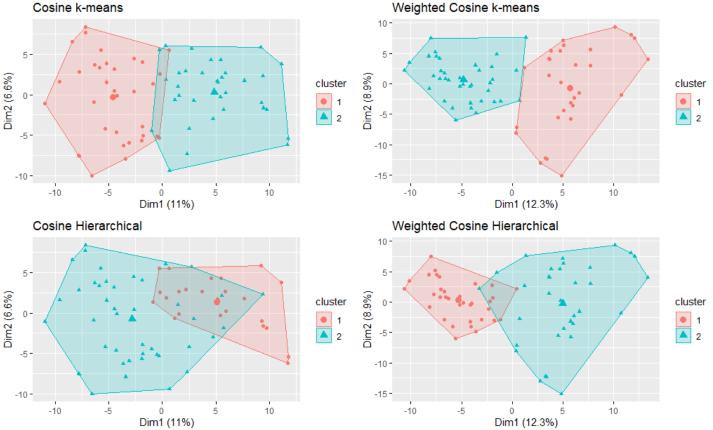

Figure 13: Clustering Visualization with Minword 1, dim 300

Analysis

minword 1

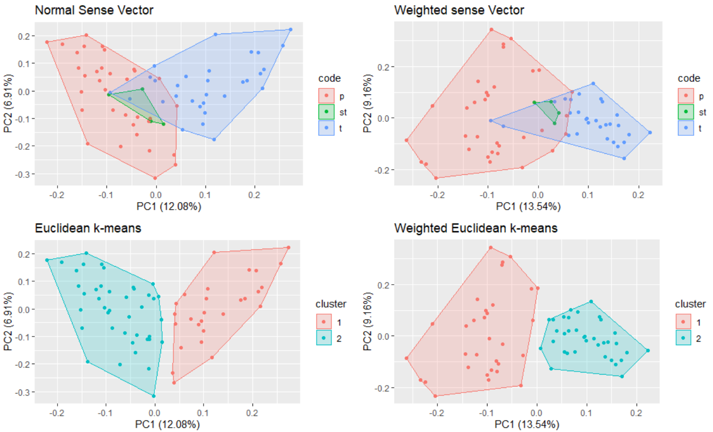

Figure 14: Clustering Visualization with Minword 1, dim 500

Analysis

minword 1

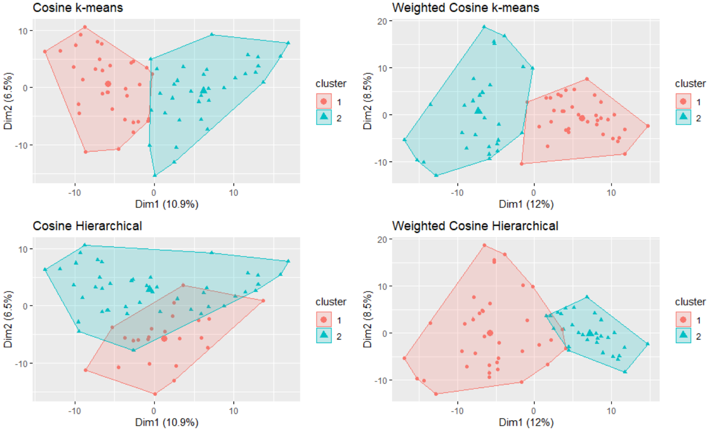

Figure 15: Clustering Visualization with Minword 1, dim 500

Analysis

minword 1

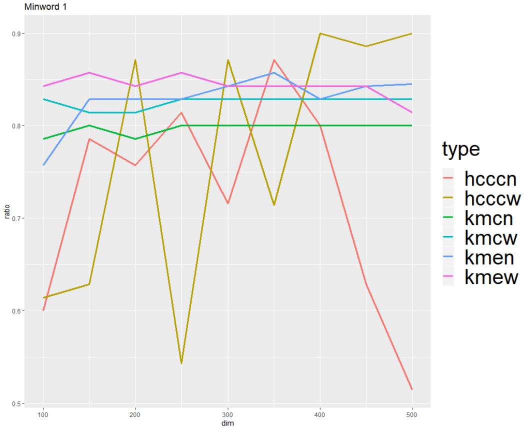

Figure 16: minword 1 comparison

Analysis

- hcccn = Hiearachical Clustering Cosines Normal
- hcccw = Hiearachical Clustering Cosines Weight
- kmcn = K-means Cosine Normal
- kmcw = K-means Cosine Weight
- kmen = K-means Euclidean Normal
- kmew = K-means Euclidean Weight

Analysis

Clustering Result Minword 2

Minword 2	k-means Clustering			
Distance	Euclidean		Cosine	
Dim	Ordinary	Proposal	Ordinary	Proposal
100	80.00%	71.43%	64.29%	**80.00%**
150	77.14%	**82.86%**	65.71%	**75.71%**
200	81.43%	**84.29%**	65.71%	**81.43%**
250	83.33%	74.29%	81.43%	**82.86%**
300	82.86%	75.71%	70.00%	**82.86%**
350	74.29%	74.29%	78.57%	74.29%
400	74.29%	**81.43%**	78.57%	74.29%
450	84.29%	84.29%	78.57%	**82.86%**
500	84.29%	84.29%	78.57%	**82.86%**

Table 5: k-means Clustering Accuracy with Minword 2

Analysis

Clustering Result Minword 2

Minword 2	Euclidean Hierarchical Clustering					
Linkage	Single		Average		Complete	
Dim	Ordinary	Proposal	Ordinary	Proposal	Ordinary	Proposal
100	52.86%	52.86%	54.29%	54.29%	68.57%	**75.71**%
150	52.86%	52.86%	54.29%	52.86%	54.29%	**55.71**%
200	52.86%	52.86%	54.29%	54.29%	54.29%	**57.14**%
250	54.29%	52.86%	54.29%	54.29%	54.29%	**81.43**%
300	52.86%	52.86%	54.29%	54.29%	74.29%	**78.57**%
350	52.86%	52.86%	54.29%	54.29%	67.14%	51.43%
400	52.86%	52.86%	54.29%	54.29%	62.86%	**85.71**%
450	52.86%	52.86%	54.29%	54.29%	54.29%	**55.71**%
500	52.86%	52.86%	54.29%	54.29%	61.43%	58.57%

Table 6: Euclidean Hierarchical Clustering Accuracy with Minword 2

Analysis

Clustering Result Minword 2

Minword 2 Linkage	Cosine Hierarchical Clustering Single		Average		Complete	
Dim	Ordinary	Proposal	Ordinary	Proposal	Ordinary	Proposal
100	51.43%	**52.86**%	52.86%	**57.14**%	65.71%	**81.43**%
150	52.86%	52.86%	57.14%	55.71%	57.14%	57.14%
200	52.86%	52.86%	55.71%	**57.14**%	75.71%	61.43%
250	52.86%	52.86%	57.14%	54.29%	77.14%	71.43%
300	52.86%	52.86%	57.14%	51.43%	77.14%	62.86%
350	52.86%	52.86%	52.86%	51.43%	75.71%	**82.86**%
400	51.43%	**52.86**%	52.86%	52.86%	75.71%	62.86%
450	52.86%	52.86%	54.29%	**61.43**%	77.14%	62.86%
500	52.86%	52.86%	54.29%	52.86%	80.00%	62.86%

Table 7: Cosine Hierarchical Clustering Accuracy with Minword 2

Analysis

minword 2

Figure 17: Clustering Visualization with Minword 2, dim 100

ANALYSIS

minword 2

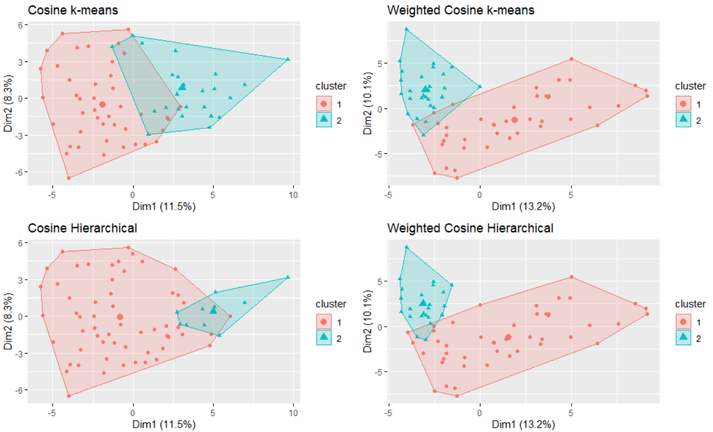

Figure 18: Clustering Visualization with Minword 2, dim 100

Analysis

minword 2

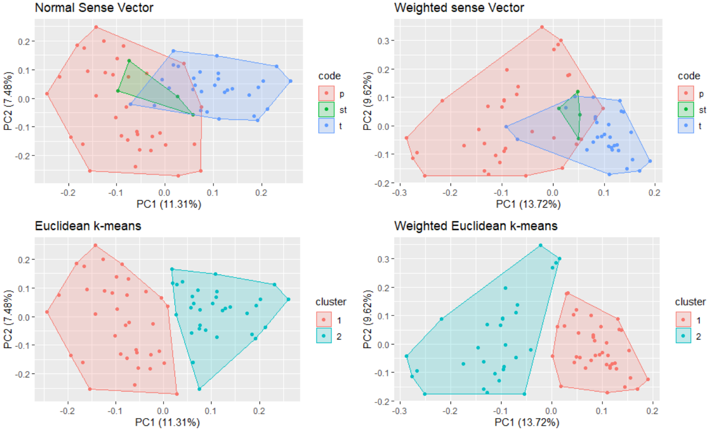

Figure 19: Clustering Visualization with Minword 2, dim 300

Analysis

minword 2

Figure 20: Clustering Visualization with Minword 2, dim 300

Analysis

minword 2

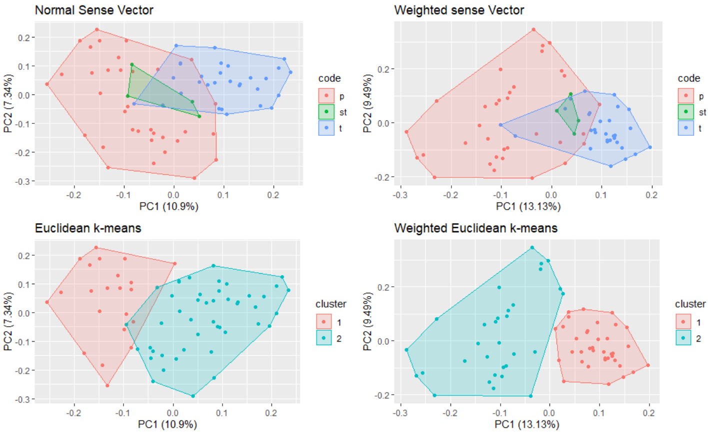

Figure 21: Clustering Visualization with Minword 2, dim 500

Analysis

minword 2

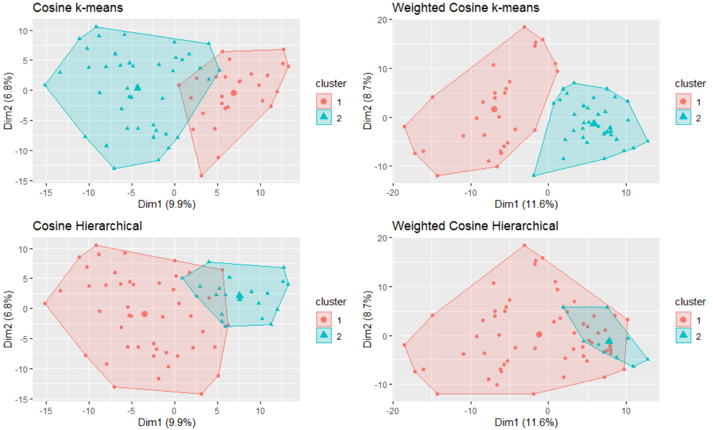

Figure 22: Clustering Visualization with Minword 2, dim 500

Analysis

minword 2

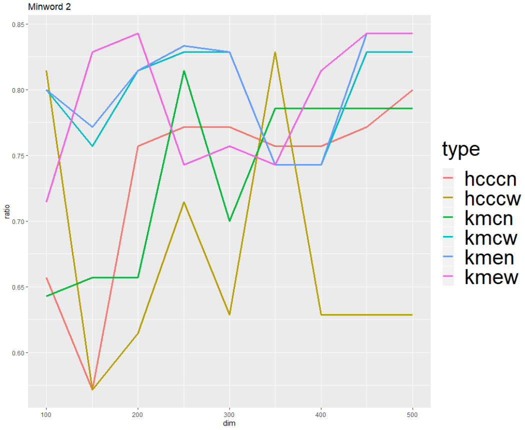

Figure 23: minword 2 comparison

ANALYSIS

- hcccn = Hiearachical Clustering Cosines Normal
- hcccw = Hiearachical Clustering Cosines Weight
- kmcn = K-means Cosine Normal
- kmcw = K-means Cosine Weight
- kmen = K-means Euclidean Normal
- kmew = K-means Euclidean Weight

Analysis

Clustering Result Minword 3

Minword 3	k-means Clustering			
Distance	Euclidean		Cosine	
Dim	Ordinary	Proposal	Ordinary	Proposal
100	77.14%	72.86%	52.86%	**68.57%**
150	68.57%	**84.29%**	57.14%	**71.43%**
200	67.14%	**75.71%**	57.14%	**72.86%**
250	68.57%	**82.86%**	61.43%	**80.00%**
300	71.43%	**81.43%**	64.29%	**81.43%**
350	77.14%	75.71%	71.43%	**74.29%**
400	67.14%	**75.71%**	64.29%	**82.86%**
450	72.86%	**81.43%**	70.00%	**82.86%**
500	51.43%	**75.71%**	67.14%	**80.00%**

Table 8: k-means Clustering Accuracy with Minword 3

Analysis

Clustering Result Minword 3

Minword 3	Euclidean Hierarchical Clustering					
Linkage	Single		Average		Complete	
Dim	Ordinary	Proposal	Ordinary	Proposal	Ordinary	Proposal
100	52.86%	52.86%	54.29%	52.86%	54.29%	54.29%
150	54.29%	52.86%	54.29%	54.29%	58.57%	**62.86**%
200	52.86%	52.86%	54.29%	54.29%	54.29%	**61.43**%
250	52.86%	52.86%	54.29%	54.29%	62.86%	60.00%
300	52.86%	52.86%	54.29%	54.29%	60.00%	**61.43**%
350	52.86%	52.86%	54.29%	54.29%	55.71%	**57.14**%
400	52.86%	52.86%	54.29%	54.29%	78.57%	51.43%
450	52.86%	52.86%	54.29%	54.29%	60.00%	55.71%
500	52.86%	52.86%	54.29%	54.29%	62.86%	54.29%

Table 9: Euclidean Hierarchical Clustering Accuracy with Minword 3

Analysis

Clustering Result Minword 3

Minword 3	Cosine Hierarchical Clustering					
Linkage	Single		Average		Complete	
Dim	Ordinary	Proposal	Ordinary	Proposal	Ordinary	Proposal
100	54.29%	52.86%	54.29%	52.86%	58.57%	**74.29**%
150	52.86%	52.86%	55.71%	55.71%	68.57%	55.71%
200	54.29%	52.86%	55.71%	55.71%	64.29%	**74.29**%
250	51.43%	**52.86**%	57.14%	**60.00**%	64.29%	**78.57**%
300	52.86%	52.86%	57.14%	51.43%	64.29%	**78.57**%
350	52.86%	52.86%	57.14%	52.86%	67.14%	62.86%
400	52.86%	52.86%	57.14%	54.29%	68.57%	62.86%
450	52.86%	52.86%	57.14%	52.86%	64.29%	62.86%
500	52.86%	52.86%	54.29%	**55.71**%	67.14%	**80.00**%

Table 10: Cosine Hierarchical Clustering Accuracy with Minword 3

Analysis

minword 3

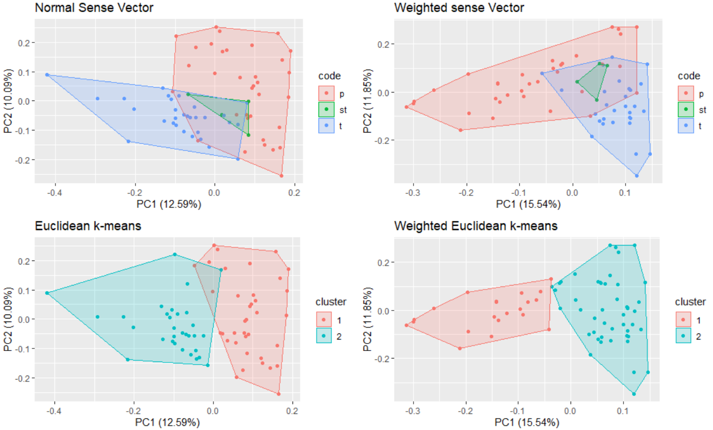

Figure 24: Clustering Visualization with Minword 3, dim 100

Analysis

minword 3

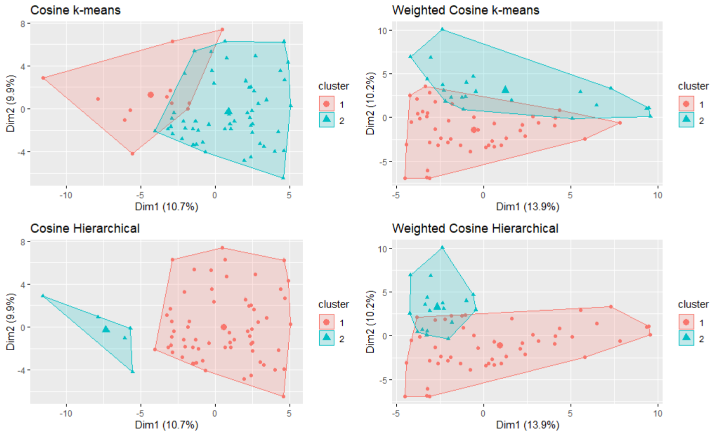

Figure 25: Clustering Visualization with Minword 3, dim 100

Analysis

minword 3

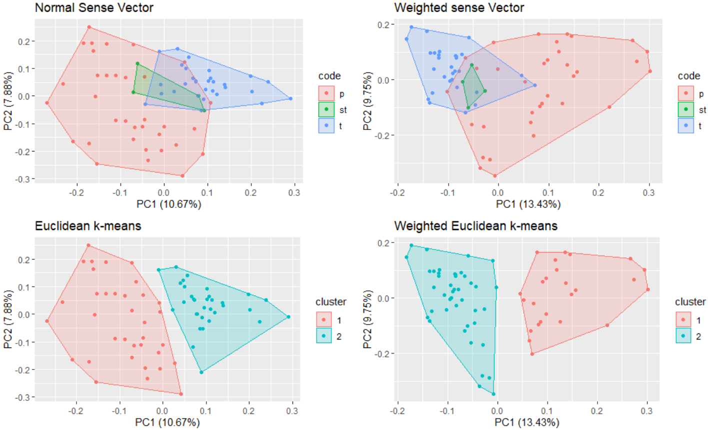

Figure 26: Clustering Visualization with Minword 3, dim 300

Analysis

minword 3

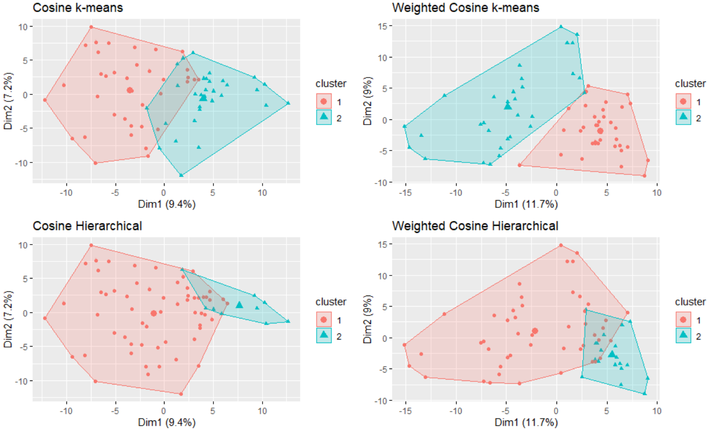

Figure 27: Clustering Visualization with Minword 3, dim 300

Analysis

minword 3

Figure 28: Clustering Visualization with Minword 3, dim 500

Analysis

minword 3

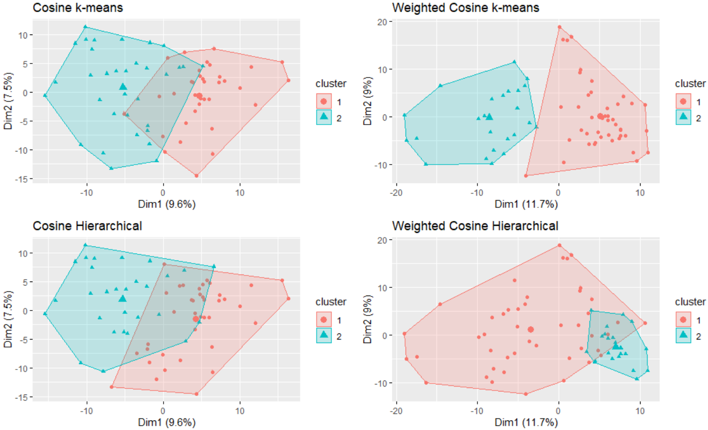

Figure 29: Clustering Visualization with Minword 3, dim 500

Analysis

minword 3

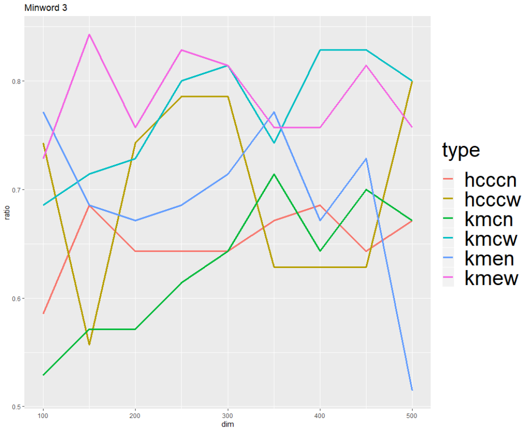

Figure 30: minword 3 comparison

Analysis

- hcccn = Hiearachical Clustering Cosines Normal
- hcccw = Hiearachical Clustering Cosines Weight
- kmcn = K-means Cosine Normal
- kmcw = K-means Cosine Weight
- kmen = K-means Euclidean Normal
- kmew = K-means Euclidean Weight

Conclusion

- Weighting context generally advance the performance of the clustering method.
- Hierarchical clustering does not give evidence to replace the k-means clustering method.
- In either method, using weighting method is generally make performance better than the previous model.
- Applying the weight method to other corpus that are bigger and general needs to be done for further investigation.

Appendix

Skip-Gram (Mikolov 2013)

- given a pair of words (w_t,c), the probability that word c is observed in the context of target word w_t is

$$\begin{aligned} P(D=1|v(w_t),v(c)) &= P(\text{observing } v(c)|v(w_t)) \\ &= \frac{exp^{v(w_t)^T v(c)}}{\sum exp^{v(w_t)^T v(c)}} \end{aligned}$$

- the probability of not observing word c in the context of target word w_t is

$$\begin{aligned} P(D=0|v(w_t),v(c)) &= P(\text{not observing } v(c)|v(w_t)) \\ &= 1 - P(D=1|v(w_t),v(c)) \end{aligned}$$

APPENDIX

- word embeddings are learned by maximizing the objective function:

$$J(\theta) = \sum_{(w_t,c_t)\in D^+} \sum_{c \in c_t} logP(D = 1|v(w_t), v(c)) + \sum_{(w_t,c_t)\in D^-} \sum_{c \in c_t} logP(D = 0|v(w_t), v(c))$$

where c'_t is randomly sampled noisy context words for word w_t.

After training, weight matrix for corpus is obtained.

Appendix

Cosine Distance Matrix

- Cosine Distance of Word vectors for clustering.
- So produced the Cosine Distance Matrix using R package 'proxy'.

	X1	X2	X3	X4	X5	$\cdots$
X1	0.00	0.33	0.36	0.35	0.44	$\cdots$
X2	0.33	0.00	0.50	0.41	0.50	$\cdots$
X3	0.36	0.50	0.00	0.47	0.52	$\cdots$
X4	0.35	0.41	0.47	0.00	0.60	$\cdots$
X5	0.44	0.50	0.52	0.60	0.00	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

REFERENCE

- Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." arXiv preprint arXiv:1301.3781 (2013)
- Xin Rong. "word2vec Parameter Learning Explained." arxiv preprint arXiv:1411.2738 (2014)
- Neelakantan, Arvind, et al. "Efficient Non-parametric Estimation of Multiple Embeddings perWord in Vector Space." arXiv preprint arXiv:1504.06654 (2015)

Reference

- Hornik, K., Gr¨un, B. An r package for fitting topic models. *Journal of Statistical Software, 40(13), 1-30. Retrieved from* `https://epub .wu.ac.at/3987/`
- *김인영. "다의어 분석을 위한 군집화 방법." Master Thesis (2018)*
- *Thanaki, Jalaj. "파이썬 자연어 처리의 이론과 실제 : 효율적인 자연어 처리를 위한 머신 러닝과 딥러닝 구현하기." 서울, 에이콘 (2018)*