# Visualizing Brands and Extracting Keywords from User-Generated Content using Distributed Representation

양호성 조성준
hoseong@dm.snu.ac.kr zoon@snu.ac.kr

December 27, 2015

**Abstract**

소비자가 가진 브랜드에 대한 인식은 구매에 직접적 영향을 주기 때문에 매우 중요하다. 기업에서는 경쟁사 대비 소비자가 인식하는 자사 브랜드의 포지셔닝을 확인하기 위해 지각도(Perceptual Map)를 사용하여 경쟁사 분석 및 서비스 개선 전략 수립 등 의사결정에 활용하고 있다. 이를 위해 전통적으로 기업들은 설문조사를 통해 데이터를 수집했으나 설문조사 방법은 주관적이고, 설문 인원 및 주제가 제한되는 등 많은 문제점이 존재한다. 이러한 단점을 보완하기 위해 기업들은 UGC(User generated content)를 활용하기 시작하였다. UGC는 적은 비용으로 소비자들의 살아있는 의견을 들을 수 있지만, 대부분 텍스트 데이터이고 양이 많아 분석이 어렵다. 따라서 특정 단어의 빈도만 살펴보거나 빈도의 추이를 살펴보는 방법을 사용한다. 그러나 빈도 기반 방법은 브랜드 간의 유사도 계산이 어려워 포지셔닝을 확인하기 어렵다. 또한, 전처리가 많아 분석 과정에서 주관이 많이 개입되므로 재현성과 객관성이 떨어진다. 본 연구에서는 이러한 단점을 보완하기 위해 브랜드를 하나의 분산 표상(Distributed representation)으로 표현하는 Brand2Vec 방법을 제안한다. 브랜드 벡터를 활용해 소비자가 인식하는 브랜드의 계층적 관계를 dendrogram을 시각화하고, t-SNE를 이용하여 여러 브랜드의 포지셔닝을 시각화한다. 시각화뿐만 아니라 다른 브랜드에 비해 상대적으로 자사 브랜드의 특징을 나타내는 키워드를 추출하는 방법을 제안한다. 이러한 과정이 parameter에 강건하고 사람의 개입을 최소화하였기 때문에 객관성과 재현성을 확보하였다. 추후 리뷰 데이터뿐만 아니라, 각종 UGC를 활용하여 연예인, 정치인 등 다양한 곳에 확장할 수 있을 것이다.

## 1 서론

### 1.1 연구 배경

마케팅에서 '브랜드'에 대한 소비자들의 인식을 파악하는 것은 매우 중요하다. 특히 비슷한 성능의 제품이 다양하게 있을 때, 브랜드 이미지는 구매 의사에 중요한 영향을 끼친다. 따라서 자사 브랜드에 대한 소비자들의 인식은 물론이고 타사 브랜드와 자사 브랜드에 대한 상대적 인식의 차이를 파악하는 것도 중요하다. 이런 인식의 차이를 시각화한 도구가 지각도(Perceptual Map)이다. 지각도는

소비자의 인식 속에 존재하는 제품이나 브랜드의 상대적 위치를 나타낸 것을 말하며 포지셔닝맵(Positioning Map)이라고도 불린다.

브랜드 지각도를 그리기 위해 많이 사용되는 방법은 분석자가 파악하고 싶은 속성에 대해 설문조사를 실시하여 데이터를 수집한 후, MDS(Multidimensional scaling)나 대응분석(correspondence analysis)를 사용하여 2차원으로 시각화하는 것이다 [1, 2]. 이러한 설문 조사 방식은 분석이 용이하고, 의사결정자가 원하는 속성을 데이터에 반영할 수 있다는 장점이 있다. 그러나 시간적, 공간적, 금전적 한계로 인하여 제한된 표본과 주제를 대상으로 조사할 수밖에 없고, 설문자의 의도가 개입되기 때문에 객관적으로 사용자의 브랜드에 대한 인식을 파악할 수 없다.

이러한 단점을 보완하기 위해 UGC(User-generated contents)를 활용하여 소비자의 상대적인 브랜드 인식을 시각화하려는 시도가 있었다 [3, 4]. UGC는 사용자가 자발적으로 만든 블로그, 게시판, 채팅, 트위터, 이미지, 비디오, 음성, 광고 등 다양한 형태의 콘텐츠를 말한다 [5]. UGC는 적은 비용으로 다양한 공간에서 쉽게 데이터를 수집할 수 있고, 새로운 이슈에 대해서 지속적이며 자발적으로 업데이트 되기 때문에 소비자들의 살아있는 의견을 들을 수 있다. 또한, 사용자가 작성한 UGC의 특성에 따라 시간, 평점, 작성한 위치, 공유된 횟수 등 다양한 정보를 함께 수집할 수 있다. 무엇보다도 수많은 사람들이 자신의 의견을 표현 및 공유하고, 공유한 콘텐츠를 많은 사람들이 소비하기 때문에 브랜드에 미치는 영향력이 크다.

기존에 지각도를 그리기 위해 주로 사용한 방법은 UGC의 텍스트 데이터에서 브랜드가 등장한 횟수 정보를 활용하는 것이다. 그러나 브랜드가 언급된 횟수 데이터를 통해 브랜드 간의 유사성을 파악하기 어려울 뿐만 아니라 UGC의 장점인 브랜드에 관해 언급된 소비자들의 다양한 텍스트 정보를 활용하지 못하게 된다. 소비자들이 브랜드에 대해 언급한 다양한 단어 정보를 통해 자사 브랜드의 키워드를 추출할 수 있으면 보다 소비자 인식에 대해 풍부한 해석이 가능하여 의사결정에 도움을 줄 수 있을 것이다.

## 1.2 연구 내용

본 연구에서는 단어를 벡터로 표현하는 Word2Vec [6] 방법을 발전시켜 브랜드를 벡터로 표현할 수 있는 Brand2Vec 방법을 제안한다. 기존의 방법론과 달리 리뷰 데이터에 등장하는 모든 단어를 활용하였으며, 브랜드와 단어가 동시에 분산 표상(Distributed representation) 벡터로 표현되기 때문에 유사도 계산이 가능하다. 본 연구에서는 이러한 장점을 활용하여 브랜드 간의 유사도 계산을 통해 브랜드의 계층적 관계와 브랜드 간의 포지셔닝을 시각화하고, 브랜드와 단어 간의 유사도 계산을 통해 키워드를 추출하는 방법에 대해서 살펴본다.

본 논문에서는 제안하는 Brand2Vec 방법은 Word2Vec과 Doc2Vec [7] 방법을 확장한 것으로, 인공신경망을 사용해 제품 리뷰 문서에 나와 있는 단어의 language model을 학습시킴과 동시에 브랜드를 분산 표상 벡터로 표현하는 방법이다. 브랜드가 분산 표상 벡터로 표현되기 때문에 기존의 설문조사나 빈도기반 방법과 다르게 객관적으로 브랜드 간의 유사도를 계산할 수 있다. 이러한 성질을 활용하면 Dendrogram, t-SNE를 이용하여 브랜드들 간의 계층적 관계 및 포지셔닝을 시각화할

수 있다.

또한, 동일 제품군을 판매하는 브랜드의 경우 경쟁사와 두드러지는 점을 찾아내는 것이 중요하다. Brand2Vec은 단어와 벡터 간의 유사도 계산이 가능하기 때문에 브랜드와 유사한 단어를 추출할 수 있다. 이러한 성질을 활용하여 경쟁 브랜드와 비교해 상대적인 특징을 나타내는 키워드를 추출할 수 있다. 제안한 방법을 Apple, Microsoft 두 브랜드에 적용해서 타 브랜드에 비해 상대적으로 특징적인 단어를 추출하고 정성적으로 검증하였다.

본 연구의 contribution은 다음과 같다. 첫째, 브랜드를 벡터로 표현하여 브랜드 간 유사도 측정 및 계층적 관계와 포지셔닝을 시각화하여 비즈니스 의사결정에 도움을 주었다. 둘째, 브랜드 리뷰에서 브랜드 간의 차이를 나타내는 키워드를 자동화된 방법으로 추출하여 브랜드 간의 차이를 파악하는 데 추가적인 해석력을 제공하였다. 셋째, Parameter 탐색을 통해 parameter 민감도가 낮음을 확인하였다. 게다가, 전처리도 적기 때문에 객관성과 재현성을 확보하였다. 넷째, 자연어 처리 분야에서만 주로 사용되던 분산 표상 방법을 비즈니스 분야로 확장하였다. 마지막으로, 동일한 방법론으로 브랜드뿐만 아니라 영화, 연예인, 정치인 등 다양한 대상을 표현하는 데 확장할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 UGC를 활용한 사례와 Word2Vec에 대한 설명을 하고, 3장에서 제안한 Brand2Vec 방법에 대한 설명과 함께 parameter 탐색 과정에 대해 설명한다. 4장에서는 parameter 탐색 결과와 함께 Brand2Vec 방법을 활용한 브랜드 시각화와 키워드 추출결과를 확인한다. 마지막 5장은 본 연구의 의의와 추가 발전 방향에 대해서 서술하는 것으로 구성하였다.

# 2 관련 연구

## 2.1 UGC를 활용한 브랜드 지각도

UGC는 설문조사 방법에 비해 자사 브랜드 뿐만 아니라 경쟁사 브랜드에 대한 데이터도 쉽게 수집할 수 있고, 소비자들의 살아있는 의견이 반영되어 있다는 장점 때문에 브랜드 지각도로 시각화하려는 다양한 시도가 있었다. [4]에서는 소셜 미디어에서 고급 와인 브랜드가 등장한 빈도를 가지고 대응분석을 하여 와인 브랜드들을 시각화하였으며 [3]에서는 뉴스 사이트에서 해당 브랜드가 언급된 횟수를 수집하고, MDS(Multidimensional scaling)와 Minimum spanning tree를 활용하여 브랜드들을 2차원으로 시각화하였다. 두 방법의 공통점은 온라인상에서 해당 브랜드가 언급된 횟수만을 고려하였다는 것이다. 즉, 사용자들이 브랜드의 제품이나 서비스에 대해 언급한 단어들을 고려하지 않았다. 게다가 UGC 상에서 브랜드 언급 횟수가 비슷하다고 해서 비슷한 성격의 브랜드라고 볼 수 없으므로, 정보 손실이 크고 객관적인 방법이라 할 수 없다.

[8]에서는 LDA(Latent Dirichlet Allocation)를 사용하여 브랜드별로 각 토픽에 해당하는 단어 분포 정보를 비교하여 MDS로 시각화하였다. 이 방법은 전체 텍스트 데이터는 활용하였으나, LDA 모델을 사용하였기 때문에 parameter에 매우 민감하고 계산 복잡도가 높아 많은 양의 데이터를 학습하는데 부적절하다. 또한, 토픽을 추출하는 데 있어서 주관이 많이 개입되기 때문에 재현이 어렵다. 게다가 LDA 방법은 자사 브랜드의 특징을 나타내는 키워드를 추출할 수 없으므로 해석이

제한적이다.

하지만 본 연구에서 제안하는 Brand2Vec 방법은 많은 양의 데이터도 빠르게 처리할 수 있으며, 소비자들이 브랜드에 대해 언급하는 모든 텍스트 정보를 반영하여 각 브랜드 간의 유사도 계산은 물론이고 브랜드와 단어 간 유사도 계산이 가능하여 각 브랜드의 특징을 나타내는 키워드 추출이 가능하다. 이러한 과정에 사람의 개입을 최소화하였기 때문에 재현성과 객관성을 확보하였다.

## 2.2 분상 표상 방법

**단어의 분산 표상 방법** 텍스트 데이터를 기계학습 알고리즘에 적용하기 위해서는 숫자로 변환하는 과정이 필요하다. 이 과정은 크게 이산(Discrete) 방법과 분산(Distributed) 방법이 있다. 이산 방법에서 대표적인 것은 One-hot encoding으로, 총 $V$개의 단어가 있다면 $V$차원의 벡터에서 한 원소만 1이고 나머지는 0으로 표현하는 방법이다. 그러나 이러한 방법은 단어 벡터가 단어의 의미(semantic) 정보나 구문(syntactic) 정보를 반영하지 못할 뿐만 아니라, 단어와 단어 사이의 유사도를 계산할 수 없다.

이러한 단점들을 보완하고자 단어를 분산 표상으로 표현하는 방법이 제안되었다 [9]. [10]에서는 인공신경망을 활용하여 language modeling을 학습하는 동시에 단어를 분산 표상으로 표현하는 방법을 제안하였다. 이와 같은 분산 표상 방법은 이산 방법과 다르게 단어를 의미와 구문 정보를 반영한 벡터로 표현할 수 있다. 예를 들어, "big"이란 단어와 "biggest"라는 단어의 관계는 "small", "smallest"의 관계와 같다고 할 수 있다. 이러한 관계가 분산 표상 방법으로 표현한 단어 벡터 사이에 유지되므로 아래와 같은 식이 성립된다.

$$vector(\text{“}biggest\text{”}) - vector(\text{“}big\text{”}) \approx vector(\text{“}smallest\text{”}) - vector(\text{“}small\text{”})$$

이러한 장점 때문에 [10] 이후 다양한 분산 표상 방법이 등장하였다 [11, 12]. 그러나 이러한 방법들은 계산 복잡도가 매우 높아서 많은 양의 텍스트를 표현하기에 부적절했다. 2013년에 이러한 단점을 보완해 줄 수 있는 Word2Vec 모델이 제안되었다 [6]. Word2Vec은 간단한 인공신경망 모델을 활용하여 수억 개의 텍스트 데이터도 효율적으로 벡터로 표현할 수 있다. 또한, 전처리 과정이 거의 필요 없고, 표현된 단어 벡터가 문맥 정보를 포함하고 있다는 장점 때문에 감정분석(Sentiment analysis), 기계번역(Machine translation) 등 다양한 자연어 처리 분야 연구에서 활용되고 있다 [13, 14].

Word2Vec 모델은 주변 몇 개의 문맥 단어들로부터 다음 단어를 예측하는 모델인 CBOW(Continuous Bag-of-Word) 모델과 한 단어들로부터 주변 문맥 단어들을 예측하는 Skip-gram 모델이 있다. 먼저, 그림 1 은 CBOW 모델을 시각화한 것이다.

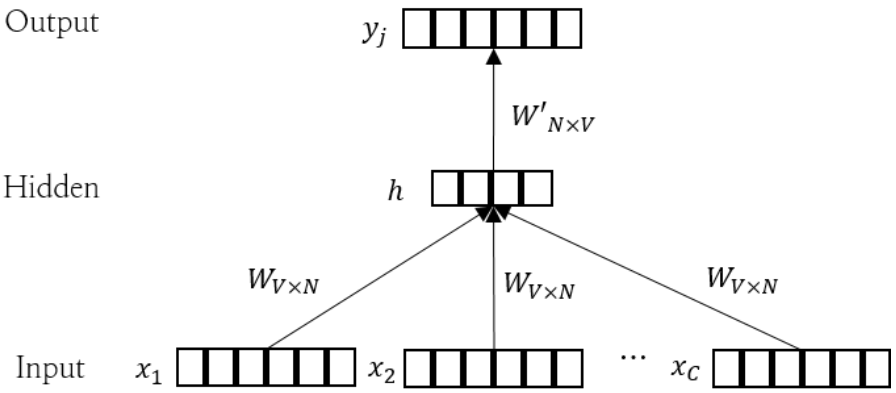


그림 1: CBOW 모델의 구조

CBOW 모델은 Window size(혹은 문맥 단어) $C$개 만큼 입력 노드가 있고, 은닉층(hidden layer)이 하나이면서 활성함수(Activation function)가 없는 인공신경망 모델이다. 입력 노드 $x_i$는 해당 단어에 대해 One-hot encoding으로 표현된 값으로 각각 $V$차원 벡터이다. 은닉층 $h$는 $N$차원으로 $N$은 표현하고자 하는 단어 벡터의 차원 수와 같다.

입력층과 은닉층 사이의 가중치 메트릭스를 $W_{V\times N}$, 은닉 노드와 출력 노드 사이의 가중치 메트릭스를 $W'_{N\times V}$ 이라고 한다. 목적함수는 (1)과 같이 주어진 $C$개의 단어가 있을 때 다음 단어가 나타날 확률을 최대화하는 것이다.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_t|w_{t+j}) \tag{1}$$

이 때 $T$는 전체 단어 개수이며, $2c$는 window size를 의미한다. $p(w_t|w_{t+j})$는 식 (2)와 같이 Softmax 함수를 이용하여 계산한다.

$$\log p(w_O|w_I) = \frac{\exp({v'_{w_O}}^T)\cdot h}{\sum_{j=1}^{V}\exp({v'_{w_j}}^T\cdot h)} \tag{2}$$

$w_I$와 $w_O$는 각각 입력, 출력 단어를 의미하며, $v'_{w_O}$ 은 출력 단어 벡터 값으로 $W'_{N\times V}$ 메트릭스의 $o$번째 행을 의미한다. 은닉층 $h$는 입력 단어 벡터들의 평균 값이다.

$$h = \frac{1}{C}W\cdot(x_1+x_2+\cdots+x_C) = \frac{1}{C}(v_{w_1}+v_{w_2}+\cdots+v_{w_C}) \tag{3}$$

$v_{w_i}$는 입력 단어 벡터 값으로 $W_{V\times N}$ 메트릭스의 $i$ 번째 열을 의미한다. Stochastic Gradient Descent와 Backpropagation [9]을 이용하여 $W_{V\times N}$ 과 $W'_{N\times V}$을 학습하며, 최종적으로 단어 $w_k$의 벡터는 $W_{V\times N}$의 $k$번째 열 벡터 혹은 $W'_{N\times V}$ 의 $k$번째 행 벡터가 된다.

Skip-gram 모델은 그림 2와 같은 구조이며, CBOW 모델에서 입력층과 출력층이 바뀐 것을 제외하고 흡사하다.

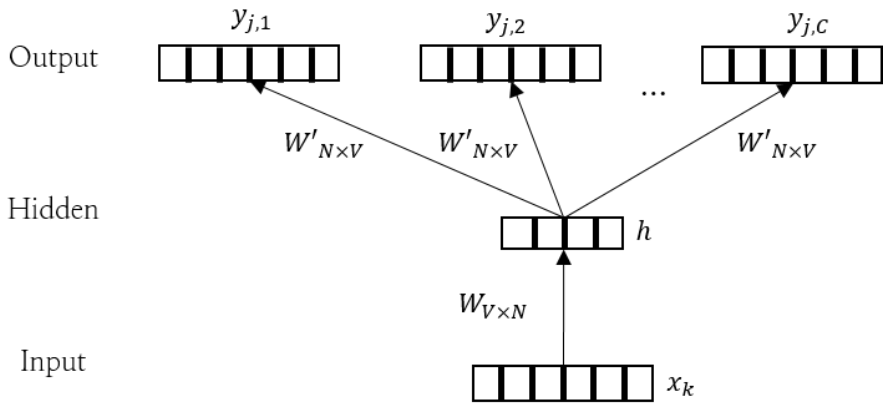


그림 2: Skip-gram 모델의 구조

목적함수는 식 (4)와 같이 주어진 한 단어로부터 문맥 단어들을 예측하는 확률을 최대화 하는 것이다.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{4}$$

입력노드 개수가 바뀌었기 때문에 $p(w_{t+j}|w_t)$은 아래와 같이 계산한다.

$$\log p(w_O|w_I) = \prod_{c=1}^{c}\frac{\exp({v'_{w_O}}^T)\cdot h}{\sum_{j=1}^{V}\exp({v'_{w_j}}^T\cdot h)} \tag{5}$$

그리고 Skip-gram과 달리, CBOW는 입력층 단어가 하나이므로 CBOW와 달리 은닉노드의 값은 식 6과 같다.

$$h = W\cdot x_k = W_k \tag{6}$$

$W_{V\times N}$과 $W'_{N\times V}$에 대한 학습은 CBOW 모델과 같이 Backpropagation을 통해 이뤄진다. 그 밖에 두 모델의 학습 속도를 빠르게 하기 위해 Negative Sampling이나 Hierarchical softmax등의 방법도 사용되고 있다 [15].

**문서의 분산 표상 방법: Doc2Vec** Word2vec은 단어를 벡터로 표현하는 비교사 방법이다. 단어뿐만 아니라, 문장 혹은 문서를 분산 표상 방법으로 표현하는 Doc2Vec [7] 방법도 제안되었다. 기존에 문서를 표현하는 데 있어서 TF-IDF [16] 방법이 많이 활용되었지만, 문서와 단어 수가 늘어남에 따라 차원이 매우 커지고, 희소행렬(Spare matrix)로 문서를 표현하기 때문에 차원의 저주 문제가 발생하는 단점이 있다.

Doc2Vec은 TF-IDF와 같은 Bag-of-words [17] 방법의 단점을 보완하여 감성(Sentiment) 분류 문제에서 매우 높은 성능을 보여주었다. Doc2Vec 방법은 그림 3과 같이 단어를 학습함과 동시에 문장 혹은 문서의 고유한 벡터를 같이 학습하게 된다. 그림 3의 좌측은 Word2vec의 CBOW와 같이 주변 문맥 단어와 문서 벡터를 통해 다음 단어를 예측하는 모델인 PV-DM(Paragraph Vector-Distributed Memory)이고, 우측은 Skip-gram 모델과 비슷하게 문서 벡터로부터 주변 단어를 예측하는 모델인 PV-DBOW(Paragraph Vector-Distributed Bag-of-words)이다.

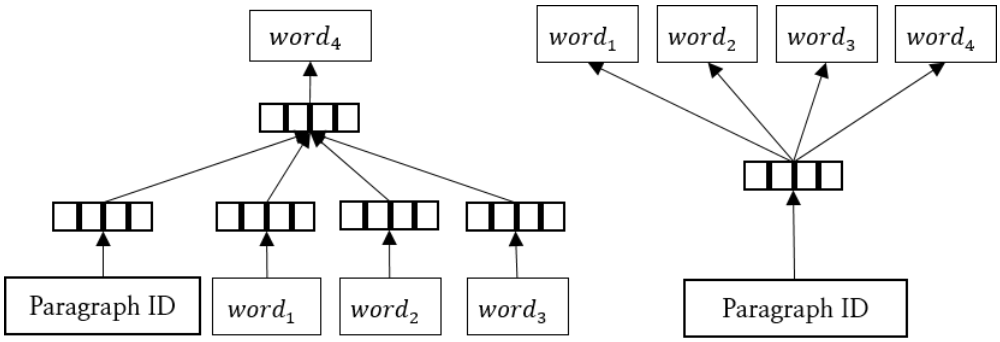


그림 3: Doc2Vec에서 제안한 PV-DM과 PV-DBOW 모델 구조

[7]에서는 PV-DM이 PV-DBOW보다 나은 감성 분류 성능을 보인다고 하였으며, 두 모델을 합칠 경우 대체로 가장 성능이 좋다고 하였다. 또한, 정보 검색(Information retrieval)에서도 Bag-of-words 모델에 비해 낮은 오차율을 보인다고 한다.

# 3 제안하는 방법

## 3.1 브랜드 분산 표상: Brand2Vec

본 연구에서는 문서를 벡터로 표현하는 Doc2Vec 방법을 발전시켜 브랜드를 벡터로 표현하는 Brand2Vec 방법을 제안한다. 기존의 Doc2Vec 방법이 각 문서를 하나의 벡터로 표현하는 방법이었다면 Brand2Vec은 동일한 브랜드에 대한 리뷰들을 각각 하나의 문서로 생각하고 학습하는 방법이다. Brand2Vec 모델의 구조는 그림 4와 같다. 주어진 window size 단어들과 해당하는 브랜드 벡터를 통해 다음 단어를 예측하는 모델로서, 단어와 브랜드를 동시에 학습하는 구조이다.

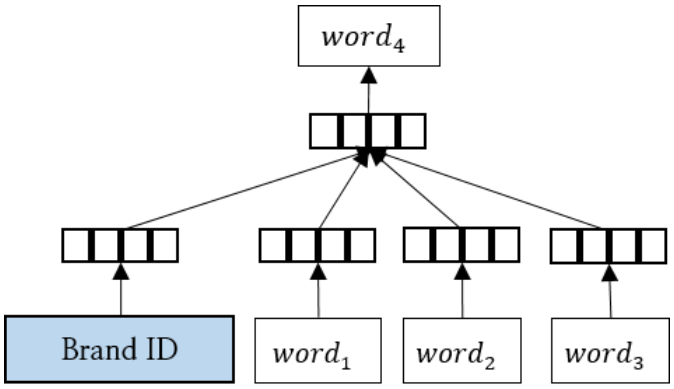

그림 4: 제안하는 Brand2Vec 모델 구조

Brand2Vec 모델의 목적함수는 다음과 같다.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_t|w_{t+j}) + \frac{1}{T_b}\sum_{i=1}^{T_b}\sum_{k=1}^{T_k}\log p(b_i|w_k) \tag{7}$$

Word2Vec과 마찬가지로 목적함수를 최대화 하도록 Backpropagation을 통해 학습한다. $T_b$는 브랜드의 개수이며, $T_k$는 해당 브랜드 $b_i$에 해당하는 리뷰의 단어 개수를 의미한다. 목적함수의 앞부분은 식 (1)의 CBOW 모델과 동일하나 뒷부분에 브랜드 벡터에 대한 계산이 추가되었다. 이 식의 의미는 리뷰에서 등장하는 단어($w_k$)를 통해, 해당 브랜드($b_i$)를 예측할 조건부 확률을 의미한다. 이러한 목적함수를 사용하여 Backpropagation을 통해 리뷰에 등장하는 단어와 브랜드 벡터를 동시에 학습할 수 있다.

### 3.2 Parameter 탐색

Word2Vec 방법은 단어 벡터 차원 수, window size, training epoch, negative sampling 적용 여부 등 다양한 parameter 들이 존재한다. 처음 Doc2Vec을 제안한 연구 [7]에서는 단어 차원 수와 window size에 대해 parameter 탐색을 하였다. 그리고 Word2Vec은 인공신경망을 활용한 방법이기 때문에 적당한 수의 training epoch이 중요하다. 따라서, 본 연구에서는 브랜드와 단어의 차원 수, window size, training epoch을 주요 parameter 탐색 대상으로 삼아 두 가지 실험을 하였다.

실험 1에서는 parameter에 따라 브랜드 벡터를 가지고 제품의 카테고리를 분류하는 성능을 확인한다. 브랜드 정보가 벡터에 잘 반영되었다면, 해당 브랜드가 주력으로 하는 카테고리를 분류할 수 있을 것이다. 예를 들어, Electronics, Clothing, Beauty 제품 카테고리가 있을 때, Samsung 브랜드는 Electronics로 LOREAL은 Beauty로 분류할 수 있을 것이다. 브랜드 차원 수, window size, training epoch에 따라서 Logistic Regression으로 분류 성능을 확인함으로써 브랜드 벡터가 잘 학습되었는지 확인한다.

그러나 Brand2Vec은 단어와 브랜드 벡터를 동시에 학습하기 때문에 실험 1을 통해서 브랜드 벡터가 잘 학습된 parameter를 찾더라도 단어 벡터들이 잘 학습되었는지 알 수 없다. 왜냐하면,

전체 데이터에서 브랜드 수와 단어 개수가 차이가 크게 나기 때문이다. 따라서 Doc2Vec 방법으로 각 리뷰 문서를 분산 표상 벡터로 표현하고 문서 벡터가 Electronics, Clothing 또는 Beauty 카테고리인지 분류하는 실험 2를 실행하였다. 실험 1과 마찬가지로 문서 차원 수, window size, training epoch에 따라 Logistic Regression으로 카테고리 분류 성능을 확인하였다.

### 3.3 시각화 방법 : 계층적 군집화 및 t-SNE

브랜드를 분류할 때 계층적으로 분류할 수 있다. 예를 들어, 브랜드를 제조업, 서비스업, 건설업과 같이 크게 분류할 수도 있지만, 제조업도 좀 더 세분화하여 자동차, 가구, 섬유 등 하위분류로 나눠서 묶을 수 있다. 본 연구에서는 이와 같은 계층적 구조를 시각화하기 위해 계층적 군집화 방법을 사용하였다. 계층적 군집화 방법은 크게 응집(agglomerative) 방법과 분할(divisive) 방법이 있다. 응집 방법은 bottom-up 방식으로 데이터 각각을 하나의 군집이라고 생각하고 합쳐가면서 군집 개수를 줄이는 방법이다. 반대로 분할 방법은 top-down 방식으로 전체 데이터를 하나의 군집으로 간주하고 작게 쪼개는 방식이다. 하지만 분할 방법은 총 $n$개의 데이터가 있을 때, $2^n - 1$가지 수로 나눌 수 있으므로, 경우의 수가 기하급수적으로 증가하는 문제가 있다.

따라서 본 연구에서는 응집 계층적 군집화를 실시하였다. 계층적 군집화에서 두 군집 간의 거리를 계산하는 방법은 군집 간의 데이터 중 최소 거리를 계산하는 단일연결(single linkage) 방법, 최대 거리를 계산하는 완전연결(complete linkage) 방법, 평균값을 계산하는 평균연결(average linkage) 방법 등이 있다. 본 연구에서는 군집화하면서 생기는 정보의 손실을 고려하는 ward's variance minimization algorithm [18]을 사용하였다. 새로운 데이터를 군집화할 때 오차제곱합(error sum of squares)을 최소화하도록 군집화를 진행하는 방법이다.

Ward 방법의 각 군집화 단계에서 군집 간의 거리를 계산하는 과정을 효율적으로 할 수 있는 방법을 Wishart[19]가 Lance-Williams 업데이트 공식[20]을 적용하여 제안하였다. 이 방법에서는 군집화 단계마다 클러스터 간의 거리를 아래와 같이 계산한다.

$$d(i \cup j, k) = \alpha_i d(i,k) + \alpha_j d(j,k) + \beta d(i,j) + \gamma |d(i,k) - d(j,k)| \tag{8}$$

이 때, $i$, $j$는 각 군집을 의미하며 $i \cup j$는 두 군집을 하나로 합치는 것을 의미한다. ward 방법에서 $\alpha_i = \frac{|i|+|k|}{|i|+|j|+|k|}$, $\beta = -\frac{|k|}{|i|+|j|+|k|}$, $\gamma = 0$ 이다. 또한, $|i|$는 군집 $i$의 크기를 의미한다.

이러한 계층적 군집화 과정은 dendrogram을 통해 시각적으로 이해할 수 있다. dendrogram은 계층적 군집화 단계에서 어떤 군집끼리 합쳤는지 확인할 수 있는 나무 구조의 시각화 방법이다. 예를 들어, 그림 5는 5개 지역의 상대적인 거리 정보 데이터를 이용하여 ward 방법을 적용한 dendrogram 이다. 이를 통해 Beijing, Seoul이 서로 가깝게 위치했을 뿐만 아니라 두 도시가 Paris, London보다 Tokyo와 가깝다는 계층적 정보도 시각적으로 확인할 수 있다.

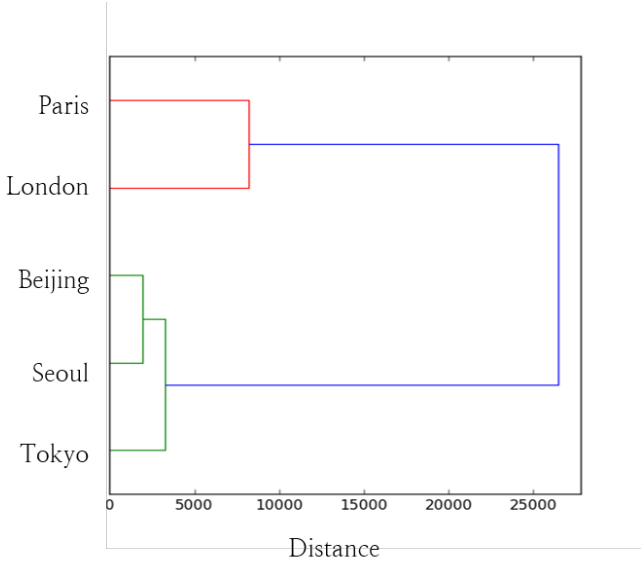


그림 5: Dendrogram 예시

그러나 계층적 군집화 방법을 통해 군집 간의 계층적 관계와 거리는 확인할 수 있지만, 다양한 브랜드의 상대적 유사도를 한눈에 파악하기 어렵다. 따라서 본 연구에서는 여러 브랜드의 포지셔닝을 확인하기 위해 t-SNE [21]를 사용하였다. t-SNE는 SNE(Stochastic Neighbor Embedding) [22]를 발전시킨 방법으로, PCA(Principal components analysis), LLE(Linear local embedding)등 기존의 차원축소 방법보다 고차원 데이터의 구조를 저차원에서 잘 표현한다. 따라서 딥러닝과 같이 고차원의 벡터를 사용하는 문제에서 많이 적용되고 있다[23, 24]. 본 연구에서 제안하는 Brand2Vec 또한 브랜드를 100에서 500차원으로 표현하기 때문에 t-SNE를 사용하면 고차원의 브랜드 벡터를 2차원으로 시각화하여 포지셔닝을 확인할 수 있다.

# 4 실험 결과 및 활용방안

## 4.1 데이터 설명

본 연구에서는 미국의 대표적인 종합 온라인 쇼핑몰 www.amazon.com 의 리뷰 데이터 [25]를 활용하였다. 데이터는 카테고리별로 리뷰 텍스트, 리뷰어 ID, 별점, 작성한 시간, 가격, 브랜드 등의 정보가 포함되어 있다. Word2Vec 방법을 활용하였기 때문에 소문자 변환, 특수문자 제거, 공백 단위 파싱 등 최소한의 전처리만 실시하였다.

먼저 parameter 탐색에 활용한 데이터는 다음과 같다. Electronics, Clothing, Beauty 각 카테고리에서 리뷰 수가 많은 3,000개의 브랜드를 선택하였다. 그리고 카테고리별 리뷰 수의 불균형을 해결하기 위해 50만 개씩 임의로 선택하여 총 150만 개의 리뷰 데이터를 활용하였다.

Application에 활용한 데이터는 Electronics 카테고리에 해당하는 리뷰만 선택하였으며, 2012년 이후 데이터만을 사용하였다. 총 9,557개의 브랜드가 있었으나, 리뷰가 10개 미만으로 달린 것을 제외하고, 총 5,079 개의 브랜드에 대한 2,944,904개의 리뷰 문서를 활용하였다. 총 사용된 토큰은

21,828,099개이고, 고유한 토큰은 886,768개이다. 한 개의 리뷰 문서는 평균적으로 74.12개의 토큰으로 구성되었다. Electronics 데이터의 브랜드 별 리뷰 개수 보면 그림 6와 같이 Logitech, Sony, Generic 순으로 많은 것을 알 수 있다.

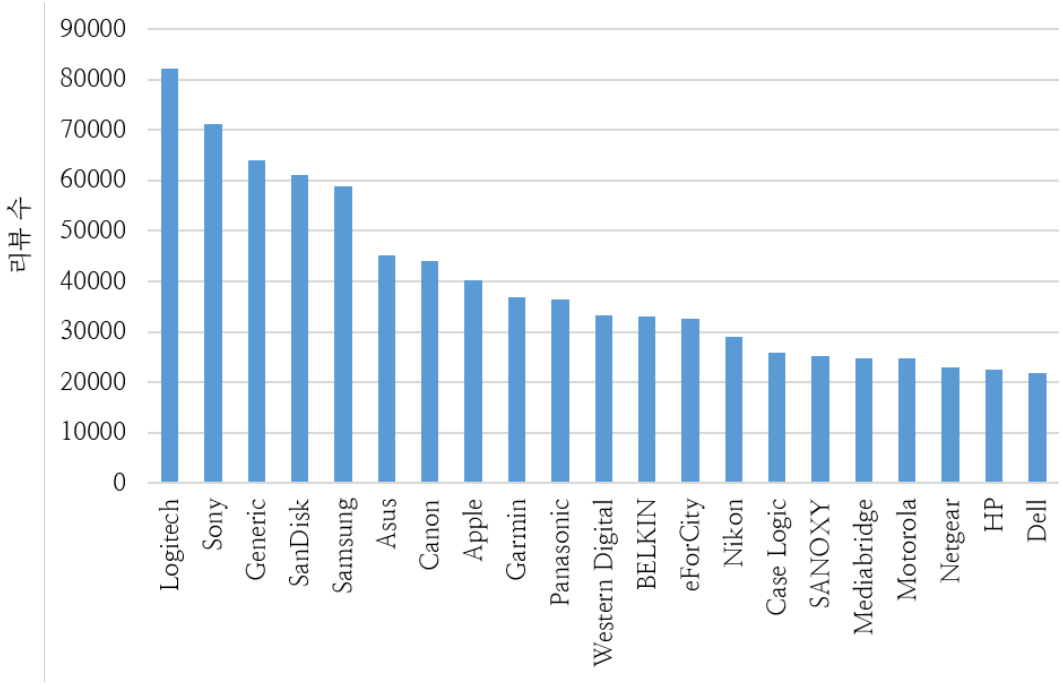


그림 6: Electronics 브랜드에 따른 리뷰 개수

### 4.2 Parameter 탐색 결과

Parameter 탐색을 위해 두 가지 실험을 하였다. 실험 1 에서는 제안한 Brand2Vec 방법으로 브랜드를 벡터로 표현하여 Logistic Regression 모델의 독립변수로 사용하고, 카테고리 정보를 종속변수로 하였다. Window size, 차원 수, epoch을 바꿔가며 10-fold 교차검증으로 분류 성능을 확인하였다. Training epoch을 10번으로 했을 때, Window size와 브랜드 벡터의 차원 수에 따른 실험 1의 결과는 표 1과 같다. 동일한 결과를 시각화한 결과는 그림 7과 같다.

표 1: Parameter에 따른 브랜드 분류 성능

| | | 브랜드 벡터의 차원 수 | | | | |
|---|---|---|---|---|---|---|
| | | 100 | 200 | 300 | 400 | 500 |
| Window Size | 4 | 0.9397 | 0.9390 | 0.9421 | 0.9440 | 0.9462 |
| | 6 | 0.9364 | 0.9405 | 0.9397 | 0.9443 | 0.9484 |
| | 8 | 0.9357 | 0.9399 | 0.9405 | 0.9402 | 0.9467 |
| | 10 | 0.9341 | 0.9389 | 0.9413 | 0.9407 | 0.9486 |
| | 12 | 0.9362 | 0.9404 | 0.9411 | 0.9406 | **0.9490** |

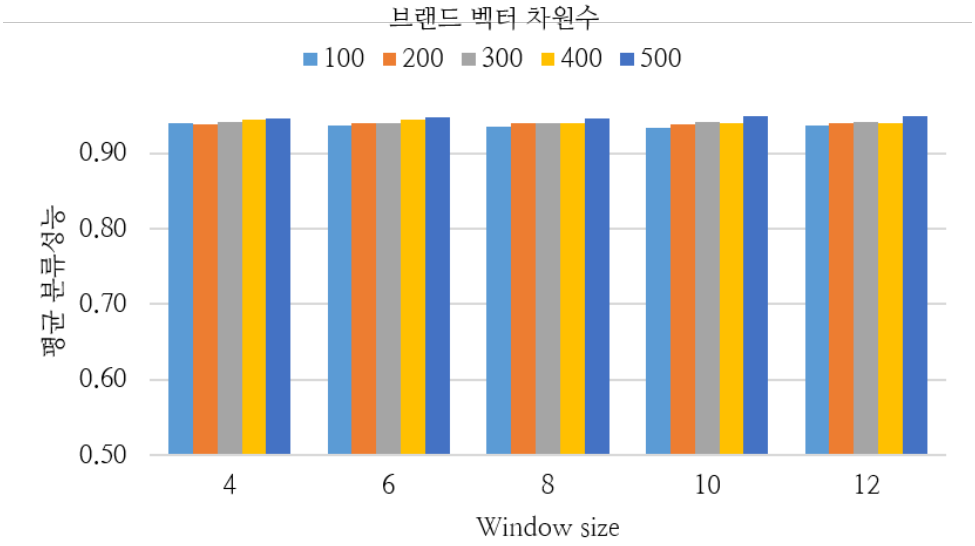


그림 7: Parameter에 따른 브랜드 분류 성능

Window size와 차원 수에 따라 약간의 성능 차이가 있지만, 분류 정확도가 93%에서 95%사이로 parameter에 따른 민감도가 매우 낮음을 알 수 있다. 게다가 3-class 분류 문제임에도 불구하고 분류 성능이 95% 정도로 매우 높은 것을 통해 브랜드 벡터가 해당 브랜드의 특징을 잘 반영하고 있음을 알 수 있다.

그중에서도 가장 성능이 좋았던 경우가 window size가 12이고, 차원 수가 500이었을 때 0.9490의 분류 정확도를 보였다. 해당 경우에 training epoch에 따른 성능 변화를 살펴보면 아래 그림 8과 같다. Epoch이 증가함에 따라 성능이 향상되며, 6번 이상이면 충분히 학습되는 것을 알 수 있다.

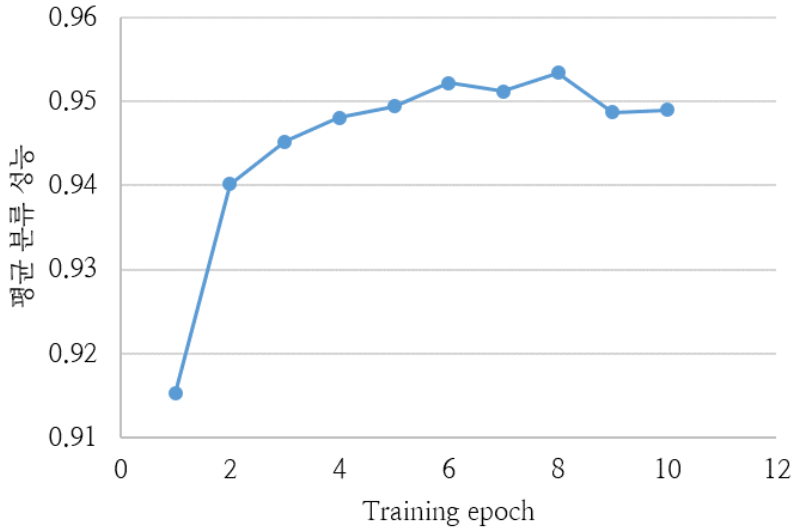


그림 8: Training epoch에 따른 분류 성능의 변화

실험 2 에서는 브랜드 단위가 아닌 문서 단위의 학습이 제대로 이루어졌는지 확인하기 위해 Doc2Vec 방법으로 각 리뷰 문서를 벡터로 표현하고, parameter에 따른 모델의 성능 변화를 10-fold 교차검증으로 확인하였다. Window size와 차원 수에 따른 실험 결과는 표 2 및 그림 9와 같다.

표 2: Parameter에 따른 문서 분류 성능

| | | 문서 벡터의 차원 수 | | | | |
|---|---|---|---|---|---|---|
| | | 100 | 200 | 300 | 400 | 500 |
| Window Size | 4 | 0.8503 | 0.8527 | 0.8558 | 0.8577 | **0.8589** |
| | 6 | 0.8400 | 0.8420 | 0.8445 | 0.8455 | 0.8473 |
| | 8 | 0.8287 | 0.8290 | 0.8307 | 0.8327 | 0.8335 |
| | 10 | 0.8155 | 0.8165 | 0.8178 | 0.8193 | 0.8202 |
| | 12 | 0.8047 | 0.8030 | 0.8046 | 0.8061 | 0.8074 |

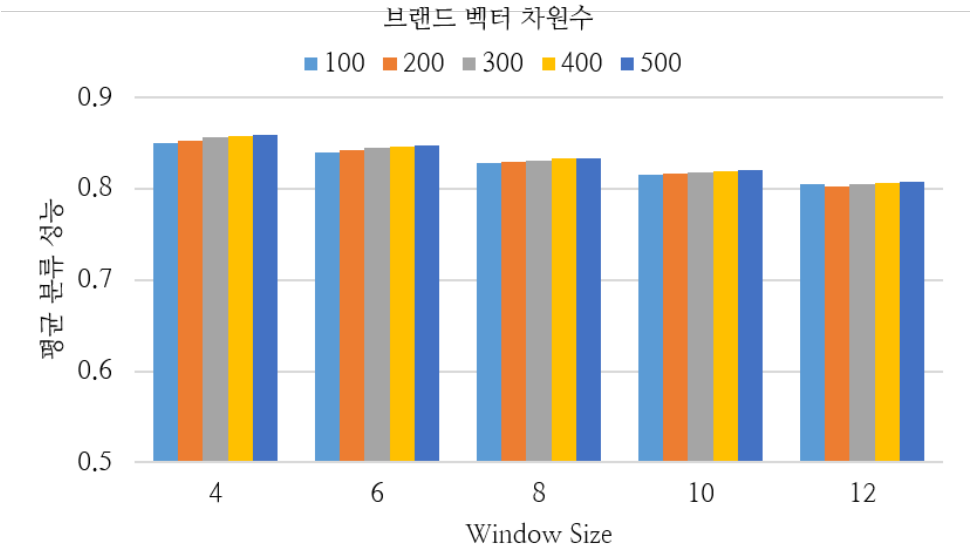


그림 9: Parameter에 따른 문서 분류 성능

실험 2의 경우도 실험 1과 마찬가지로 parameter에 따른 성능의 차이가 0.8에서 0.86으로 매우 적다. 실험 1과 마찬가지로 차원 수가 클수록 성능이 향상되었지만, window size는 작을 경우 성능이 좋음을 알 수 있다. 가장 성능이 좋았던 경우는 window size가 4이고, 차원 수가 500일 때였으며, 이 경우에 training epoch에 따른 성능 변화는 그림 10과 같다. Epoch이 커짐에 따라 문서를 분류하는 성능이 높아지는 것으로 보아 학습이 잘 되고 있음을 알 수 있다.

실험 1과 실험 2의 결과를 통해 차원 수가 클 경우에 브랜드와 문서를 모두 잘 표현하는 것을 알 수 있다. 반면, 문서 단위로 표현할 경우 window size가 작을 경우가 좋지만, 브랜드 단위로 표현하면 window size가 클 경우에 성능이 좋았다. 두 경우를 모두 고려하여 차원 수는 500으로 선택하고, window size 는 4와 12의 중간값인 8을 선택하였다. Training epoch은 10번 정도면 두 경우 모두 충분히 학습되는 것 같다고 판단하여, 10번으로 선택하였다.

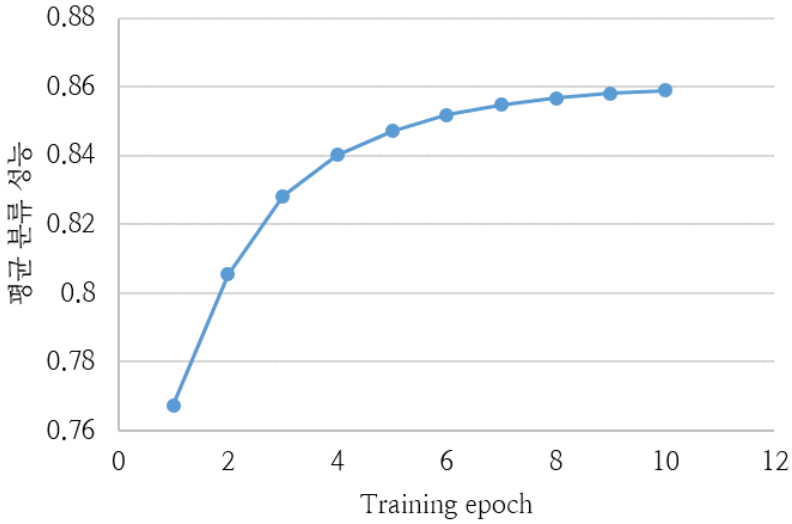


그림 10: Parameter에 따른 문서 분류 성능

## 4.3 Applications

### 4.3.1 Brand Visualization

Parameter 탐색 결과를 바탕으로 Electronics 카테고리 제품 약 3백만 개 리뷰에 대해서 Brand2Vec 모델링을 실시하였다. 각각의 브랜드를 하나의 벡터로 표현하였기 때문에 코사인 유사도(cosine similarity)를 이용하여 브랜드 간의 유사도 계산이 가능하다. 예를 들어 Samsung과 Canon 브랜드 벡터와 유사한 브랜드를 나열한 결과는 표 3과 같다.

표 3: Samsung, Canon 과 유사한 브랜드

| | **Samsung** | | **Canon** | |
|---|---|---|---|---|
| | 브랜드 벡터 | 유사도 | 브랜드 벡터 | 유사도 |
| 1 | Acer | 0.5991 | Nikon | 0.8371 |
| 2 | Toshiba | 0.5541 | Focus Camera | 0.7949 |
| 3 | Proscan | 0.5025 | Fujifilm | 0.7776 |
| 4 | Lenovo | 0.4961 | Sigma | 0.7640 |
| 5 | Sharp | 0.4939 | Pentax | 0.7629 |
| 6 | HP | 0.4935 | Tamron | 0.7615 |
| 7 | TCL | 0.4666 | Tokina | 0.7094 |
| 8 | Dell | 0.4654 | Rokinon | 0.7041 |
| 9 | VIZIO | 0.4636 | SSE | 0.6905 |
| 10 | Kocaso | 0.4568 | Vivitar | 0.6746 |

표 3을 통해 Samsung과 유사한 브랜드로 Acer, Toshiba 등 비슷한 제품을 생산하는 브랜드가 같이 등장함을 알 수 있다. 또한, 카메라 제품을 생산하는 Canon 브랜드와 함께 Nikon, Focus Camera 등 비슷한 제품을 생산하는 브랜드들이 유사한 벡터로 표현됨을 알 수 있다.

추가적으로 Brand2Vec 방법은 브랜드뿐만 아니라 단어도 벡터로 표현하기 때문에 단어와 브랜드 벡터 사이에도 유사도 계산이 가능하다. 예를 들어 *computer*, *desktop* 두 단어 벡터의 평균값과 상위 50개 브랜드의 코사인 유사도를 계산하여 유사도가 높은 순으로 정렬하면 표 4와 같다.

표 4: (*computer* + *desktop*)/2 값과 유사한 브랜드

| | **브랜드** | **유사도** |
|---|---|---|
| 1 | Dell | 0.1418 |
| 2 | StarTech | 0.1229 |
| 3 | SIB | 0.1040 |
| 4 | HP | 0.1035 |
| 5 | Cooler Master | 0.0996 |

비슷하게 *earphone*, *headphone* 단어 벡터의 평균값과, *camera*, *cameras* 단어 벡터의 평균값에 각각 유사한 브랜드는 표 5와 같다. 표 5의 (a)에 Sennheiser, Monster 모두 이어폰과 헤드폰을 주로 판매하는 브랜드이며, (b)에서 Canon, Nikon과 같은 카메라 제조 브랜드가 등장함을 알 수 있다. 표 4와 표 5를 통해 브랜드 벡터가 제품의 속성을 내포하고 있으며, 브랜드 벡터와 단어 벡터를 같은 공간에서 비교할 수 있음을 알 수 있다.

표 5: 단어 벡터 평균값과 유사한 브랜드

(a) (*earphone*+*headphone*)/2와 유사한 브랜드

| | **브랜드** | **유사도** |
|---|---|---|
| 1 | Sennheiser | 0.2117 |
| 2 | Monster | 0.1875 |
| 3 | JVC | 0.1456 |
| 4 | Monoprice | 0.1346 |
| 5 | Bose | 0.1233 |

(b) (*camera*+*cameras*)/2와 유사한 브랜드

| | **브랜드** | **유사도** |
|---|---|---|
| 1 | Canon | 0.1991 |
| 2 | Nikon | 0.1873 |
| 3 | Neewer | 0.1410 |
| 4 | Case Logic | 0.0779 |
| 5 | Panasonic | 0.0748 |

좀 더 많은 브랜드의 상대적인 위치를 확인하기 위해 리뷰가 많은 상위 50개 브랜드의 브랜드 벡터를 대상으로 Ward 방법을 적용한 응집 계층적 군집화와 t-SNE를 통해 시각화하였다. 먼저 각각의 브랜드 벡터에 대해 코사인 유사도를 계산하여 pair-similarity 매트릭스를 만들고 계층적 군집화 결과를 그림 11과 같이 dendrogram으로 나타내었다.

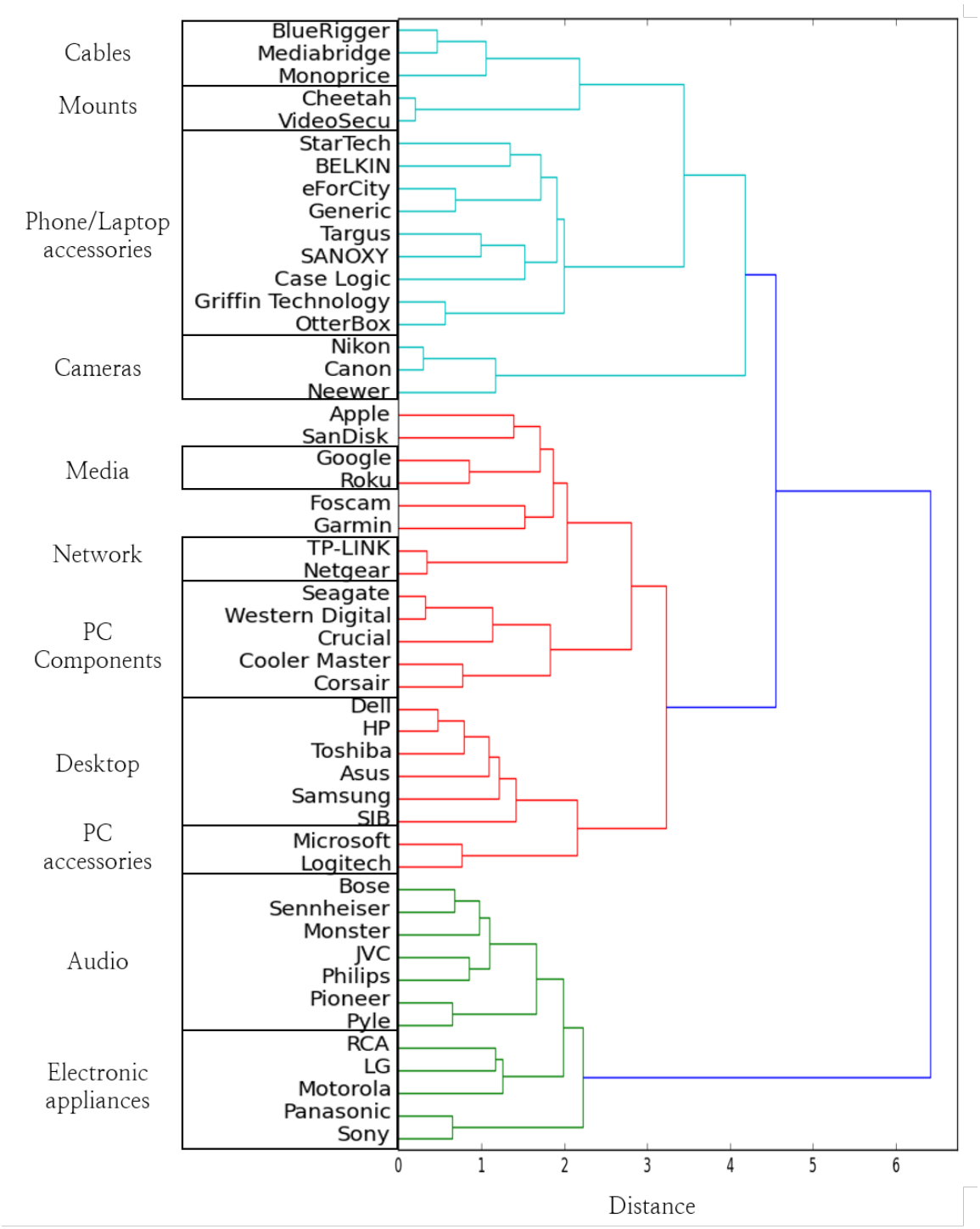


그림 11: Ward 방법을 이용한 계층적 군집화 결과

그림 11의 좌측 상단에 보면 BlueRigger, MediaBridge, Monoprice 등 HDMI 케이블, USB 케이블 등을 만드는 회사들이 가까이 위치한 것을 알 수 있다. Cheetah, VideoSecu 브랜드는 TV 거치대(Mounts)를 주로 판매하는 브랜드이다. 따라서 각각 `Cables`, `Mounts`라고 각각 군집 이름을 정했다. 두 군집의 브랜드 모두 TV나 PC의 부속품을 생산하는 브랜드라 유사하게 분류할 수 있는

데, dendrogram 상에서도 다른 군집에 비해 거리가 가까운 것을 확인할 수 있다. 비슷하게 Dell, HP 등 컴퓨터를 주로 생산하는 브랜드 군집을 `Desktop`이라 하고, Microsoft, Logitech 등 마우스나, 키보드 등 컴퓨터 주변 기기를 주로 판매하는 브랜드를 `PC accessories`라고 이름 지을 수 있다. 두 그룹 역시 dendrogram 상에서 다른 그룹에 비해 거리가 가까움을 알 수 있다. 이처럼 브랜드 벡터를 이용하여 계층적 군집화를 실시하면 각 브랜드 간의 유사도뿐만 아니라 브랜드 간의 계층적 정보도 확인할 수 있다.


그림 12: t-SNE 를 이용한 50개 브랜드 벡터 시각화

계층적 군집화 방법보다 더 많은 브랜드 간의 관계를 시각화하기 위해서 그림 12와 같이 t-SNE를 사용하여 50개 브랜드를 2차원으로 표현하였다. Dendrogram과 마찬가지로 비슷한 제품을 생산하는 브랜드끼리 가까이 위치한 것을 알 수 있다. Cheetah, VideoSecu와 같이 TV 거치대를 판매하는 브랜드와 Pyle, Pioneer와 같이 Car Audio를 주로 생산하는 브랜드가 매우 가깝게 위치한 것을 통해 특정 제품에 특화된 브랜드끼리 가깝게 위치한 것을 확인할 수 있다. 또한, Apple이 데스크탑 브랜드와 묶이지 않고 BELKIN, Generic, SANOXY처럼 핸드폰 Accessory 브랜드와 비슷한 것도 확인할 수 있다. 일반적으로 Apple의 경쟁사로 Samsung을 생각하지만, 실제 소비자의 인식에서 Apple은 Samsung보다 Accessory 브랜드와 더 유사함을 알 수 있다. 이처럼 브랜드 벡터를 사용하여 t-SNE를 적용하면 dendrogram 보다 브랜드 간의 유사도를 쉽게 확인할 수 있고, 소비자들의 리뷰 정보를 활용한 객관적인 브랜드 지각도를 그릴 수 있다.

#### 4.3.2 키워드 추출

앞 절에서 살펴보았듯이 Brand2Vec은 브랜드와 단어를 같은 공간에 표현하기 때문에, 단어와 브랜드 사이의 유사도 계산이 가능하다. 표 6은 Samsung, Canon 브랜드 벡터와 유사한 단어 벡터를 나열한 것이다. Samsung 브랜드 벡터와 가장 유사한 단어로 “samsung”, “samsung’s”, “smart”가 등장하였고, Canon 브랜드 벡터와 가장 유사한 단어로 “canon”, “zoom” 등의 용어가 등장하였다. 이를 통해 각각의 브랜드 벡터가 해당 브랜드의 제품 특징을 잘 반영하고 있는 것을 알 수 있다.

표 6: Samsung, Canon 브랜드 벡터와 유사한 단어

| **Samsung** | | **Canon** | |
|---|---|---|---|
| 단어벡터 | 유사도 | 단어벡터 | 유사도 |
| samsung | 0.3117 | canon | 0.4292 |
| samsung’s | 0.2840 | zoom | 0.4077 |
| smart | 0.2519 | dslr | 0.4064 |
| google | 0.2493 | telephoto | 0.3893 |
| ativ | 0.2316 | nikon | 0.3769 |
| lg | 0.2297 | slr | 0.3765 |
| series | 0.2272 | bodies | 0.3626 |
| plasma | 0.2257 | megapixel | 0.3525 |
| smarttv | 0.2242 | macro | 0.3479 |
| viera | 0.2187 | point-and-shoot | 0.3393 |

실제 마케팅에서 중요한 것은 경쟁사와 비교했을 때 상대적인 자사 브랜드의 특징을 찾는 것이다. 위의 방법을 조금 발전시켜 같은 단어에 대해 다른 브랜드와 자사 브랜드 간의 상대적 거리를 계산하면 자사 브랜드의 특징적인 단어를 추출할 수 있다.

표 7: 500개 형용사 중 상위 10개 목록

| | 단어 | 빈도 |
|---|---|---|
| 1 | great | 9,071 |
| 2 | good | 7,610 |
| 3 | easy | 3,794 |
| 4 | new | 2,863 |
| 5 | cable | 2,434 |
| 6 | nice | 2,143 |
| 7 | small | 2,100 |
| 8 | first | 2,022 |
| 9 | old | 2,020 |
| 10 | best | 1,818 |

표 8: 분석 대상으로 삼은 9개 브랜드

| | 브랜드 | 리뷰 수 |
|---|---|---|
| 1 | Samsung | 58,852 |
| 2 | Asus | 45,222 |
| 3 | Apple | 40,282 |
| 4 | HP | 22,582 |
| 5 | Dell | 21,727 |
| 6 | Microsoft | 18,437 |
| 7 | Toshiba | 12,645 |
| 8 | Acer | 9,589 |
| 9 | Lenovo | 9,908 |

본 연구에서는 리뷰에서 일반적으로 많이 사용되는 형용사를 추출하기 위해서 임의로 추출한 3만 개의 리뷰에서 많이 사용되는 형용사 500개를 추출하였다. 그 중 대표적인 형용사는 표 7과 같다. 분석 대상으로 삼은 브랜드는 표 8과 같이 대표적인 데스크탑 관련 브랜드 9개를 선택하였다.

각 형용사와 브랜드 간의 상대적 거리를 계산하기 위해 다음과 같은 거리 척도를 도입하였다. 먼저, 추출한 500개의 형용사에 해당하는 벡터 표현 집합을 $A \in \{vector(great), vector(good), \ldots\}$ 라고 하자. 그리고 비교하고 싶은 브랜드 벡터 집합을 $B \in \{vector(Samsung), vector(Asus), \ldots\}$ 라고 하자. 이 때, 단어와 브랜드 간의 거리 척도는 식 9와 같다.

$$P(B_i|A_j) = \frac{\exp(B_i \cdot A_j)}{\sum_{brands} \exp(B_{brand} \cdot A_j)} \tag{9}$$

제안한 거리 척도값은 0과 1 사이의 값을 가진다. 그리고 어떤 $X$ 브랜드에 대해 거리 척도값이 높은 단어는 타 브랜드에 비해서 $X$ 브랜드와 거리가 가깝다는 것을 의미한다.

대표적으로 Apple, Microsoft 브랜드에 대해 척도값이 1에 가까운 키워드들을 추출한 결과는 표 9와 같다. 표 9의 3번째 열에 해당하는 '빈도'는 해당 단어가 각 브랜드 리뷰에서 나온 횟수를 의미한다. 500개의 형용사 중 타 브랜드에 비해 Apple 브랜드와 가까운 단어는 "air", "classic", "cute" 등이었다. "air", "classic" 등 Apple 제품 관련된 형용사들이 등장한 것으로 보아 Apple 브랜드의 특징을 잘 나타내는 단어들이 추출됨을 알 수 있다. 그리고, "compatible", "handy", "sturdy", "heavier" 등 제품의 특징 관련된 이야기들도 많이 등장하므로 제안한 방법을 통해 Apple 제품의 특징을 파악할 수 있다. 또한, "magnetic", "popular", "stronger" 등 상대적으로 빈도수가 낮더라도 브랜드의 특징을 나타내는 단어를 추출할 수 있다.

표 9: 제안한 방법으로 추출한 키워드

(a) Apple 브랜드 키워드

| **Apple** | | |
|---|---|---|
| 단어 | 척도값 | 빈도 |
| air | 0.99 | 2,144 |
| classic | 0.99 | 706 |
| cute | 0.99 | 118 |
| magnetic | 0.99 | 89 |
| proprietary | 0.99 | 121 |
| popular | 0.99 | 80 |
| compatible | 0.99 | 400 |
| magic | 0.99 | 392 |
| versatile | 0.99 | 95 |
| similar | 0.99 | 384 |
| white | 0.99 | 587 |
| handy | 0.99 | 088 |
| short | 0.99 | 524 |
| sturdy | 0.99 | 221 |
| stronger | 0.99 | 60 |
| substantial | 0.99 | 38 |
| heavier | 0.99 | 126 |
| impossible | 0.99 | 143 |
| protective | 0.99 | 158 |

(b) Microsoft 브랜드 키워드

| **Microsoft** | | |
|---|---|---|
| 단어 | 척도값 | 빈도 |
| vertical | 0.99 | 99 |
| stiff | 0.99 | 165 |
| closer | 0.99 | 112 |
| comfortable | 0.99 | 2,034 |
| traditional | 0.99 | 195 |
| key | 0.99 | 2,669 |
| natural | 0.99 | 786 |
| mechanical | 0.99 | 225 |
| couch | 0.99 | 139 |
| ergonomic | 0.99 | 1,403 |
| responsive | 0.99 | 558 |
| soft | 0.99 | 360 |
| harder | 0.99 | 158 |
| smooth | 0.99 | 581 |
| sensitive | 0.99 | 308 |
| uncomfortable | 0.99 | 235 |
| regular | 0.99 | 558 |
| love | 0.99 | 2,991 |
| easier | 0.99 | 508 |

Microsoft의 경우 “vertical”, “stiff”, “ergonomic” 등 전반적으로 마우스나, 키보드 관련된 형용사들이 등장하였다. 이와 같은 단어가 등장한 이유는 일반적으로 Microsoft는 윈도우 7과 같은 PC 운영체제를 만드는 회사로 인식하고 있으나, 본 연구에서는 Electronics 카테고리의 제품 리뷰만을 대상으로 했기 때문이다. Microsoft의 경우 Electronics 카테고리 내에서는 키보드나 마우스 제품이 두드러지는 브랜드임을 알 수 있다.

표 9와 같은 단어가 추출된 이유를 확인하기 위해 PMI(Pointwise Mutual Information) [26]를 계산하여 함께 자주 등장하는 단어를 추출하였고, 실제 원문을 확인해 보았다. 표 10는 Apple 브랜드 키워드 중 일부 대해서 PMI가 높은 단어 중 유의미한 단어와, 실제 원문을 정리한 결과이다.

“air”, “classic” 등의 단어는 `Macbook air`, `iPod classic` 등 Apple 제품명이므로 Apple 브랜드를 나타내는 키워드임을 쉽게 알 수 있다. “compatible” 단어에 대해서는 리뷰에서 “cable”, “not” 등의 단어가 상대적으로 자주 등장하였다. 해당 단어가 등장한 원문을 살펴보면 사용자들이

호환성에 관련된 문제를 많이 이야기하는 것을 알 수 있다. “sturdy”와 “protective” 는 서로 상반되는 의미일 수도 있는데, 실제 사용된 문맥을 보면 디자인 측면에서 iPad나 iPhone의 메탈소재가 튼튼한(sturdy) 느낌을 주지만, 한편으로는 내구성 문제 때문에 케이스가 꼭 필요함을 이야기하고 있다.

Microsoft 경우에 PMI가 높은 주변 단어와 원문을 확인해 본 결과가 표 11와 같다. `Vertical scrolling` 같은 마우스 기능에 대한 이야기가 많이 등장하였으며, “stiff”, “ergonomic” 같은 경우는 키보드의 디자인에 대한 언급임을 알 수 있다. 원문을 살펴보면 “stiff” 같은 경우는 사용자의 불만이 많이 등장하는 반면, “ergonomic”은 긍정적으로 사용하는 것을 알 수 있다.

표 10: Apple 브랜드의 키워드와 해당 원문

| 키워드 | 주변 단어 | 실제 원문 |
|---|---|---|
| air | macbook, ipad, pro | “it has proven to be a great purchase the macbook **air**”<br>“Being able to see whatever is on my ipad with **air** play can also access all movies music” |
| compatible | cable, not | ”USB connection port is not **compatible** with the connector in my car.”<br>“Apple TV is not **compatible** with amazon.com” |
| sturdy | feels, seems | “Apple uses aluminum for the back cover of their tablet instead of plastic. It feels **sturdy**.”<br>“The device is incredibly thin and light though it still feels very **sturdy**.” |
| protective | case, cover | “You’ll want a **protective** case and some screen protectors to keep your ipad looking fresh.”<br>“Make sure you get a **protective** cover for it. so it doesn’t get scratched up.” |

표 11: Microsoft 브랜드의 키워드와 해당 원문

| 키워드 | 주변 단어 | 실제 원문 |
|---|---|---|
| vertical | horizontal, scrolling | “This mouse really helps windows become more usable especially switching through apps **vertical** and horizontal scroll”<br>“I think it’s because the ratio of **vertical** movement to horizontal movement is different from my current mouse” |
| stiff | bar, buttons | “I just returned a microsoft arc mouse because the buttons were too **stiff**”<br>“I am unable to type with this keyboard because the space bar is so **stiff**” |
| ergonomic | keyboard, shape | “**Ergonomic** shape is all what I was concerned and I managed to meet my needs”<br>“I’ve always wanted to try the **ergonomic** style keyboards and the price was right” |

# 5 결론 및 의의

본 연구에서는 UGC를 활용하여 브랜드를 분산 표상 벡터로 표현하는 방법인 Brand2Vec 방법을 제안하였다. 먼저, 벡터 차원 수, window size, training epoch 등의 parameter에 대해 브랜드 벡터로 카테고리를 분류하는 실험 1과 문서 벡터로 카테고리를 분류하는 실험 2를 통하여 parameter 탐색을 하였다. parameter에 따른 분류 정확도의 차이가 적은 것을 통해 Brand2Vec 방법론이 parameter에 강건함을 확인하였다.

브랜드 시각화는 dendrogram과 t-SNE를 사용하였다. Dendrogram을 통해 계층적 군집화 과정을 시각화하였다. 이를 통해, Dell, HP 등의 브랜드가 브랜드 단위로 유사할 뿐만 아니라, Dell, HP가 속해있는 `Desktop` 브랜드 군집이 Logitech, Microsoft 등이 속해있는 `PC accessories` 브랜드 군집과 유사함을 확인할 수 있었다. t-SNE를 통해서는 다수 브랜드의 상대적 거리를 2차원으로 시각화하여 브랜드들의 포지셔닝을 확인하였다. 기존의 방법과 달리 소비자가 브랜드에 대해 언급한 모든 텍스트 정보를 반영한 객관적인 지각도를 그릴 수 있었다.

또한, Brand2Vec 방법은 브랜드뿐만 아니라 단어도 같은 공간에 표현하기 때문에 브랜드와 단어 간의 유사도 계산이 가능하다. 본 연구에서는 이러한 성질을 활용하여 타 브랜드에 비해 두드러지는 키워드를 추출할 수 있는 방법을 제안하였다. Apple은 “air”, “compatible”, “sturdy” 등의 단어를 추출할 수 있었으며, Microsoft는 “vertical”, “scroll”, “stiff” 등 키워드를 추출하였고, 원문을 확인하여 정성적으로 검증하였다.

본 연구의 한계는 다음과 같다. Brand2Vec은 브랜드를 하나의 벡터로 표현하였기 때문에 브랜드의 다양한 차원에 대해 분석하지 못하였다. 일반적으로 소비자들은 브랜드를 다차원으로 인식한다. 예를 들어 브랜드의 가격, 품질, 서비스 등 다양한 측면에 대해 소비자들의 생각이 있을 수 있다. 이러한 문제는 향후 토픽 모델링 방법을 적용하여 보완할 수 있을 것이다. 또한, 각 브랜드에 대한 감성 정보를 모델에 반영하지 못했다. 리뷰 데이터에는 별점 정보가 포함되어 있으므로 각 리뷰 문서가 브랜드에 대해 긍정적인 의견인지 부정적인 의견인지 유추할 수 있다. 추가로 별점 정보를 브랜드와 같이 모델에 반영한다면 긍정, 부정 정보도 시각화할 수 있을 것이다.

향후 시간에 따른 브랜드 인식변화를 확인하기 위해 서로 다른 시간대의 문서를 학습한다면 동적 분석이 가능할 것이다. 또한, 동일한 방법으로 리뷰 데이터 뿐만 아니라 Facebook, Youtube 등 각종 소셜미디어를 활용할 수도 있을 것이다. 그리고 브랜드뿐만 아니라 영화, 연예인, 정치인, 운동선수 등도 UGC 상에서 활발히 언급되기 때문에 벡터로 표현하여 의사결정에 활용할 수 있을 것으로 기대한다.

## Acknowledgements

This work was supported by the BK21 Plus Program(Center for Sustainable and Innovative Industrial Systems, Dept. of Industrial Engineering, Seoul National University) funded by the Ministry of Education, Korea (No. 21A20130012638), the National Research Foundation(NRF) grant funded by the Korea government(MSIP) (No. 2011-0030814), and the Institute for Industrial Systems Innovation of SNU.

## 참고문헌

[1] Dong Jin Kim, Woo Gon Kim, and Jin Soo Han. A perceptual mapping of online travel agencies and preference attributes. *Tourism management*, 28(2):591–603, 2007.

[2] I-Ping Chiang, Chih-Ying Lin, and Kaisheng M Wang. Building online brand perceptual map. *CyberPsychology & Behavior*, 11(5):607–610, 2008.

[3] Paul Dwyer. Inferring brand proximities from user-generated content. *Journal of Brand Management*, 19(6):467–483, 2012.

[4] Mignon Reyneke, Leyland Pitt, and Pierre R Berthon. Luxury wine brand visibility in social media: an exploratory study. *International Journal of Wine Business Research*, 23(1):21–35, 2011.

[5] Marie-Francine Moens, Juanzi Li, and Tat-Seng Chua. *Mining user generated content*. CRC Press, 2014.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[7] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[8] Seshadri Tirunillai and Gerard J Tellis. Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation. *Journal of Marketing Research*, 51(4):463–479, 2014.

[9] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5:3, 1988.

[10] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

[11] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[12] Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Honza Cernocky, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE, 2011.

[13] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013.

[14] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284, 2015.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[16] Gerard Salton. Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*, 1989.

[17] Zellig S Harris. Distributional structure. *Word*, 1954.

[18] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[19] David Wishart. Mode analysis: A generalization of nearest neighbor which reduces chaining effects. *Numerical taxonomy*, 76(282-311):17, 1969.

[20] G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies: 1. hierarchical systems. *The Computer Journal*, 9(4):373–380, 1967.

[21] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[22] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 833–840, 2002.

[23] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 1, pages 740–750, 2014.

[24] Abdel-rahman Mohamed, Geoffrey Hinton, and Gerald Penn. Understanding how deep belief networks perform acoustic modelling. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4273–4276. IEEE, 2012.

[25] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

[26] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.