Brand2vec : Distributed representation of Brands and Applications

서울대학교 산업공학과
석사과정 양호성

Word Cloud by http://www.tagxedo.com/

0 Motivation

Q. 모든 Review 정보를 반영하여 브랜드를 하나의 ‘벡터’로 표현 할 수 없을까?

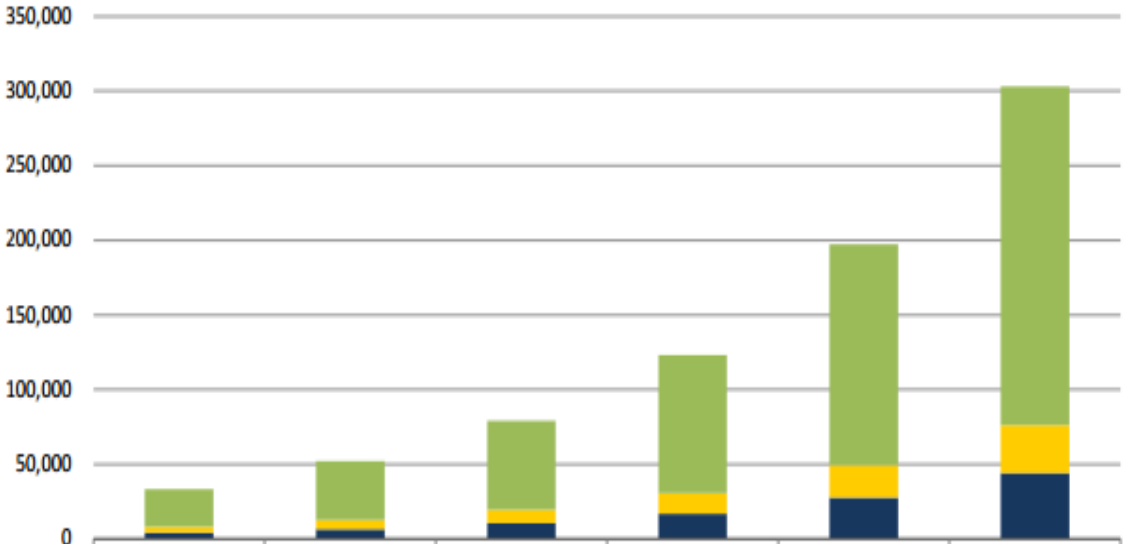
Q. 그리고 그 ‘벡터‘를 이용해 Business 적인 Application에 적용할 수 없을까?

Doc2vec과 Class2vec을 활용한 **Brand2vec**을 제안하고,
다양한 방법으로 Brand Vector Representation 의 활용방안을 제안한다

1 Introduction

Internet의 발전으로 누구나 쉽게 자신의 의견을 온라인 공간상에서 표출할 수 있게 됨에 따라, 비정형 데이터가 급증함.
이러한 변화에 맞춰, 경영진들은 기존의 설문조사 방식의 단점을 보완하고자
빅데이터 텍스트 분석을 통하여 소비자들의 생생한 의견을 분석하고자 함

Total Worldwide Digital Archive Capacity, by Content Type, 2010-2015 (Petabytes)



	2010	2011	2012	2013	2014	2015
File	25,127	39,237	59,600	92,536	147,885	226,716
Database	4,065	6,179	9,140	13,824	21,532	32,188
E-mail	4,025	6,575	10,411	16,796	27,817	44,091

*File = File-based or Unstructured data

Source : Enterprise Strategy Group, 2010

설문조사

- 통계적인 분석이 쉽고, 원하는 질문에 대한 결과를 얻기 용이함

- 표본 수 제한 및 많은 비용 발생
- 특정 설문 환경에서 실시하기 때문에 조사자의 주관이 개입

Review / Social Media

- 표본 수가 매우 많아 bias를 줄일 수 있음
- 소비자들의 실제 목소리를 반영

- 데이터가 매우 방대함
- noise가 심해 분석이 어려움
- '텍스트' 데이터를 분석하기 어려움

1 Introduction

기존에도 Social Media 데이터를 활용하여 경영전략 수립에 도움을 줄 수 있는 연구들이 이루어 졌으며,

특히, 텍스트 데이터 중 Social Media 데이터를 분석하는 것을 Opinion mining 이라고도 부르면서 소비자들로부터 제품이나 브랜드에 대한 유의미한 반응을 찾아내어 전략을 수립하기 위해 활용하고 있다.

관련 연구

- 다양한 Text mining 기법을 활용하여 Business 분야에 적용하는 연구가 이뤄짐.
- 특정 브랜드의 Social Media 페이지 정보를 활용하거나[2], 트위터에서 브랜드의 sentiment 정보를 이용하거나[4], LDA(Latent Dirichlet Allocation)을 이용한 방법[7] 등을 통해 보다 나은 의사결정을 위한 방법을 제시

관련 실제 적용 사례

- SKT에서는 Social Media Buzz량을 통하여 광고 효과를 분석[1)]
- LG 전자는 Social Media 데이터를 이용해 제품 개발 프로세스에 적용함.[2)]
- 다음소프트, 솔트룩스, LG CNS 등은 자체 Social Media Text 분석 시스템을 구축하여 다양한 산업군에 적용하고 있음.

1) http://www.bizwatch.co.kr/pages/view.php?uid=8966
2) www. etnews.com/20141124000268

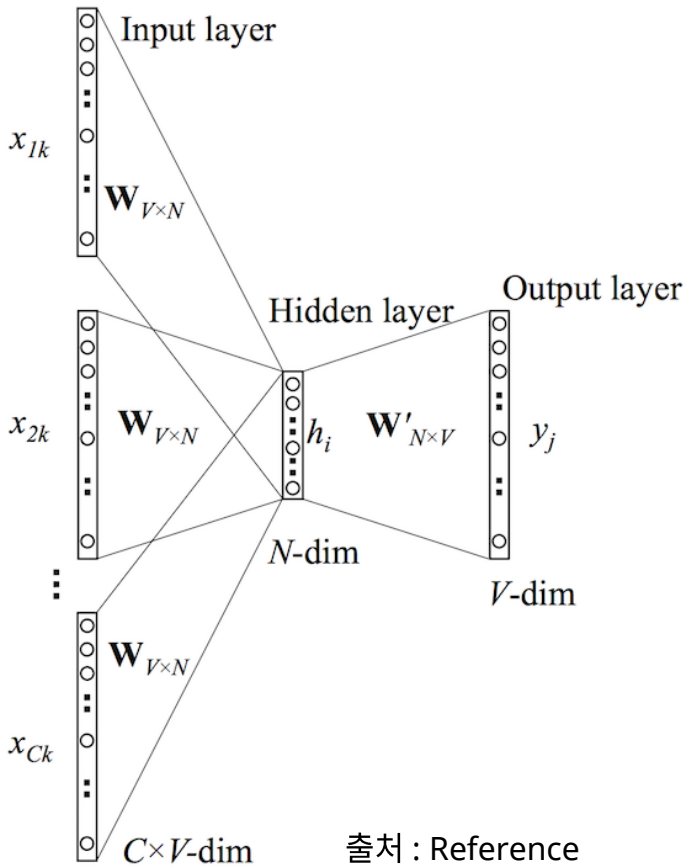
2 Word Representation

Text를 분석하기 위해서는 Text를 '숫자' 로 변환하는 과정이 필요하다.
단어를 숫자로 표현하는 방법은 크게 단어의 등장 빈도를 계산하는 Discrete 한 방법과,
Neural network 등을 통해 Continuous 하게 나타내는 방법이 있다.

Discrete representation	Distributed Representation
예) One-hot vector Word-word / Word-Document Matrix $dog = \begin{bmatrix}1\\0\\0\end{bmatrix}\quad cat = \begin{bmatrix}0\\1\\0\end{bmatrix}\quad pig = \begin{bmatrix}0\\0\\1\end{bmatrix}$	예) Word2vec, Glove $dog = \begin{bmatrix}1.5\\0.3\\0.8\end{bmatrix}\quad cat = \begin{bmatrix}1.8\\1.1\\0.2\end{bmatrix}\quad pig = \begin{bmatrix}1.6\\-2.3\\-1.5\end{bmatrix}$
- 전체 문서 혹은 특정 범위에서 등장한 단어의 비율 / 확률을 계산하는 방법 - 단어와 단어 사이의 'similarity' 비교가 불가능 - 문맥 상 중요한 단어더라도 '빈도'가 낮으면 무의미한 단어로 분석	- Neural Network 를 통해 단어를 continuous 한 vector 로 변환 - Bengio[1] 가 처음 제안해서 최근 Mikolov[5] 의 방법(Word2vec)이 주목받음 - 단어 별 similarity 계산이 가능하며, 비슷한 문맥에서 사용되는 단어들이 similarity 가 높음

2 Word Representation

아래 그림은 주어진 C 개의 window 사이즈 만큼의 단어가 있을 때, 그 다음에 등장할 단어의 확률인 $p(w_O|w_I)$ 를 Maximize 하는 $W_{V\times N}$과 $W'_{N\times V}$ 을 구하는 모델이다.
Stochastic Gradient Descent 방법 등을 이용해 학습된 $W_{V\times N}$, $W'_{N\times V}$ 의 평균값을 해당 단어의 representation 이라 한다.



출처 : Reference [6]

- 초기화
  - Vocabulary size V, Hidden layer size N (=Word Vector Size)
  - Input node 값들은 각 단어들의 One-hot encoding 값으로 x_1=[1,0,...0], x_2=[0,1,...0], x_V=[0,0,...1]
  - $W_{V\times N}$과 $W'_{N\times V}$ 는 random 하게 초기화

x_k에 해당하는 단어가 나왔다면, $x_k^T \cdot W = W_{k.} = v_{wI}$
지정된 window 를 돌면서 각 단어마다 나오는 v_{wI} 값을 평균하여 h 를 구한다.

최종적으로 Input 단어가 주어졌을 때 다음 단어를 예측하게 될 확률은

$$p(w_O|w_I) = \frac{\exp(v'^T_{w_O} \cdot v_{wI})}{\sum_{j=1}^{V} \exp(v'^T_{w_j} \cdot v_{wI})}$$

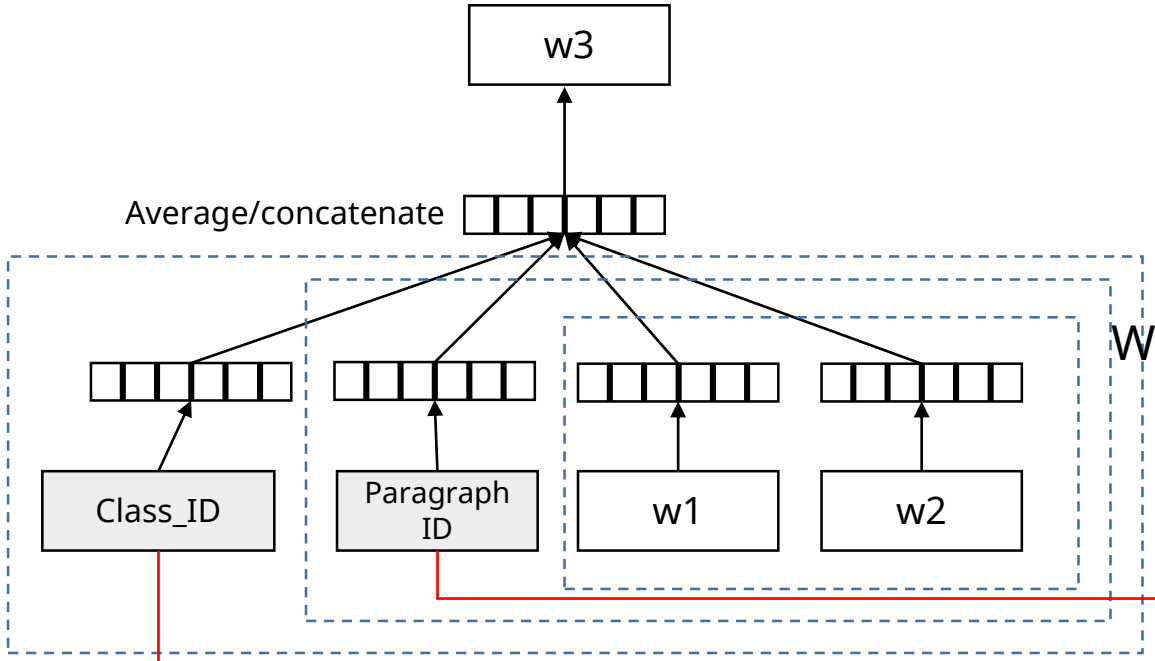
- Training

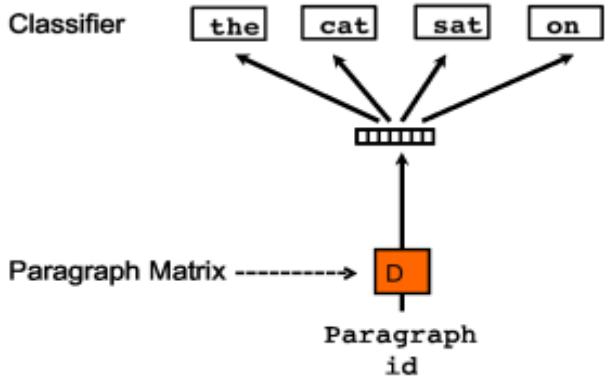
$\max p(w_O|w_I)$
Stochastic Gradient Descent (+Negative Sampling)

2 Word Representation

기존의 Word2Vec 방법을 응용하여 각각의 문장 혹은 문서를 단어와 함께 학습을 시키는 방법이 등장하였다.[3][8]

Doc2vec 방법에는 Word와 Paragraph 정보를 통해 다음 단어를 예측하는 Distributed Memory Model과 Paragraph 만으로 단어들을 예측하는 Distributed Bag of Words Model 이 있으며, Class2vec 에서는 긍정/부정과 같은 class 부여





Distributed Bag of Words

Doc2Vec : Distributed Memory Model

Class2Vec : 이 문서가 긍정적인 Review 이면 Positive class / 그렇지 않으면 Negative class

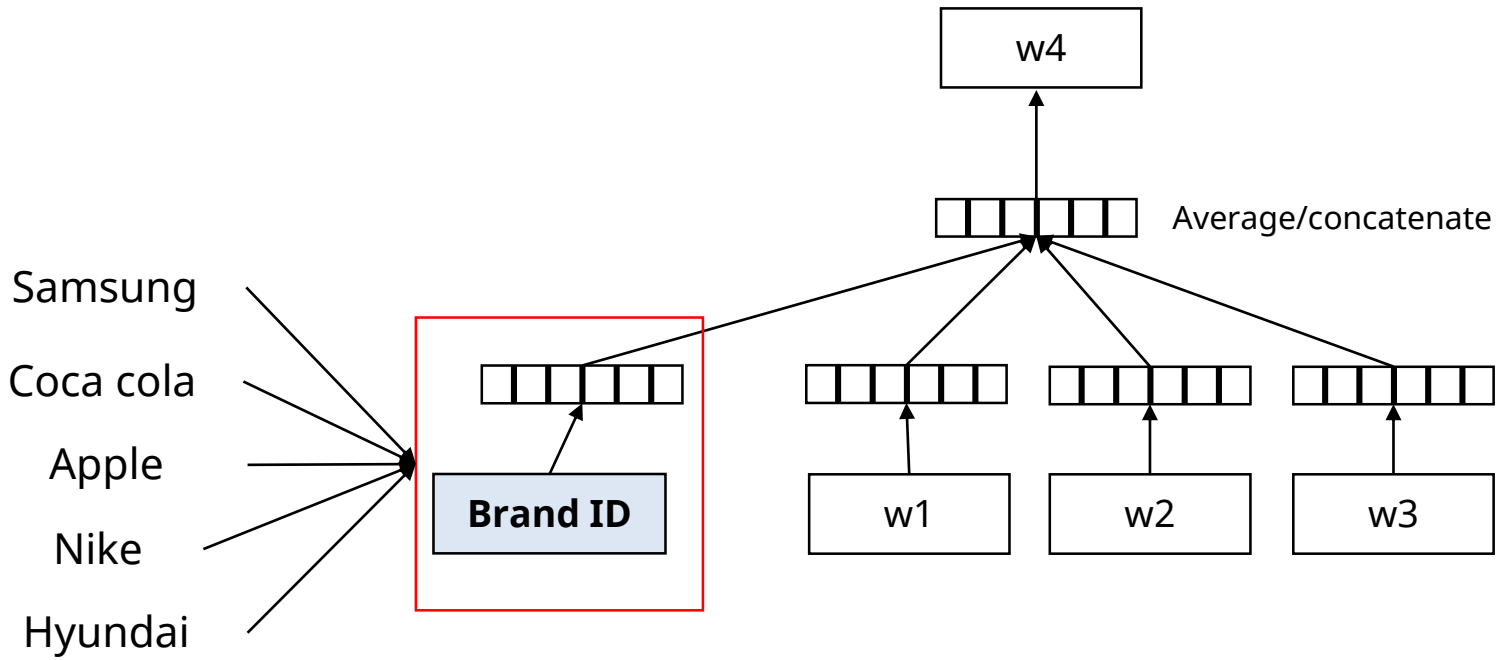
3 Proposed Method – Brand2vec

Distributed Memory Model[3] & Class2vec[8] 의 아이디어를 접목

특정 브랜드에 대해 Social media 혹은 product review data에서 말하는 모든 정보들을 모아

특정 브랜드를 '하나'의 vector로 표현 하면?

예) Samsung 제품에 대한 Review 인 경우 Brand_ID = 'Class_brands_Samsung' 과 같이 고유한 ID를 부여하여
모든 Samsung 제품의 Review를 반영한 하나의 Vector를 생성할 수 있음



3 Proposed Method - Modeling

Amazon Review Data[4] 중 Electronics 카테고리의 2012년도 이후 약 3백만개의 review, 3억개에 달하는 token 들에 대해 분석을 실시함

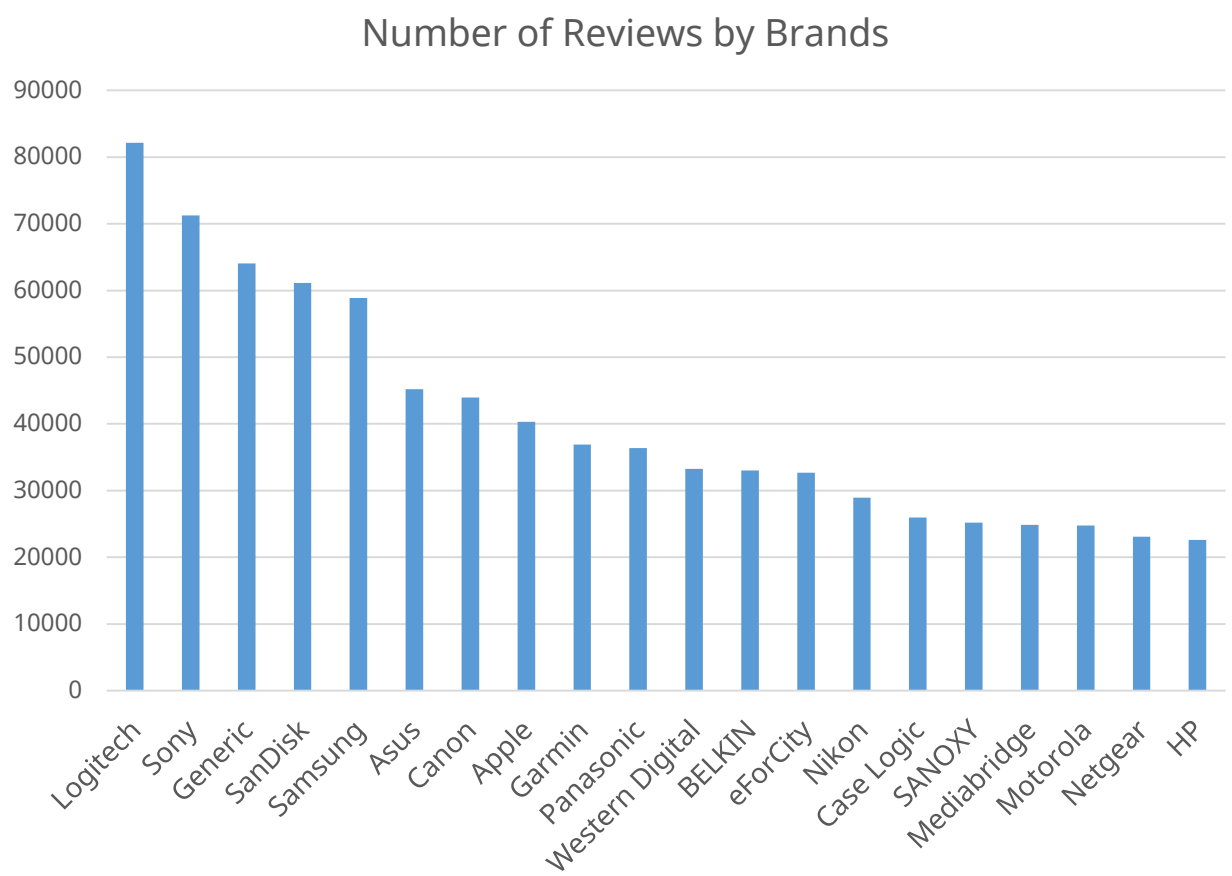
총 9,557개의 브랜드 중 Logitech, Sony, Generic 브랜드 순으로 review가 많이 달려 있는 것을 알 수 있다.

- 데이터 셋에 대한 설명

구분	
2012년 이후 Electronics 카테고리의 review	5,566,912 reviews
Review가 없거나, brand 정보가 없어서 제외	2,971,378 reviews
Total tokens	297,756,650 token
Number of unique words	2,154,172 words

Year	Num of Reviews
2012	611,669
2013	1,405,001
2014	954,708

총 Brands :	9,557

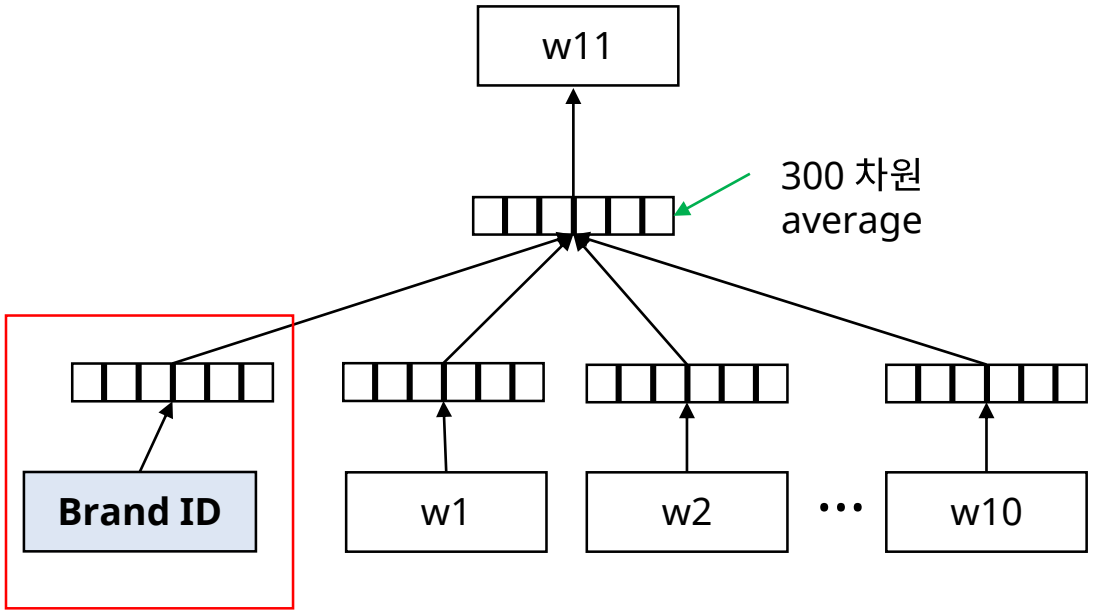


3 Proposed Method - Modeling

Word2vec을 장점 중 하나는 전처리를 최소화하여도 유의미한 결과를 얻을 수 있다는 점이다.
특수문자 제거, 숫자 제거, 소문자로 변환의 최소한의 전처리를 하였다.
Gensim[1] 패키지를 활용하였으며, 여러 논문에서 일반적으로 사용하는 Parameter 를 이용하여 학습하였다.

- Preprocessing
  1) #,?,@ 등 특수문자 제거
  2) 숫자 제거
  3) 소문자

- Parameter setting
  1) Distributed Memory model
  2) Average / concatenation 방법 중 average
  3) Brand vector, Word vector dimension : 300
  4) 300번 이하로 등장한 단어는 제거
  5) Window : 10
  6) Epoch : 10



*training time : 약 30분 (Intel i7-4790, 8 core)

1) https://github.com/piskvorky/gensim/

4 Result & Application

Modeling 결과를 확인하기 위해 대표적으로 Samsung, Apple 의 Brands vector에 대해서
다른 Brand Vector, 그리고 단어 Vector 들을 각각 cosine similarity를 기준으로 정렬하면 아래와 같다.
Brand Vector 끼리 비교했을 때는 비슷한 브랜드가 등장하고, 단어와 비교했을 때는 각 Brand의 제품들이 등장함.

Samsung Brand Vector vs 다른 Brand Vector

Rank	Brand Vector	Similarity
1	brand_Marquis	0.4510
2	brand_Asus	0.4023
3	brand_Acer	0.3935
4	brand_PIPO Technology	0.3875
5	brand_Proscan	0.3864
6	brand_Le Pan	0.3842
7	brand_Toshiba	0.3795
8	brand_Sharp	0.3762
9	brand_Lenovo	0.3636
10	brand_GT-B9150ZKYXAR	0.3552
11	brand_Apple	0.3547
12	brand_LG	0.3545
13	brand_Hannspree	0.3538
14	brand_Kocaso	0.3511
15	brand_Archos	0.3507
16	brand_HP	0.3473
17	brand_SKYTEX Technology Inc.	0.3463
18	brand_Digital2	0.3360
19	brand_DPI	0.3347
20	brand_Properss	0.3291

Samsung Brand Vector vs Word Vectors

Rank	Word Vectors	Similarity
1	samsung	0.5248
2	galaxy	0.4550
3	samsung's	0.4417
4	tab	0.4318
5	note	0.3549
6	ppi	0.3484
7	samsungs	0.3435
8	tablet	0.3415
9	ativ	0.3213
10	jellybean	0.3195
11	siii	0.3161
12	multitasking	0.3118
13	android	0.3011
14	beautiful	0.2986
15	sii	0.2894
16	apps	0.2854
17	smarttv	0.2849
18	smart	0.2834
19	inch	0.2809
20	tablets	0.2795

Apple Brand Vector vs 다른 Brand Vector

Rank	Brand Vector	Similarity
1	brand_Piel Frama	0.4034
2	brand_Odoyo	0.3659
3	brand_PhotoFast	0.3629
4	brand_Samsung	0.3547
5	brand_Sling Grip	0.3482
6	brand_Jisoncase	0.3457
7	brand_Love My iPad	0.3413
8	brand_JETech	0.3344
9	brand_iFlash	0.3306
10	brand_UNIQUE FINDZ	0.3286
11	brand_Twelve South	0.3264
12	brand_Otto Case, LLC	0.3217
13	brand_Yoobao	0.3155
14	brand_ZeroChroma	0.3124
15	brand_Poetic	0.3096
16	brand_Siskiyou	0.3095
17	brand_Gearonic	0.3071
18	brand_Nccypo	0.3067
19	brand_Gooband accessory for Apple	0.3023
20	brand_Speck	0.3001

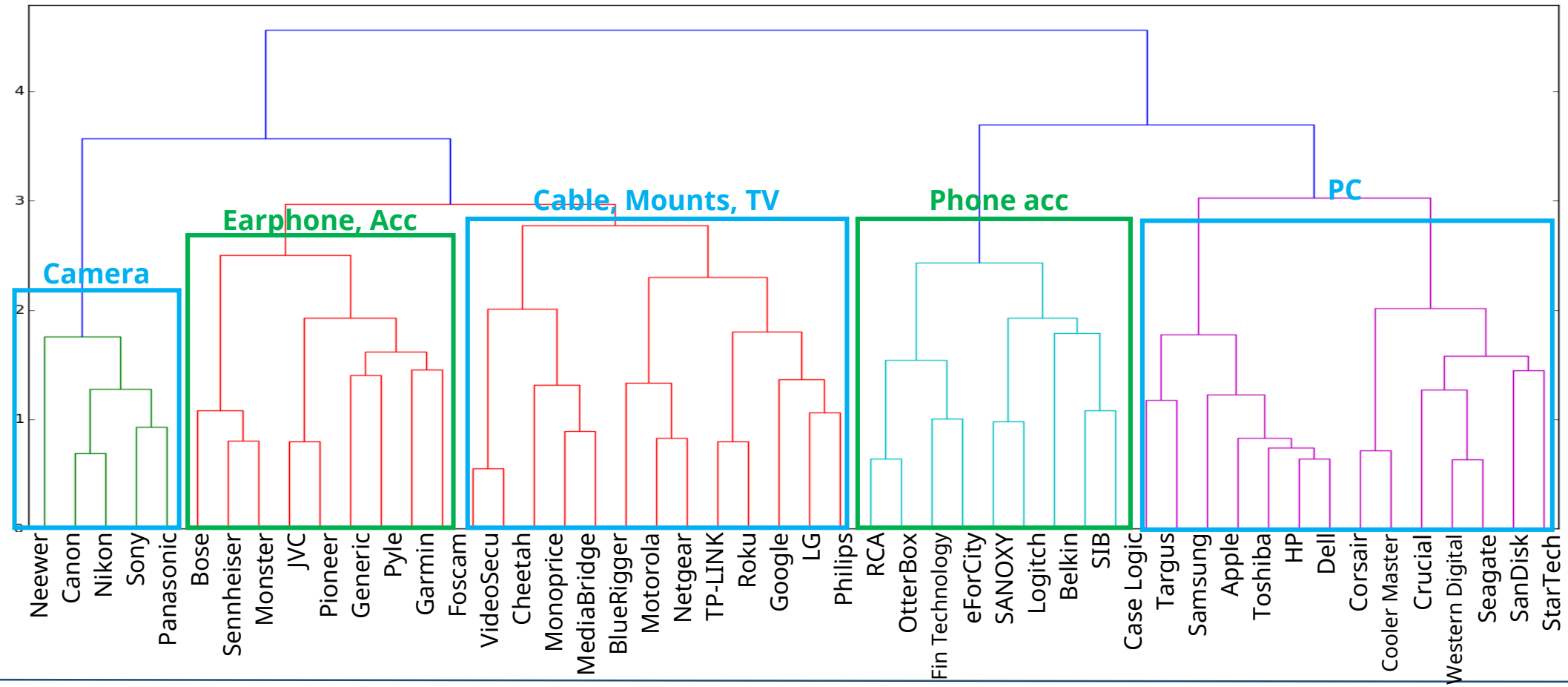
Apple Brand Vector vs Word Vectors

Rank	Word Vectors	Similarity
1	apple	0.5391
2	apple's	0.5017
3	ipad	0.4745
4	retina	0.4595
5	mini	0.3595
6	capsule	0.3560
7	ipad's	0.3402
8	imac	0.3275
9	ios	0.3144
10	icloud	0.3097
11	air	0.2993
12	facetime	0.2968
13	imacs	0.2929
14	generation	0.2921
15	ipads	0.2919
16	newest	0.2891
17	apples	0.2858
18	multitasking	0.2848
19	gen	0.2764
20	macbook	0.2735

4 Result & Application

Review 수가 많은 상위 50개 브랜드를 선정하여 Agglomerative hierarchical clustering 실시함.

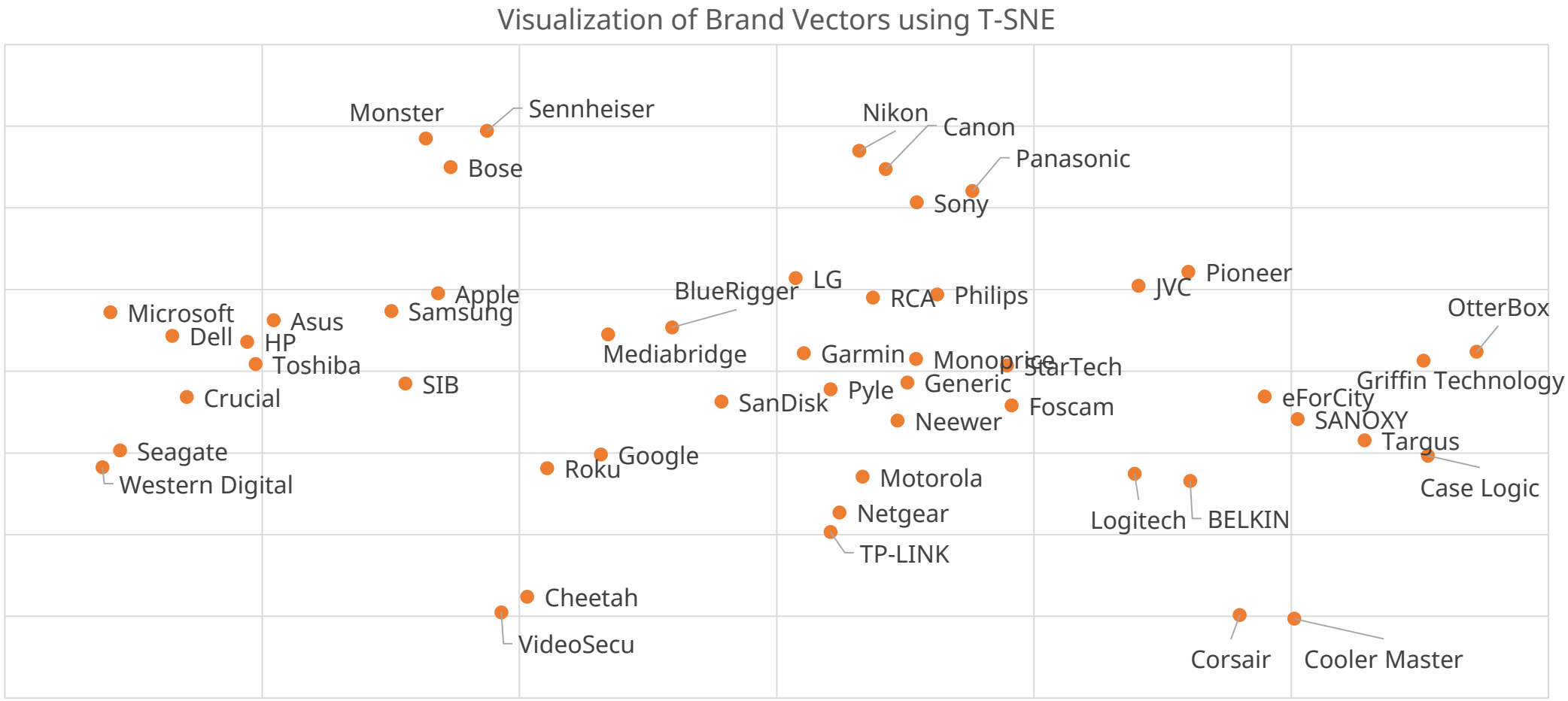
각 브랜드 vector의 cosine similarity를 계산해 pair-distance matrix를 만들고, Ward ' s minimum variance method 사용하였다. Canon, Nikon과 같이 비슷한 제품군의 브랜드끼리 가깝게 위치한 것을 확인할 수 있다.



4 Result & Application – Perceptual Map

아래는 50개 Brands 의 Brands Vector를 T-SNE를 이용하여 시각화 한 것이다

Dendrogram 결과와 같이 비슷한 브랜드들이 묶이는 것을 통해 Brand2Vec 가 Brand 의 특성을 반영했다고 볼 수 있다.



4 Result & Application – Product Properties

Brand2Vec 방법을 통해 representation이 잘 되었다면 각각의 Vector는 각 Brand가 생산하는 제품들의 Property를 포함하고 있을 것이다.

아래와 같이 Computer, Earphone, Camera 라는 단어 Vector에 대해 각각 가장 가까운 Brand Vector를 찾아보았다.

(computer + desktop)/2

Rank	Brand	Similarity
1	Dell	0.470087
2	HP	0.450589
3	Toshiba	0.354632
4	Asus	0.286443
5	Microsoft	0.281808
...		
46	Pioneer	-0.102164
47	eForCity	-0.105718
48	Nikon	-0.108203
49	Neewer	-0.121619
50	Sony	-0.132765

(earphone+ headphone)/2

Rank	Brand	Similarity
1	Sennheiser	0.346347
2	Monster	0.275864
3	Monoprice	0.184732
4	Bose	0.149591
5	JVC	0.129364
...		
46	Neewer	-0.079763
47	Garmin	-0.087942
48	LG	-0.09826
49	Generic	-0.101777
50	Dell	-0.117044

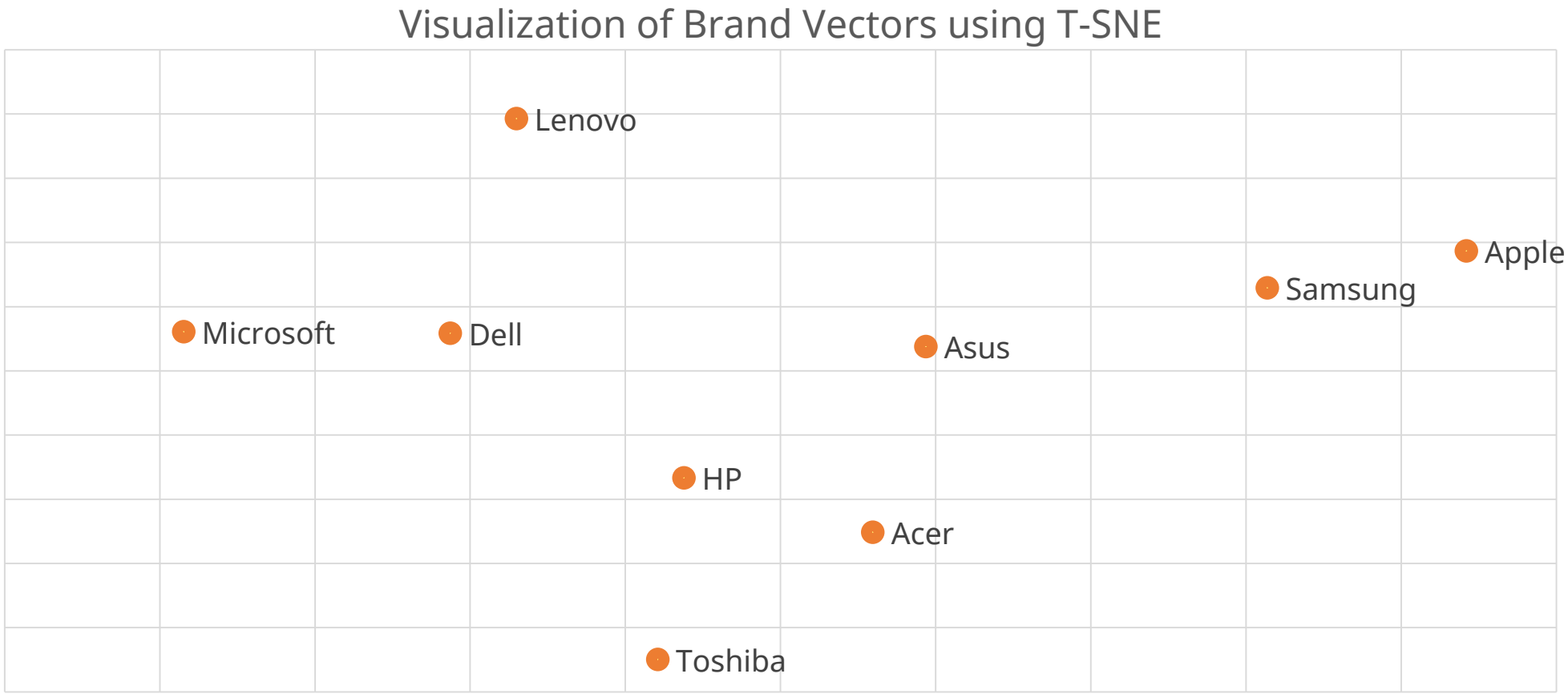
(camera + cameras)/2

Rank	Brand	Similarity
1	Canon	0.474762
2	Nikon	0.413864
3	Foscam	0.320791
4	Panasonic	0.308198
5	Sony	0.239091
...		
46	Asus	-0.140747
47	RCA	-0.150497
48	Microsoft	-0.152087
49	Philips	-0.172114
50	Monoprice	-0.173792

4 Result & Application – Perceptual map

Brand Vector를 통해서 비슷한 제품군을 갖고 있는 Brand 간의 Perceptual Map(Positioning map)을 그릴 수도 있다.

그러나, T-SNE를 통한 시각화나, Brand Vector 가지고는 축의 의미나, 상대적 거리가 '왜' 발생했는지에 대한 설명력이 부족하다.



4 Result & Application – Conditional probability

그럼 저렇게 브랜드 vector 가 다르게 되는 '이유' 는 무엇일까?

같은 단어라도 각 '브랜드' 에서 쓰이는 문맥이 다르기 때문에 특정 단어의 브랜드에 대한 설명력을 $P(Brands|word)$를 통해 알아 볼 수 있다

일반적인 Review 에서 많이 사용되는 '형용사' adj_j 추출[1]

예) $adj \in \{good, nice, new \ \ldots\}$

비교하고 싶은 Brand Vectors B_i,

예) $i \in \{Apple, Samsung \ \ldots\}$

모든 Brand와 형용사들 간에 Conditional Probability 를 구한다

$$P(B_i \mid w_j) = \frac{\exp(B_i \cdot w_j)}{\sum_{brands} \exp(B_{brand} \cdot w_j)}$$

주어진 형용사에 대해 개별 Brand 와 Brand 비교

예) "special" 이라는 단어는 Apple brand Vector와 가까울 것인가, Microsoft brand Vector와 가까울 것인가

그래서, 각 브랜드의 특징을 나타내는 형용사를 추출할 수 있을까?

[1] 3백만개의 Review 중 3만개만 Random Sampling 하였으며, NLTK를 이용해 Part of Speech tagging을 실시

4 Result & Application – Conditional probability

예를 들어, 위에서 실시한 Desktop 제조업체 9개를 이용하여 $P(B_i \mid w_j)$ 를 구하면 각 브랜드 별로 특징을 나타내는 단어들을 뽑을 수 있다. 그 중 대표적으로 Samsung, Apple, Microsoft 의 예시가 아래와 같다.

그 단어들의 문맥상 쓰임새를 파악하기 위해 해당 브랜드의 Review에 대해서 PMI [1] 가 높은 단어들 혹은 원문을 확인하였다

word	Samsung
active	1
vibrant	0.9994
dynamic	0.99793
picky	0.99677
fat	0.99618
higher	0.99408
pic	0.99394
song	0.99333
popular	0.99311
blurry	0.99226
pull	0.99085
write	0.98579
different	0.98198
special	0.9786
upper	0.97353
dish	0.96545
music	0.96043
better	0.94718
worse	0.94411
ridiculous	0.9336
professional	0.8956

glasses, 3d

Trackpad, mouse ..

Network, TV

Monitor, cameras

word	Apple
air	1
previous	0.9999636
adhesive	0.9993592
magic	0.9992156
white	0.9987635
greatest	0.9968496
latest	0.995961
larger	0.9953194
sorry	0.9914955
dry	0.9900548
heavier	0.9893667
protective	0.987897
recent	0.9759585
monthly	0.9743298
tired	0.9676014
stronger	0.9642975
substantial	0.9606566
clearer	0.953744
extra	0.9512706
safe	0.9501784
thermal	0.9471483

Glass, screen

Standpoint, Keyboard

Elite, positioning

Carrying ,waiting

Hot keys

word	Microsoft
ergonomic	1
flat	0.9999853
harder	0.9999626
natural	0.9999315
neat	0.9998434
visual	0.9998108
solar	0.9997959
traditional	0.9996345
sweet	0.9995185
comfortable	0.9968041
uncomfortable	0.9967529
yellow	0.9964272
magnetic	0.9957358
optional	0.9948526
included	0.9942972
full	0.9924303
unreliable	0.992372
usable	0.9891648
virtual	0.9870451
hot	0.9856131
convenient	0.9855615

[1] window size = 5, 10번 미만으로 나온 word는 제거함

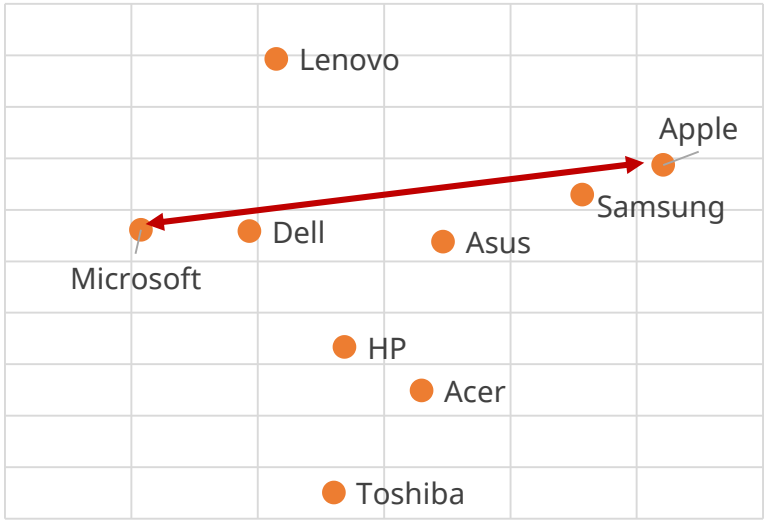
4 Result & Application – Conditional probability

앞의 결과를 이용하여, Brand 와 Brand 간의 차이를 나타내는 단어를 추출할 수도 있다.

이와 같이 다양한 방법으로 Business 의사결정에 활용할 수 있을 것으로 기대된다.

Apple이 더 높은것			
Word	Apple	Microsoft	Apple-Microsoft
previous	0.99996	0.00001	0.99995
adhesive	0.99936	0.00000	0.99936
magic	0.99922	0.00078	0.99844
white	0.99876	0.00058	0.99818
greatest	0.99685	0.00004	0.99681
latest	0.99596	0.00000	0.99596
larger	0.99532	0.00018	0.99514
sorry	0.99150	0.00208	0.98942
dry	0.99005	0.00106	0.98899
protective	0.98790	0.00000	0.98789
heavier	0.98937	0.00735	0.98201
recent	0.97596	0.00000	0.97596
monthly	0.97433	0.00309	0.97124
substantial	0.96066	0.00000	0.96066
tired	0.96760	0.01562	0.95198
extra	0.95127	0.00506	0.94621
newer	0.94451	0.00000	0.94450
expensive	0.94479	0.00101	0.94378
phenomenal	0.94284	0.00007	0.94277
third	0.94263	0.00009	0.94253

Microsoft가 더 높은것			
Word	Apple	Microsoft	Apple-Microsoft
ergonomic	0.00000	1.00000	-1.00000
flat	0.00000	0.99999	-0.99999
harder	0.00000	0.99996	-0.99996
natural	0.00000	0.99993	-0.99993
neat	0.00000	0.99984	-0.99984
visual	0.00000	0.99981	-0.99981
solar	0.00000	0.99980	-0.99980
traditional	0.00000	0.99963	-0.99963
sweet	0.00000	0.99952	-0.99952
comfortable	0.00003	0.99680	-0.99678
optional	0.00000	0.99485	-0.99485
included	0.00000	0.99430	-0.99430
uncomfortable	0.00250	0.99675	-0.99425
full	0.00000	0.99243	-0.99243
unreliable	0.00000	0.99237	-0.99237
magnetic	0.00426	0.99574	-0.99148
usable	0.00000	0.98916	-0.98916
virtual	0.00000	0.98705	-0.98704
hot	0.00024	0.98561	-0.98538
rechargeable	0.00000	0.98536	-0.98536



4 Result & Application – Frequency **기반 방법과 비교**

기존에 대부분의 마케팅 활용 측면의 Text mining 기법들은 Frequency 기반의 방법들이었다.
그러나, 이러한 방법은 빈도수가 적은 단어는 무시하거나, 무의미한 단어들이 많이 등장하게 됨.
반면에, Brand Vector 방법은 Reproducible 하고, 모든 review 들에 대해 문맥 정보를 객관적으로 반영함

	장점	단점
Frequency based	▪ 문서 수가 상대적으로 적은 text 에 대해서도 분석 가능 ▪ 직관적	▪ 무의미한 단어들이 많이 등장 ▪ 동의어 처리에 어려움 (eg. K2 등산복/소총) ▪ 많은 수작업이 필요하기 때문에 주관이 개입 될 가능성이 큼
Brand Vector approach	▪ Brand 라는 포괄적인 의미를 하나로 표현할 수 있음 ▪ Brand 단어가 포함되지 않은 review에 대한 정보도 포함	▪ 수작업을 최소화하여 Reproducible 하기 때문에 객관성을 확보 ▪ 직관적이지 않음 ▪ 동의어 처리가 가능 (BrancVector_K2 와 WordVector_K2)

5 Summary & Future work

Branc2Vec으로 모든 Review 정보를 반영한 하나의 Brand Vector 로 표현하였고, 이를 통해 브랜드 간의 상대적 거리가 유지되는지 확인하였고, conditional probability 를 활용하여 Brand 별 특징적인 keyword 를 추출할 수 있었다.

추가적으로 최적화된 parameter search 방법과, 시간에 따른 변화 추이를 분석할 수 있을 것이다.

- Summary
  - Branc2Vec으로 모든 Review 정보를 반영한 하나의 Brand Vector 로 나타냄
  - Brand Vector 의 다양한 활용방안을 제시함
    - Hierarchical Clustering, Perceptual Map 을 통해 유사 브랜드끼리 묶이는지 확인함
    - Brand의 Product Property가 유지되는지 확인함
    - Conditional Probability를 활용한 Brand 별 유의미한 단어 추출함
  - 이러한 방법을 통해 Brand2Vec 의 유용성을 확인하였으며, 향후 다양한 분야에 활용할 수 있을 것으로 기대함

- Future work
  - Brand2Vec 에 최적화 된 parameter search
  - Desktop 브랜드에 한정하지 않고, 더 다양한 분야에 대해 실험하면서 정성적 검정
  - Multidimensional scaling 과 같은 방법을 적용하여 더 발전된 Perceptual Map
  - 시간에 따른 review data를 활용하여 브랜드의 시간 변화 추이를 분석

6 Reference

reference

[1] Bengio, Yoshua, et al. "A neural probabilistic language model." *The Journal of Machine Learning Research* 3 (2003): 1137-1155.

[2] He, Wu, Shenghua Zha, and Ling Li. "Social media competitive analysis and text mining: A case study in the pizza industry." *International Journal of Information Management* 33.3 (2013): 464-472.

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

[4] McAuley, Julian, et al. "Image-based recommendations on styles and substitutes." *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015.

[5] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

[6] Mostafa, Mohamed M. "More than words: Social networks' text mining for consumer brand sentiments." *Expert Systems with Applications* 40.10 (2013): 4241-4251.

[7] Rong, Xin. "word2vec Parameter Learning Explained." *arXiv preprint arXiv:1411.2738* (2014).

[8] Sachan, Devendra Singh, and Shailesh Kumar. "Class Vectors: Embedding representation of Document Classes." *arXiv preprint arXiv:1508.00189* (2015).

[9] Tirunillai, Seshadri, 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 (2014): 463-479.