The Korean Journal of Applied Statistics
2021, Vol. 34, No. 3, 389–399

DOI: https://doi.org/10.5351/KJAS.2021.34.3.389
Print ISSN1225-066X / Online ISSN 2383-5818

Modified multi-sense skip-gram using weighted context and x-means

Hyunwoo Jeong[a], Eun Ryung Lee[1,a]

[a]Department of Statistics, Sungkyunkwan University

Abstract

In recent years, word embedding has been a popular field of natural language processing research and a skip-gram has become one successful word embedding method. It assigns a word embedding vector to each word using contexts, which provides an effective way to analyze text data. However, due to the limitation of vector space model, primary word embedding methods assume that every word only have a single meaning. As one faces multi-sense words, that is, words with more than one meaning, in reality, Neelakantan (2014) proposed a multi-sense skip-gram (MSSG) to find embedding vectors corresponding to the each senses of a multi-sense word using a clustering method. In this paper, we propose a modified method of the MSSG to improve statistical accuracy. Moreover, we propose a data-adaptive choice of the number of clusters, that is, the number of meanings for a multi-sense word. Some numerical evidence is given by conducting real data-based simulations.

Keywords: word embedding, skip-gram, multi-sense word, multi-sense skip-gram, X-means clustering, weighted context vector

1. 서론

분산 단어 표현(distributed representation of words)은 대량의 텍스트 자료인 말뭉치(corpus)에서 특정 단어 주위에 발생한 주변단어(context)들을 참고하여 단어를 표현하는 방법으로 여러 자연어 처리 작업에서 눈에 띌만한 성능을 달성했다 (Mikolov 등, 2013a, 2013b). 문장에서 유사한 의미나, 구문의 역할을 가진 단어를 서로 가까이 배치하기 때문에 분산 단어 표현 방법은 텍스트 자료의 훈련과정(training)을 통해 좀 더 정확한 단어 표현을 효과적으로 찾는 데 도움을 준다고 알려져 있다. 문장에서 단어의 구문적 의미를 정밀하게 분석하기 위해서 단어 임베딩 벡터를 정확하게 표현하는 것은 자연어 처리의 핵심이다. 이때 단어를 벡터로 표현하는 것을 임베딩이라고 하고, 이러한 방법으로 얻어진 벡터를 임베딩 벡터라고 한다. 하지만 분산 단어 표현을 활용에 관한 최근의 기술적 발전에도 불구하고, 이 방법들은 각 단어가 단일 의미(single-sense) 를 가지고 있다고 가정하므로 모든 단어가 하나의 벡터 표현만을 갖게 되는 제약을 가지고 있다. 따라서, 텍스트 자료에서 흔히 볼 수 있는 다의어 혹은 동음이의어의 경우 단어 의미들을 구별하지 못한다는 문제가 발생할 수 있다.

예를 들어, ‘plant’라는 단어의 경우 식물이라는 의미와 ‘power plant’와 같은 단어 사용에서 볼 수 있듯이 공장 및 시설 이라는 문맥적으로 다른 두 가지의 의미를 가지고 있다. 이러한 단어의 단일 의미 분산 표현

[1] Corresponding author: Department of Statistics, Sungkyunkwan University, 25-2, Sungkyunkwan-ro, Jongno-gu Seoul 03063, Korea. Email: erlee@skku.edu

Published 30 June 2021 / journal homepage: http://kjas.or.kr

방법은 대략적으로 해당 의미들에 대한 평균 임베딩 벡터 값을 가질 것이며 이 두 의미들을 정확히 구별하기 어려운 문제가 있다. 따라서, 다의성을 고려할 수 있는 분산 단어표현 방법이 필요하며 다의어의 각 의미에 대해 임베딩 벡터들을 할당할 수 있는 다중 임베딩 방법이 제안되었다. (Huang 등, 2012; Neelakantan 등, 2014) 다시 말해, 한 단어에 대해 여러 개의 벡터를 할당하여 의미 구별을 가능하게 한다. 다중 임베딩 방법은 같은 단어지만 단어의 주변단어들 분포가 달라진다는 점을 활용하여 문맥벡터, 즉, 해당 단어의 주변 단어들의 임베딩 값들의 평균을 구성한다. 또한, 해당 다의어 의미들을 구별하기 위해, 단어가 발생한 모든 문장의 문맥벡터에 대한 군집화를 수행하는 방법을 주로 사용한다.

이러한 방법들에서 단어들의 의미 임베딩 값들이 해당 의미를 정확히 담고 있는지는 방법의 성능을 결정하는 중요한 문제이다. 이 때 임베딩 벡터의 차원을 증가시켜 더 높은 차원에서 단어 의미를 예측하거나, 딥러닝(deep-learning) 모형 등을 활용하여 임베딩의 정확성을 높일 수 있다. 딥러닝 모형을 활용하여 텍스트 단어의 분산 표현으로부터 정확한 단어 임베딩 시스템을 훈련하는 skip-gram 모형을 Mikolov 등 (2013b)이 제안하여 아주 큰 주목을 받았으며, 다의어를 고려하여 다중 임베딩을 가능하게 하는 multi-sense skip-gram (MSSG) 또한 Neelakantan 등 (2014) 제안되었다. MSSG는 단어가 발생한 문장에서 문맥 벡터들을 모두 추출하여 K-means 군집 방법을 통해 군집화 하여 해당단어의 의미를 구분짓게 된다. MSSG는 K-means 군집 방법을 활용하기 때문에 군집의 수인 K를 사용자가 직접 정해야하는 어려움이 있다. 무엇보다 모든 다의어들의 의미 갯수를 K 개로 고정하기 때문에, 단어들의 의미 수에 대한 변동성을 가질 수 없는 단점 또한 존재한다. 따라서, 단어에 따라 단어의 의미 갯수를 자동적으로 결정하는 다중 임베딩 방법을 개발하는 것은 중요한 과제이다.

본 논문에서는 기존의 MSSG 방법을 수정하여 정확도를 높이는 방법들을 제안하려고 한다. 먼저, K-means 대신 Dan과 Moore (2000)이 제안한 X-means 군집 방법을 적용하여 문맥 벡터들에 대한 군집화를 수행함과 동시에 군집의 개수, 즉, 다의어의 의미 갯수를 데이터로부터 자동적으로 추정해주는 다중 임베딩 방법을 제안한다. 추가적으로, 주변 단어들에서 해당 단어에 가까울 수록 더 정확한 정보를 가지고 있는 경향이 있다는 경험적 사실을 활용하여 위치에 대한 중요도를 달리하여 가중 문맥 벡터(weighted context vector)를 구성하는 방법을 제안한다. 실증 예제를 이용한 모의실험들을 수행하여 본 논문에서 제안한 방법이 기존 방법들에 비해 더 정확하고 우수한 성능을 보임을 입증한다.

2. 기존 방법 설명

이 절에서는 본 논문에서 핵심적으로 사용되는 기존 방법들인 skip-gram (Mikolov 등, 2013b)과 multi-sense skip-gram (MSSG) (Neelakantan 등, 2014) 방법들을 소개하고자 한다.

2.1. Skip-gram 방법

Mikolov 등 (2013)이 제안한 단어의 분산 표현을 사용하여 단어 임베딩을 수행하는 skip-gram 방법을 소개하겠다. 이 방법은 텍스트 자료에서 얻은 말뭉치에 대해 수행하여, 말뭉치에서 발생한 모든 단어들에 대해 딥러닝 모형을 활용하여 문장의 주변 단어를 예측하여 단어 임베딩 벡터를 학습한다. Skip-gram모델의 구조는 Figure 1의 왼쪽 패널과 같이 주어진다. Skip-gram을 이용하여 얻은 단어 w에 대한 단어 임베딩 벡터는 $v(w) \in \mathbb{R}^d$라 표시하며, 여기서 d는 임베딩의 크기가 된다. 더 자세히 말하자면, 두 단어의 쌍 (w_t, c_j)이 주어졌을 때 단어 w_t의 주변, 즉, 문맥 (context) 에서 단어 c_j가 관측될 확률은 다음과 같다.

$$P(D = 1|v(w_t), v(c_j)) = \frac{e^{v(c_j)^\top v(w_t)}}{\sum_{j'=1}^{V} e^{v(c_{j'})^\top v(w_t)}}. \tag{2.1}$$

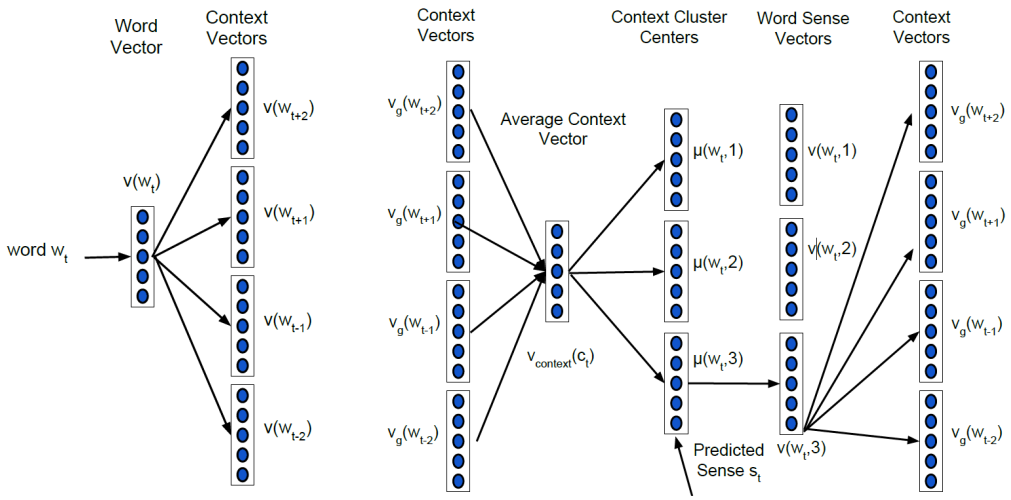

Figure 1: *(Figures 1-2, Neelakantan et al., 2015) Respective architecture of the skip-gram (left) and MSSG (right) models when the window size $R = 2$ is used and the context set C_t is $\{w_{t-1}, w_{t-2}, w_{t+1}, w_{t+2}\}$. In the MSSG model, the K-means clustering method with $K = 3$ is used to predict the senses of a multi-sense word.*

또한 w_t가 발생한 문맥에서 c_j를 발견하지 않을 확률은 $P(D = 0|v(w_t), v(c'_j)) = 1 - P(D = 1|v(w_t), v(c_j))$이 된다. noindent 따라서, 단어 열 $w_1, \ldots, w_T$을 포함하는 훈련 집합이 주어지면, 단어 임베딩은 다음의 목적함수

$$J = \sum_{t=1}^{T} \sum_{c_j \in C_t^+} \log P(D = 1|v(w_t), v(c_j)) + \sum_{t=1}^{T} \sum_{c'_j \in C_t^-} \log P\left(D = 0|v(w_t), v\left(c'_j\right)\right), \tag{2.2}$$

를 최대화함으로써 학습된다. 여기서 w_t는 훈련 집합에서 단어 w가 t번째로 발생한 문맥에서의 w를 의미하며, C_t^+는 w_t의 주변단어 집합, C_t^-는 w_t에 대한 주변단어를 아닌 단어를 무작위로 선택한 집합이다. 더 자세히 말하자면, 주어진 단어 w_t에 대한 주변단어 집합은 다음과 같다.

$$C_t^+ = \{w_{t-R}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+R}\}. \tag{2.3}$$

여기서 R은 skip-gram에서의 윈도우 크기인 튜닝 파라미터(tuning parameter)이며 단어의 문맥 범위를 지정하게 된다.

2.2. Multi-sense skip-gram (MSSG) 방법

Neelakantan 등 (2014)이 제안한 MSSG 방법은 단어 w에 대한 모든 문맥벡터를 K-means 방법으로 군집화하여 한 단어의 여러 의미들을 구분하는 것과 동시에 skip-gram처럼 딥러닝 모형을 이용하여 단어 임베딩을 함께 수행하는 다중 임베딩 방법이라 할 수 있다. MSSG 의 구조는 Figure 1의 오른쪽 패널과 같이 주어지며, 각 단어 w의 k ($k = 1, \ldots, K$)번째 의미에 대한 임베딩 벡터 $v_s(w, k)$와 해당 문맥 벡터들의 군집을 의미하는 문맥 군집의 중심 $\mu(w, k)$을 구하는 것을 목표로 한다.

먼저, 각 단어 w의 글로벌 벡터는 $v_g(w)$라고 표현하자. MSSG 방법에서 글로벌 벡터 $v_g(w)$는 주로 skip-gram 방법 결과 얻어진 단어 임베딩 벡터를 사용한다. 단어 w_t가 주어졌을 때, skip-gram 방법처럼 윈도우 크기

R을 설정하여 해당 단어의 주변단어 집합 $C_t = \{w_{t-R}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+R}\}$을 계산한다. 이 주변단어 집합 C_t을 이용하여 문맥을 표현하는 문맥 벡터 $v_{\text{context}}(C_t)$를 다음과 같이 정의한다.

$$v_{\text{context}}(C_t) = \frac{1}{2 * R} \sum_{c \in C_t} v_g(c). \tag{2.4}$$

여기에서 주변단어의 의미 벡터 대신 글로벌 벡터를 사용하여 문맥 벡터 $v_{\text{context}}(C_t)$를 정의하게 되는데, 이는 계산 부담을 크게 감소시키는 역할을 한다. 또한, R은 문맥벡터가 가지는 의미 정보의 범위를 조절하는 튜닝 파라미터가 된다. 작은 값의 R에서는 해당 단어의 적은 주변을 사용하여 국소적인 의미를 포함하는 문맥벡터가 발생될 것이고, 반대로 큰 값의 R에서는 넓은 주변을 사용하여 글로벌한 의미를 담는 문맥벡터가 생성될 것이다.

다음으로, 임의의 단어 w가 발생하는 위치들 t ($w_t = w$)에서 계산한 문맥벡터 $v_{\text{context}}(C_t)$들을 K-means 방법을 적용하여 군집화하며 문맥 군집들의 중심인 $\mu(w, k)$ ($k = 1, \ldots, K$)를 계산한다. 해당 벡터들은 고차원이기 때문에 좋은 성능을 준다고 알려져 있는 코사인 유사성을 이용하여 K-means 방법을 계산한다. 추가적으로, 군집화 결과로부터 해당 w_t 단어의 의미 s_t를 다음과 같이 예측하며,

$$s_t = \underset{k=1,\ldots,K}{\text{argmax}}\ \text{sim}\left(\mu(w, k), v_{\text{context}}(C_t)\right), \tag{2.5}$$

$\text{sim}(\mu(w, k), v_{\text{context}}(C_t))$은 두 벡터 $\mu(w, k)$와 $v_{\text{context}}(C_t)$ 사이에 코사인 값을 의미한다. 다시 말해, 해당 단어 w_t의 문맥벡터 $v_{\text{context}}(C_t))$와 군집중심들 $\mu(w, k)$, $k = 1, \ldots, K$ 사이의 코사인 값들을 계산하여 가장 코사인 유사성이 높은 군집으로 단어 w_t의 의미를 예측하게 된다. 이후, 해당 단어의 주변집합 C_t가 s_t 문맥 군집에 추가되므로 문맥 군집의 중심 $\mu(w_t; s_t)$는 새롭게 계산된다. 이때 skip-gram 모형과 유사하게 w_t의 문맥에서 주변단어 c_j를 관찰할 확률을 다음과 같이 모델링하면

$$P(D = 1 | v_s(w_t, s_t), v_g(c_j)) = \frac{e^{v_g(c_j)^\top v_s(w_t, s_t)}}{\sum_{j'=1}^{V} e^{v_g(c_{j'})^\top v_s(w_t, s_t)}}, \tag{2.6}$$

이에 해당하는 목적함수는

$$J = \sum_{t=1}^{T} \sum_{c_j \in C_t^+} \log P\left(D = 1 | v_s(w_t, s_t), v_g(c_j)\right) + \sum_{t=1}^{T} \sum_{c'_j \in C_t^-} \log P\left(D = 0 | v_s(w_t, s_t), v_g(c'_j)\right), \tag{2.7}$$

로 주어진다. 여기에서, w_t의 문맥에서 주변단어 주변단어 c_j를 관찰하지 않을 확률은

$$P\left(D = 0 | v_s(w_t, s_t), v_g(c'_j)\right) = 1 - \frac{e^{v_g(c_j)^\top v_s(w_t, s_t)}}{\sum_{j'=1}^{V} e^{v_g(c_{j'})^\top v_s(w_t, s_t)}},$$

로 계산된다. 식 (2.7)의 목적함수를 최대화함으로써 단어들에 대한 다중 임베딩 값들 $v(w_t, s_t)$, $t = 1, \ldots, T$을 학습한다.

기존의 MSSG 방법은 K-means 방법에서의 K 값을 사용자가 직접 지정해야 하며, 무엇보다, 모든 단어에서 군집의 수, 다시 말해, 각 단어에 대한 의미의 수가 K개로 모두 같다는 제약을 가지고 있다. 다시 말해, MSSG 방법은 모든 단어들을 K개의 의미에 해당하는 의미 임베딩 벡터 v_s를 생성하며, 이는 모든 단어들을 K개의 의미들을 지니는 다의어로 간주한다는 것이다.

3. 제안 방법 설명

이 절에서는 기존의 MSSG 기법의 성능을 개선하고 제약점을 완화시키는 방법들을 제안하고자 한다. 먼저, 일반적으로 텍스트에서 해당 단어에 가까울 수록 단어 의미와 연관이 높은 경향이 있다는 것을 반영하여, 주변단어들의 위치에 따라 다른 가중치(weights)를 두어 계산한 가중 문맥 벡터를 사용하는 것을 제안한다. 또한, MSSG에서 단어의 의미를 구별하기 위해 군집 분석을 진행할 때, 기존의 K-means 방법 대신 군집화를 수행하는 동시에 군집의 개수 또한 추정할 수 있는 X-means 방법을 적용하는 것을 제안한다. 제안된 방법은 군집의 개수 K를 사용자가 설정할 필요가 없으며, 단어에 따라 의미의 개수, 즉, 군집의 수를 달라지기 때문에 모든 단어의 의미의 개수가 일정하다는 기존 MSSG 가정을 완화시킨다. 나아가, 각 단어에 대하여 의미들을 구분지음과 동시에 의미 수가 되어지기 때문에 기존 MSSG 방법에서 K를 선택하는 것에 비해 계산 부담을 크게 감소시킨다.

3.1. 가중 문맥벡터

앞절 2.2에서 보듯이 기존 MSSG모델에서 주변단어들은 단어의 문맥 벡터를 구성할 때 모두 동일한 가중치를 사용한다. 하지만, 주변단어들의 위치에 따라 해당 단어의 연관성이 다를 수 있다, 예를 들어, “The power plant is working near the forest.”와 “The plant habitat near nuclear facility is big.” 두 문장들을 생각해보자. 관심 단어인 ‘plant’는 각각의 문장에서 ‘공장, 시설’ 과 ‘나무’를 두가지 뜻을 의미하는 다의어지만, 기존 MSSG 방법을 사용한다면 두 문장에서의 의미들을 구분하지 못할지도 모른다. 왜냐하면, 첫문장에서 ‘공장, 시설’와 연관된 주변 단어가 2개(power, work), ‘식물’와 관련된 주변단어가 1개(forest) 발생하며, 두 번째 문장에서 ‘공장, 시설’와 연관된 주변 단어가 2개(nuclear, facility), ‘식물’와 관련된 주변단어가 1개(habitat) 발생하여 단어 ‘plant’는 두 문장에서 모두 ‘공장, 시설’를 의미하게 될 것이다. 이런경우에, 주변단어들의 위치에 따라 가중치를 다루게 주어 문맥을 계산하는 것은 도움을 줄 수 있다. 특히, “power plant”나 “plant habitat”와 같은 구절에서 볼 수 있듯이, 목표 단어와 더 가까이 있는 주변단어들이 목표 단어의 의미에 대한 더 정확한 정보를 가질 수 있으므로 더 가까이 있는 주변단어들에 더 많은 가중치를 부여한 가중문맥 벡터를 사용하는 것을 제안한다. 구체적으로, 주변단어 집합 C_t의 윈도우 사이즈가 R일 때 가중 문맥 벡터는

$$v_{\mathrm{w,cont}}(C_t) = \sum_{|k|=1,2,\ldots,R} \mathrm{d}_k \times v_g(w_{t+k}), \tag{3.1}$$

로 주어진다. 여기에서, 가중치들 d_k, $k = \pm 1, 2, \ldots, R$은 음이 아닌 값을 가지고 $\sum_{|k|=1,2,\ldots,R} \mathrm{d}_k = 1$을 만족하며, $|k|$의 값에 따라 가중치 d_k가 단조감소(monotone decreasing)한다. 모든 가중치들을 똑같이 $d_k = 1/(2R)$로 설정하면 기존의 MSSG 방법에서 구하는 식 (2.4)의 문맥벡터 와 같아지며, 4절의 모의실험에서는 $|k|$ 값에 따라 선형적으로 감소하는 가중치 d_k를 사용했다. 즉, R 이 5일 때, 단어 w_t의 주변단어 집합

$$C_t = \{w_{t-5}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+5}\}, \tag{3.2}$$

에 다음과 같은 가중치 값들을 설정한다.

$$d_{-5} = \frac{1}{30},\ d_{-4} = \frac{2}{30}, \ldots, d_{-1} = \frac{5}{30},\ d_1 = \frac{5}{30}, \ldots, d_4 = \frac{2}{30},\ d_5 = \frac{1}{30}. \tag{3.3}$$

Figure 2는 식 (2.4)의 기존 문맥벡터와 비교하여 제안된 가중 문맥벡터에서 사용되는 기준치 체계(weight scheme)을 보여준다.

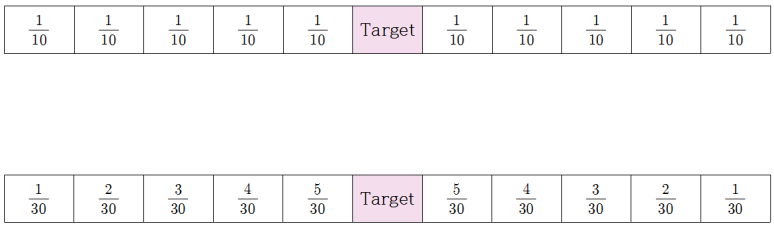

Figure 2: *Respective weight schemes of the ordinary (upper) and proposed weighted (lower) context vectors when the window size $R_m = 5$ is used.*

3.2. X-means 군집화

Dan과 Moore(2000)가 X-means 군집화 방법은 K-means 군집화 방법을 기초로 하여 군집화를 수행함과 동시에 Bayesian information criterion (BIC) 측도를 계산하여 군집의 개수 또한 최적화하는 군집 분석 방법이다. 구체적으로, 초기값으로 설정된 K 값(흔히 $K = 2$를 선택)에 대해 K-means 군집화하는 improve-parameters 단계와 BIC 기준을 이용하여 어떤 군집이 더 분할되어야 하는지 결정을 내리는 improve-structure라는 단계를 반복하여 수행한다. Improve-parameters 단계는 K-means 방법을 적용하여 K개의 군집과 그 중심을 얻으며, improve-structure 단계는 K-means 방법을 수행한 군집들에 대하여, 2-means 군집화를 반복적으로 수행하여 BIC를 기준으로 더 나은 군집을 찾는 과정이다. 이때 이미 실제 분포의 군집을 구성하고 있는 경우라면 이 과정에서 군집이 변화되지 않을 것이고, 반면에 실제 분포와 다른 군집을 구성하고 있다면 이 과정에서 군집 수를 증가시킴으로써 수정될 것이다. 이는 K-means 방법이 국소 최소값에 수렴하는 문제를 해결해 줄 뿐만 아니라 군집 수 K를 지정하지 않아도 되는 장점이 있다고 알려져 있다 (Tsunenori, 2005).

앞의 2.2절에서 설명한 기존 MSSG 모형 에서 문맥벡터들의 군집을 형성할 때 K-means 단계 대신 아래와 같은 X-means 알고리즘을 수행하여 학습시킨다.

(S1) 말뭉치에서 임의의 단어 w가 T번 사용되었을 때, T개의 식 (3.1)에서 제안한 가중문맥벡터들 $v^1_{\text{w,cont}}(C_1)$, $\ldots, v^T_{\text{w,cont}}(C_T)$ 에 대해 초기값 K에 대하여 K-means 군집화를 수행하고 해당 군집들을 $S_1, \ldots S_K$로 표현한다.

(S2) 각 $i = 1, \ldots, K$에 대하여, 아래의 과정들 1 ~ 3을 반복하여 최종 군집을 결정하여 최종 군집의 군집중심, 각 군집 내의 원소 결과를 저장한다.

1. 군집 S_i에 2-means 방법을 적용하여 군집 $S_i^{(1)}, S_i^{(2)}$를 얻는다.
2. 모든 군집들에 정규분포를 가정하고, 군집 $S_i \sim N(\mu_i, \sigma_i^2 I)$와 이를 나눈 군집 $S_i^{(1)} \sim N(\mu_i^{(1)}, (\sigma_i^{(1)})^2 \cdot I$, $S_i^{(2)} \sim N(\mu_i^{(2)}, (\sigma_i^{(2)})^2 \cdot I)$의 BIC 값들을 각각 계산하여 비교한다.
   a. $\text{BIC}(S_i) > \text{BIC}(S_i^{(1)}, S_i^{(2)})$인 경우에, 2-means를 적용하는 것을 선택하며 군집 S_i를 $(S_i^{(1)}, S_i^{(2)})$로 나누는 것을 택한다. $S_i^{(1)}, S_i^{(2)}$에서 군집을 더 나누는 것을 고려하기 위해 $S_i^{(1)}, S_i^{(2)}$ 차례대로 과정 1. 3. 를 반복한다. 만약 군집분석에서 영집합이 발생하면, 과정 3으로 이동한다.
   b. $\text{BIC}(S_i) \leq \text{BIC}(S_i^{(1)}, S_i^{(2)})$인 경우에, 더 이상 S_i를 나누지 않고 집합 S_i를 최종 군집으로 선정한다.
3. S_i에 대한 2-분할을 완료하고, S_i에 대한 최종 군집의 번호를 다시 매긴다.

Table 1: Respective meanings and titles of the abstracts when the ‘plant’ word is used (upper) and corpus vocabulary comparison with minword (bottom)

Meaning for ‘plant’	Titles of the abstract
tree	The ecodist package for dissimilarity-based analysis of ecological data
tree	Estimating and analyzing demographic models using the popbio package in R
power	Solar: solar radiation and photovoltaic systems with R
power	POPS: a software for prediction of population genetic structure using latent regression models
tree	Distance sampling in R
tree	Hyperspectral data analysis in R: The hsdar package
⋮	⋮

Minword	Vocabulary		
	Total	Retain	Ratio
1	5353	5353	100%
2	5353	3069	57%
3	5353	2226	42%
4	5353	1793	34%
5	5353	1494	28%

Minword means the total number of occurrences of a word in a corpus.

4. 실증예제에 기반한 모의실험

이 절에서는 제안된 방법들을 기존 방법과 비교하고 성능을 살펴보기 위한 실증예제에 기반한 모의실험을 실시한다. 모의실험에서, ‘plant’라는 단어를 분석의 목표(target)로 선택했고, 케임브리지 사전에 의하면, “땅 또는 물에서 자라며 줄기, 잎, 꽃을 가지고 씨를 생산하는 것”을 의미하는 식물과 “특정 제품이 만들어지거나 전력 등이 생산되는 공장”이라는 공장 두가지를 의미한다.

말뭉치는 “Journal of Statistical Software (JSS)”에 기재된 논문들의 초록들로 구성되었고, 이것을 R package ‘topicmodels’에서 사용 가능하며, 이에 대한 설명은 Grun과 Hornik (2011)에서 확인할 수 있다. 저널 JSS에 2020/06/30까지 게재된 논문들의 초록들로 중복된 자료들을 제외하여 총 568개의 문서를 얻었다. 해당 말뭉치에서 ‘plant’ 단어가 발생하는 6개의 문서에서 ‘식물’와 ‘공장’의 의미를 갖는 문장을 각각 5개와 1개를 추출하였다. 또한, “scientific research publishing (SCIRP: https://www.scirp.org)”라는 과학 저널에 기재된 논문들 중 초록에 ‘plant’ 단어를 포함하는 논문들을 검색하여 ‘plant’가 ‘식물’ 과 ‘공장’이라는 의미로 사용된 각 10개의 초록들을 추가하여 분석하려는 말뭉치에 포함했다. 다시 말해, 말뭉치를 구성하는 총 588개의 문서들, 즉, 초록들, 중 ‘plant’라는 단어는 26개의 문서 내 72개의 문장에서 발생했고, 총 72번 중에서 32번은 ‘식물’ 을, 나머지 40번은 ‘공장’을 의미하고 있다. Table 1의 위쪽 표는 ‘plant’가 등장하는 초록 제목의 예제를 알려주며, 단어 ‘plant’가 식물을 뜻하는 경우는 ‘tree’, 공장을 뜻하는 경우는 ‘power’로 표시했다.

단어 임베딩 기법을 적용시키기 전에, 프로그램 python의 spacy 패키지에서 pipe 함수를 통해 말뭉치에 대해 단어 토큰화(token)를 수행했고, token 함수를 사용하여 표제어 추출(lemmatize) 단계와 불용어 처리 등 전처리 과정을 완료하였다. 이때, 말뭉치에서 ‘a’와 ‘R’ 처럼 의미가 없는 한 글자의 단어들을 제외했다. 이후, 프로그램 python의 collections 패키지의 defalutdict, counter 함수를 이용하여 단어가 말뭉치에서 발생 하는 빈도의 최소값, 즉, minword과 해당 minword 값 이상 발생한 단어들의 수를 계산하고, 그 결과는 Table 1의 아래쪽 표에서 확인할 수 있다. 일반적으로, 아주 큰 용향의 말뭉치에서는 높은 minword 값을 사용하지만, 본 논문에서 분석하는 말뭉치는 그 양이 상대적으로 적어 데이터 손실이 크며 초록이라는 특성상 간결하게

Table 2: 10 neighbors closest to the word 'plant' labeled by meaning in the corpus

Method	Sense	Neighbors
Skip-gram	-	gas, nuclear, thermodynamic, operational, energy, grf, regulation, wild, turbine, trigger
Weighted	tree	seed, trigger, growth, fertilization, regulation, pgpr, grf, acid, gnetophyte, stomatal
	power	power, nuclear, gas, exergy, turbine, plant, reasonable, operational, thermal, energy
Non-weighted	tree	seed, trigger, fertilization, growth, regulation, acid, grf, gnetophyte, stomatal, pgpr
	power	power, nuclear, gas, exergy, plant, energy, turbine, reasonable, operational, thermal

In weighted and non-weighted method, the embedding vector of each meaning is the average of the context vectors of the corresponding meaning generated by each method. In Skip-gram method, most of words that are close to the meaning of 'power' occurred. This indicates that it does not contain two meanings: 'plant'. On the other hand, given the close words in the weighted or non-weighed method, it seems to represent the meaning well.

Table 3: The left table shows the estimation results of the proposed method for the number of meanings for the word 'plant', i.e., the results of the estimated number of clusters after performing the X-means method on weighted context vectors over 1,000 replications when $R_m = 2, 3, 4, 5, 6, 7$, $d = 400$ and minword = 3. The right table presents the averaged and median values of 1,000 simulated clustering accuracy results when 4 MSSG methods using X-means and K-means methods combined with weighted and non-weighted context vectors, are conducted and $R_m = 7$, $d = 400$, and minword = 3

R_m	Cluster number					
	2	3	4	5	6	7
1	665	249	62	19	4	1
3	645	260	71	20	3	1
5	999	1	0	0	0	0
7	1000	0	0	0	0	0
9	1000	0	0	0	0	0

Clustering	Weighted		Non-weighted	
	Mean	Median	Mean	Median
X-means	89.2	**90.3**	86.8	**87.5**
2-means	**89.7**	**90.3**	**87.3**	**87.5**
3-means	75.9	73.6	72.8	73.6
4-means	64.6	66.7	62.3	62.5
5-means	56.5	55.6	54.4	54.2

표현되므로 비교적 작은 minword 값들을 고려했다. 예를 들어, minword = 1 기준으로 해당 말뭉치에서 총 5,353개의 단어가 생성된다. 프로그램 python의 gensim패키지의 word2Vec 함수로 skip-gram 방법을 적용하여 글로벌 단어 임베딩 벡터를 얻었고, 이때 Neelakantan 등 (2014)와 같은 윈도우 크기값 $R = 5$를 정했고, 임베딩 차원 d를 여러개의 값들 300, 400, 500을 고려했다. 이후, 얻어진 글로벌 단어 임베딩 벡터들을 기반으로 기존 MSSG 방법과 3절에서 제안한 방법들을 적용한 변형 MSSG 방법을 각각 적용하여 다중 임베딩, 즉, 목표 단어의 의미 임베딩 벡터를 얻었다. 두 방법에서 모두 의미 임베딩 벡터의 차원은 글로벌 단어 임베딩 벡터의 차원과 같고 다음의 윈도우 크기 값들 $R_m = 1, 3, 5, 7, 9$을 고려했다.

먼저, 제안한 변형 MSSG 방법, 즉, 가중 문맥벡터와 X-means를 활용한 방법이 단어 'plant'의 의미를 잘 구분했는지 살펴보았고, 1,000번 X-means 군집방법을 적용했을 때 군집수, 즉, 단어 'plant'의 의미 갯수 선택 결과는 Table 3의 왼쪽 표에서 확인 할 수 있다. 이 때, $d = 400$, minword = 3, $R_m = 1, 3, 5, 7, 9$로 설정했다. 이 표로부터 R_m이 5 이상일 때 가중 문맥벡터들에 대한 X-means 방법의 의미 수 추정 측면에서 상당히 좋은 성능을 보였음을 알 수 있다. 또한, 제안된 가중 문맥벡터와 X-means 방법 각각에서 성능의 개선된 양을 살펴보기 위해, X-means 방법과 군집 수 K를 $2, 3, 4, 5$로 지정한 K-means 방법을 가중 문맥벡터와 비가중 문맥벡터에 대하여 1,000번 수행한 후 단어 'plant'의 의미 분류 정확도의 평균과 중앙값으로 비교를 수행했고, 그 결과는 Table 3의 오른쪽 표에서 확인할 수 있다. 제안한 가중 문맥벡터 방법은 군집 방법에 상관없이 더 정확한 의미 분류 결과를 주었고, 참인 군집 수 $K = 2$를 이용한 2-means 방법과 비슷하거나 더 높은 성능을 보인 것을 알 수 있다. 추가적으로, Table 4는 다른 튜닝 파라미터들-윈도우 사이즈 R_m, 임베딩벡터의 차원 d, minword의 여러

Table 4: averaged classification accuracy results for the word 'plant' over 100 replications using the MSSG and proposed modified MSSG, respectively

Method	d	Minword = 1					Minword = 2				
		$R_m = 1$	$R_m = 3$	$R_m = 5$	$R_m = 7$	$R_m = 9$	$R_m = 1$	$R_m = 3$	$R_m = 5$	$R_m = 7$	$R_m = 9$
MSSG	300	80.0	82.6	86.8	**87.6**	85.3	80.5	84.0	**87.8**	85.3	85.9
Modified MSSG	300	77.0	**78.7**	78.0	72.2	66.5	77.0	**78.7**	78.0	72.2	66.5
MSSG	400	80.1	82.3	**86.8**	85.6	85.7	79.8	83.9	**88.4**	85.5	85.8
Modified MSSG	400	77.9	82.1	87.1	88.6	**89.1**	77.9	82.1	87.1	88.6	**89.1**
MSSG	500	79.8	82.3	**86.4**	85.9	84.8	80.5	86.9	**88.9**	85.4	87.4
Modified MSSG	500	75.0	83.9	86.2	87.7	**88.3**	75.0	83.9	86.2	87.7	**88.3**
Method	d	Minword = 3					Minword = 5				
		$R_m = 1$	$R_m = 3$	$R_m = 5$	$R_m = 7$	$R_m = 9$	$R_m = 1$	$R_m = 3$	$R_m = 5$	$R_m = 7$	$R_m = 9$
MSSG	300	79.8	87.8	**90.1**	88.1	89.5	82.2	88.7	**89.9**	87.4	81.3
Modified MSSG	300	78.3	79.3	**80.8**	74.3	59.8	79.4	80.1	**85.0**	79.9	67.3
MSSG	400	81.3	85.8	**88.2**	87.1	88.0	81.6	86.7	**89.0**	87.1	80.6
Modified MSSG	400	78.0	75.5	**89.1**	89.1	89.0	77.4	72.8	87.7	**89.4**	88.0
MSSG	500	79.8	88.6	87.1	87.5	**89.4**	82.1	89.4	**90.4**	82.3	77.7
Modified MSSG	500	74.0	84.7	88.4	**89.2**	88.7	74.2	85.5	**89.1**	89.0	88.9

선택에 대해 기존 MSSG 방법($K = 2$)과 본 논문에서 제안한 변형 MSSG 방법의 정확성 결과를 보여준다. 특히, 윈도우 크기가 크고 minword을 3 이상으로 설정했을 때, 제안한 modified MSSG 방법이 단어 의미의 수를 참값 $K = 2$으로 설정한 기존 방법과 비슷한 결과를 준다는 것을 관찰할 수 있다.

5. 결론

본 논문에서는 기존 MSSG 방법의 우수한 성능에도 불구하고 다의어의 의미 임베딩의 정확성을 더 높이는 문제와 문맥벡터를 군집화하는 과정에서 최적 군집 수 K를 추정하는 문제 즉, 단어들의 의미 수를 정확하게 추정하는 문제가 여전히 있다는 점에 착안하여 위치에 따른 가중치를 다르게 준 가중 문맥벡터와 군집 분석 과정에서 최적 군집 수 K를 함께 추정해주는 X-means를 활용한 변형 MSSG 방법을 제안했다. 나아가, 제안한 변형 MSSG 방법은 모든 단어의 의미 수를 K개로 지정하게 되는 가정을 완화시키고 단어에 따라 의미 갯수를 다르게 하는 것이 가능한 유연성을 가지고 있다. 실증예제에 기반한 모의실험을 수행하여 제안한 방법이 자료로부터 단어의 의미수, 즉, 군집의 수 K 값을 정확하게 추정하며 참값인 단어 의미의 수 K를 이용한 기존 방법에 필적할만한 성능을 보일 수 있는 잠재력을 확인했다. 일반적으로, 방법의 성능은 다른 튜닝파라미터-텍스트 전처리 과정에서의 minword, 임베딩벡터의 차원, 윈도우 사이즈-의 값들에 의존하며, 이것들을 경험적으로라도 적절하게 선택하는 것은 텍스트마이닝에서 매우 중요한 문제로 더 연구할 만한 가치가 있는 주제이다. 추후, 아주 큰 용량의 텍스트 말뭉치에 제안한 변형 MSSG 방법을 적용하여 학습하고 전체 자료의 군집 분석 결과를 유사성 측도로 비교하는 후속 분석을 고려하고 있다.

References

Dan P, Moore AW (2000). X-means: extending k-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference on Machine Learning*, **727**.

Grün B and Hornik H (2011). Topicmodels: an r package for fitting topic models, *Journal of Statistical Software,*

40, 1–30.

Huang E, Socher R, Manning C, and Ng A (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, **1**, 873–882.

Huang E, Socher R, Manning C, and Ng A (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, **1**, 873–882.

Neelakantan A, Shankar J, Passos A, and McCallum A (2014). Efficient non-parametric estimation of multiple embeddings per word in vector space. *Conference on Empirical Methods in Natural Language Processing*, 1059–1069.

Rong X (2014). *Word2vec Parameter Learning Explained*, arXiv.

Mikolov T, Sutskever I, Chen K, Corrado G, and Dean J (2013a). Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems*, **26**, 3111–3119.

Mikolov T, Chen K, Corrado G, and Dean J (2013b). Efficient estimation of word representations in vector space. *International Conference on Learning Representations*.

Tsunenori I (2005). An expansion of X-means for automatically determining the optimal number of clusters. In *Proceedings of International Conference on Computational Intelligence*, **2**, 91–95.

Zheng Y, Shi Y, Guo K, Li WL, and Zhu L (2017). Enhanced word embedding with multiple prototypes. *4th International Conference on Industrial Economics System and Industrial Security Engineering*, 1–5.

Received March 13, 2021; Revised May 2, 2021; Accepted May 4, 2021

가중 문맥벡터와 X-means 방법을 이용한 변형 다의어스킵그램

정현우[a], 이은령[1,a]

[a]성균관대학교 통계학과

요 약

최근 자연어 처리 문제에서의 단어 임베딩은 아주 큰 주목을 받고 있는 연구 주제이며 스킵그램은 성공적인 단어 임베딩 기법 중 하나이다. 주변단어들 정보를 이용해서 단어들의 의미를 학습하여 단어 임베딩 벡터를 할당하며 텍스트 자료를 효과적으로 분석할 수 있게 한다. 그러나 벡터 공간 모델의 한계로 인해 기본적인 단어 임베딩 방법들은 모든 단어가 하나의 의미를 가지고 있다는 것을 가정한다. 다의어, 즉 하나 이상의 의미를 가진 단어가 실생활에서 존재 하기 때문에 Neelakantan 등 (2014)은 군집분석 기법을 이용하여 다의어의 여러 의미들에 해당하는 의미 임베딩 벡터를 찾기 위해 MSSG (multi-sense skip-gram)를 제안했다. 본 논문에서는 MSSG의 통계적 성능을 개선시킬 수 있는 변형된 MSSG 방법을 제안한다. 먼저, 가중치를 활용한 가중문맥벡터를 제안한다. 나아가, 군집의 수, 즉 다의어의 의미 수를 자료에서 자동적으로 추정해주는 x-means 방법을 활용한 알고리즘을 제안한다. 본 논문에서 수행한 실증자료를 기반한 모의실험에서 제안한 방법은 기존 방법에 비해 우수한 성능을 보여주었다.

주요용어: 단어 임베딩, 스킵그램, 다의어, 다중의미 스킵그램, X-평균 군집화, 가중문백벡터

[1]교신저자: (03063) 서울시 종로구 성균관로 25-2, 성균관대학교 통계학과. E-mail: erlee@skku.edu