2015.11.25 산업공학특론

Brand2Vec – Parameter Search

서울대학교 산업공학과

양호성, 조성준

hoseong@dm.snu.ac.kr, zoon@snu.ac.kr

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

0 실험 목적

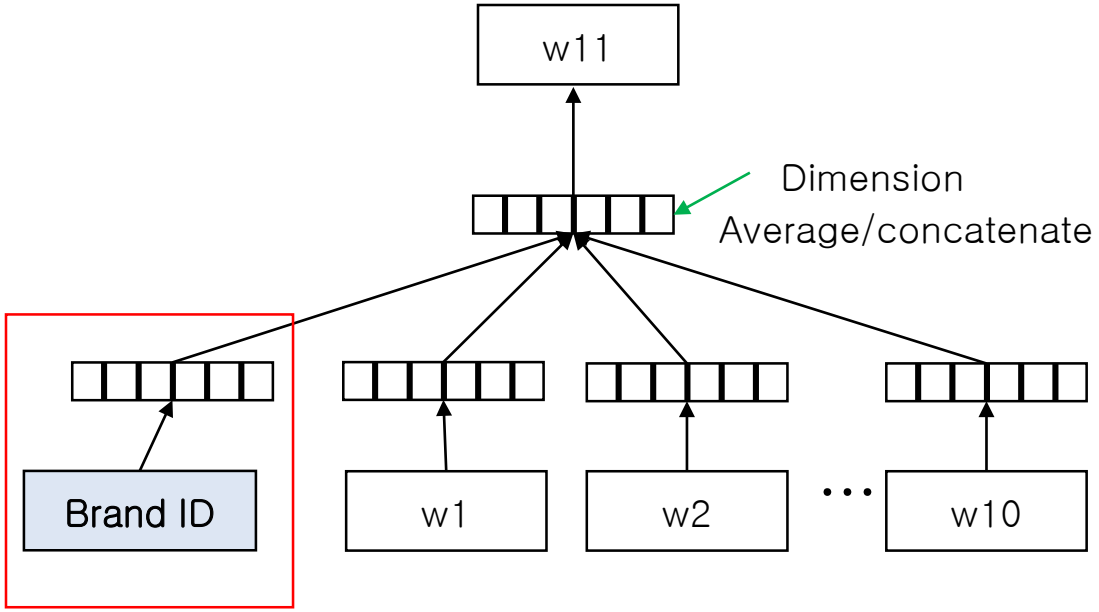
- Word2Vec 모델의 다양한 Parameter 값이 존재
  - Dimension
  - Windows
  - Training epoch
  - Methods

Review Dataset과 Branc2Vec 방법을
반영한 Parameter Search 를 해보자

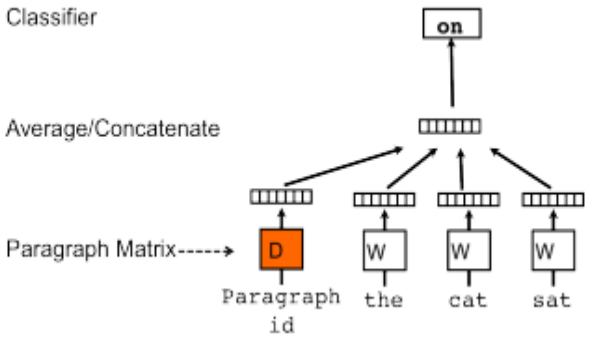
1 Word2vec / Doc2vec Parameters

Word2vec 방법론은 다양한 parameter가 존재함

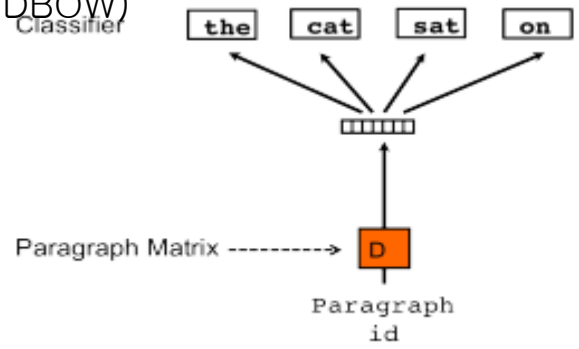
- Dimension
- Windows
- Training epoch
- Methods (PV-DM/PV-DBOW)
- Concatenate/average
- Negative Sampling, Subsampling of Frequent words 유무



distributed memory (PV-DM)



distributed bag of words (PV-DBOW)



1 기존 연구의 Parameter Setting

기존 연구에 따르면 corpus 의 특징에 따라 parameter 를 탐색

- Distributed Representations of Sentences and Documents (Quoc Le, Mikolov, 2014)

Parameters	Stanford Sentiment Treebank Dataset	IMDB dataset
windows	8	10
dimension	400	동일
Methods	Concatenate PV-DBOW & PV-DM	동일
Sentiment classifier	Logistic regression	Neural network (hidden 50)
Training epoch	x	X
sampling	x	X

  - 이 논문에 따르면, concatenate PV-DM + PV-DBOW 가 가장 성능이 좋음
  - Windows size 는 5~12까지가 보통 좋으며, 변동이 0.7% 차이로 적음

1 기존 연구의 Parameter Setting

기존 연구에 따르면 corpus 의 특징에 따라 parameter 를 탐색

- Class Vectors: Embedding representation of Document Classes (Denvendra, 2015)

Parameters	Amazon Electronic reviews & Yelp Reviews
windows	10
dimension	?
Methods	PV-DBOW
Sentiment classifier	Logistic regression
Training epoch	40
sampling	Negative, frequency sampling

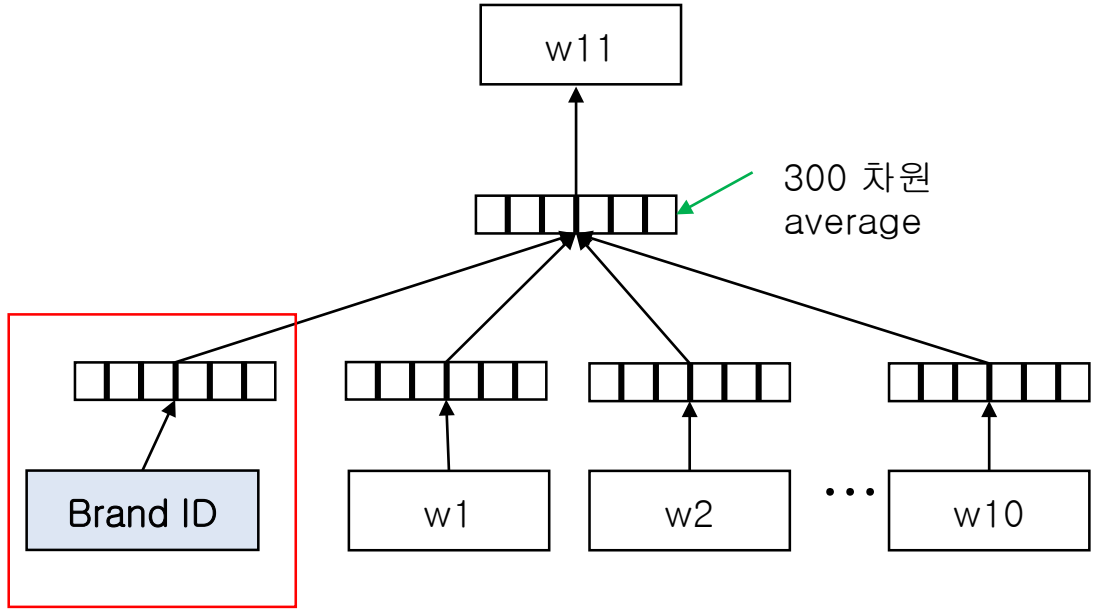
1 기존 연구의 실험 방법

일반적인 parameter 사용했으나

좀 더 객관적인 성능 척도를 이용해, embedding이 잘 되었는지 확인하고 싶음

- 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



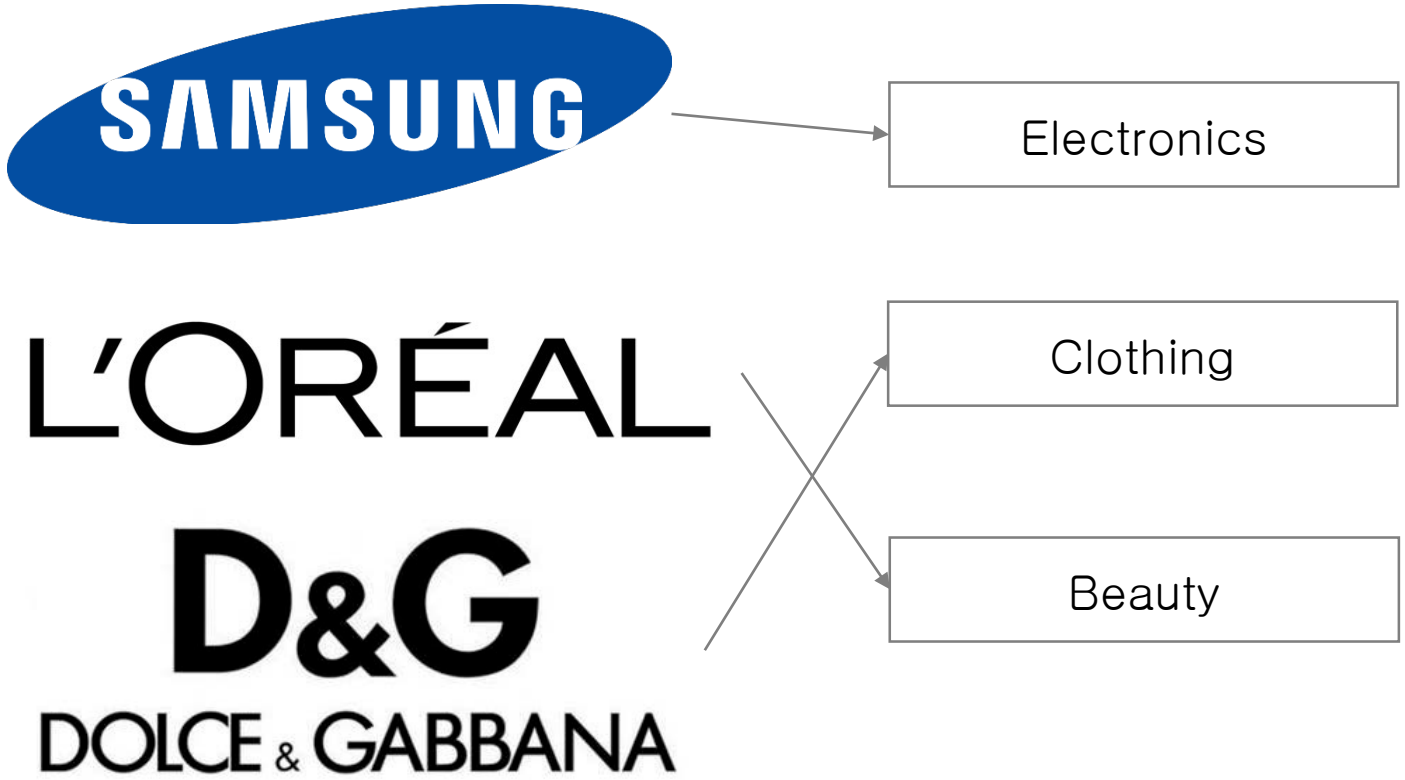
*training time : 약 30분 (Intel i7-4790, 8 core)

2 실험 설계

하지만, Brand Vector 가 잘 embedding 되었는지 확인하기 위해서는 위의 방법이 부적절함.

따라서 새로운 방법을 제안.

Brand Vector가 각 브랜드 정보를 잘 표현 한다면 각 브랜드가 주력으로 하는 제품의 category 를 구분할 수 있을 것.



Electronics, clothing, beauty 에서 각각 5만개씩 문서 추출

↓

총 15만개 이용
Brand2 Vector training

↓

Logistic Regression 으로
Category를 분류하는 Model 학습

↓

10-Fold Cross Validation

3 실험 결과

Top 500

Electronics, Clothing, Beauty 각 category 에서 review 수가 많은 500 개의 브랜드 선택

Electronics : 2백 5십만
Clothing : 6십만
Beauty : 8십만
로 각각 review 개수가 많이 다르므로, 각각 5만개씩 sampling

총, 15만개 dataset 사용

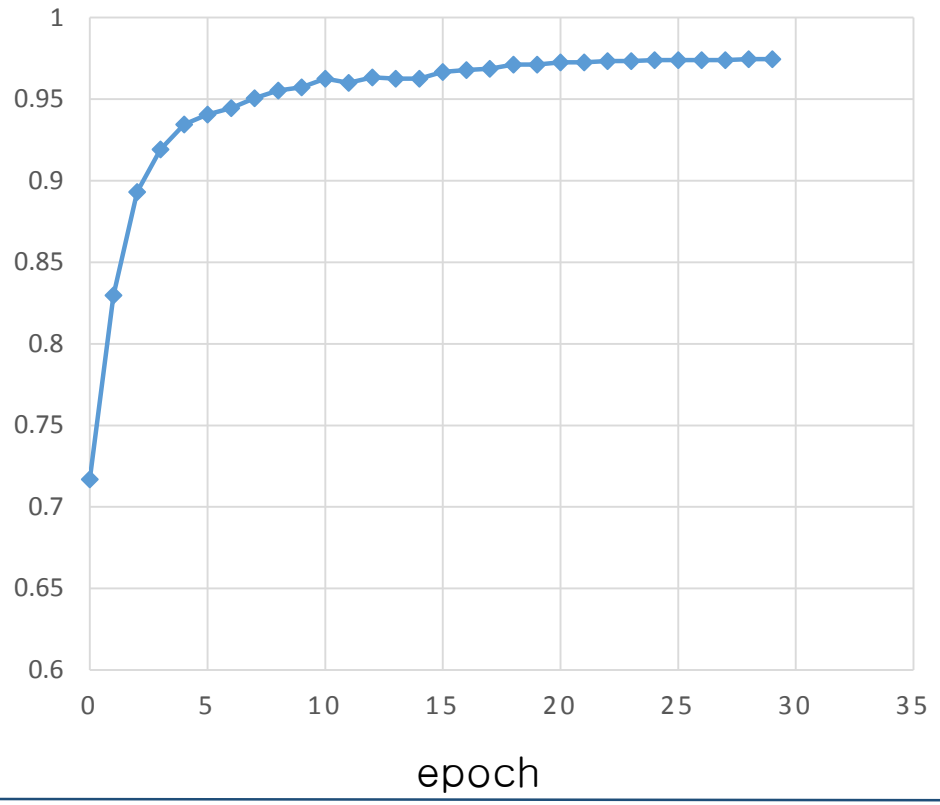
3 실험 결과

각 카테고리에서 Top 500개의 브랜드에 대해서만 review 추출 후, 5만개씩 sampling

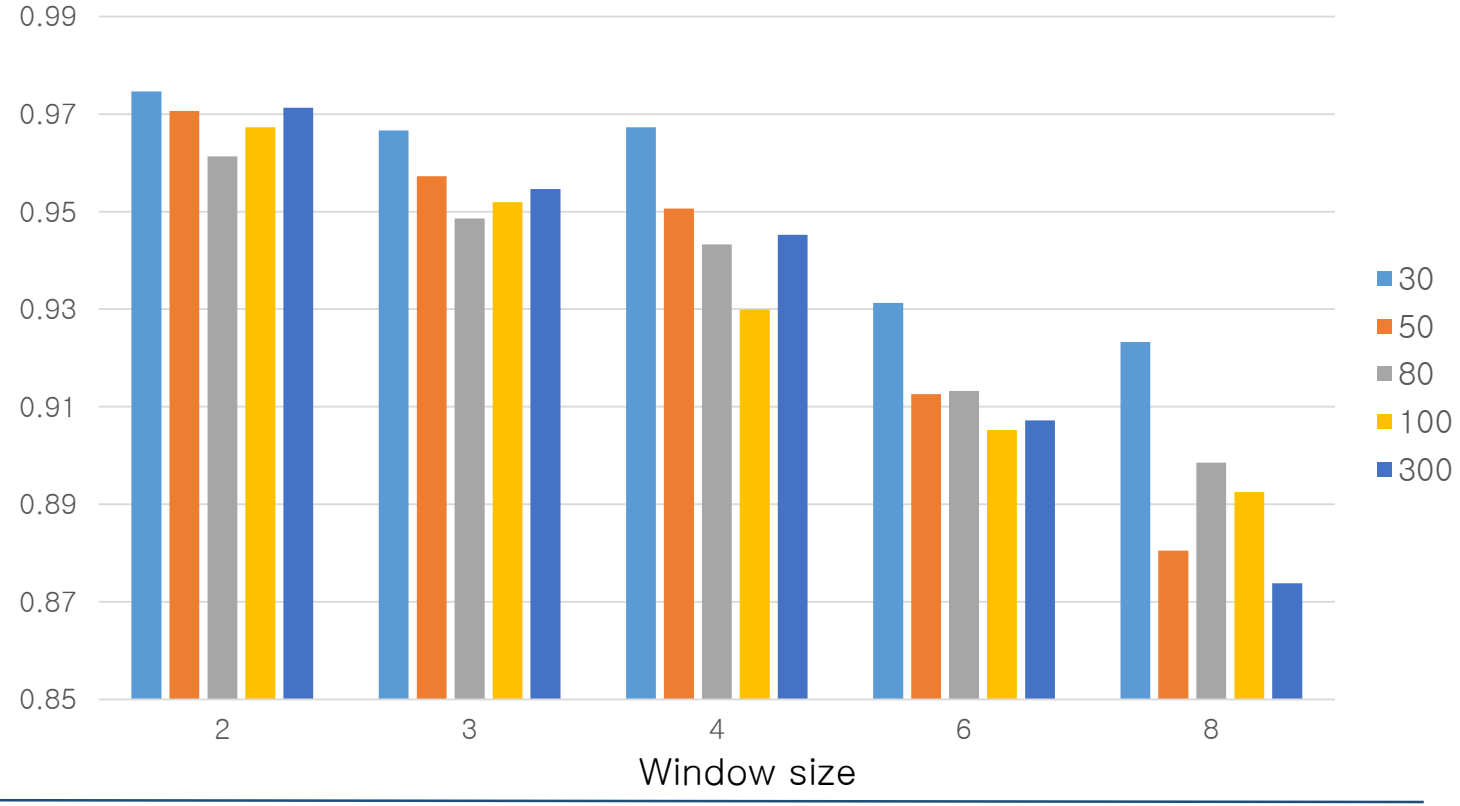
총 15만개 review

총 Brand 개수 : 1,458개

Dimension=30 일 때, epoch에 따른 성능



Window size와 dimension 에 따른 classifier 성능



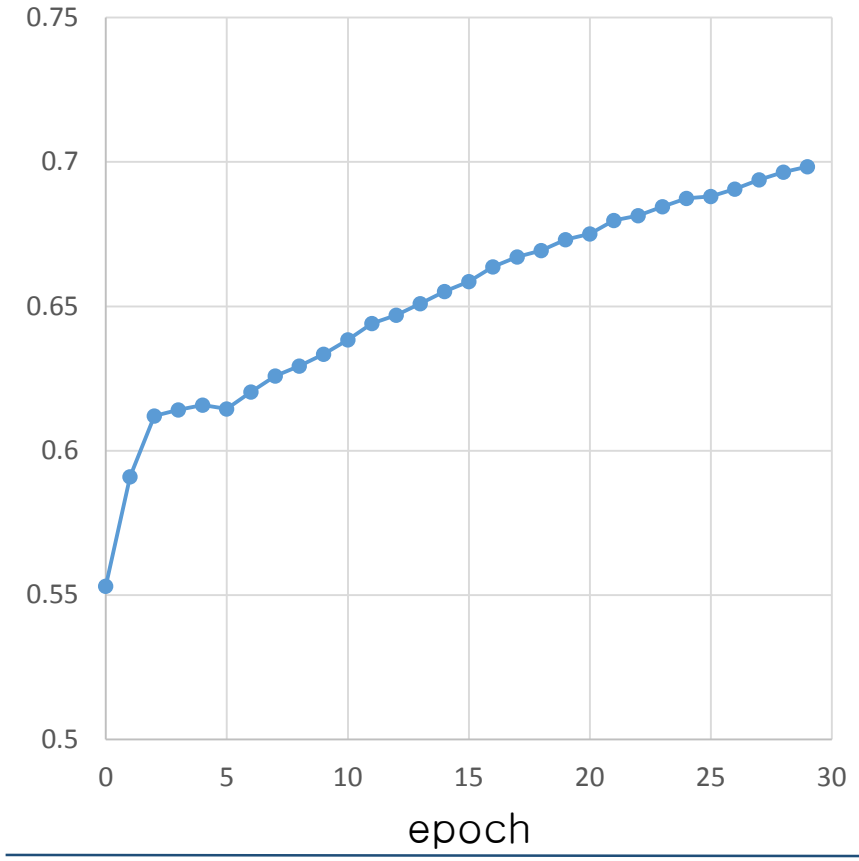
3 실험 결과

모든 브랜드에서

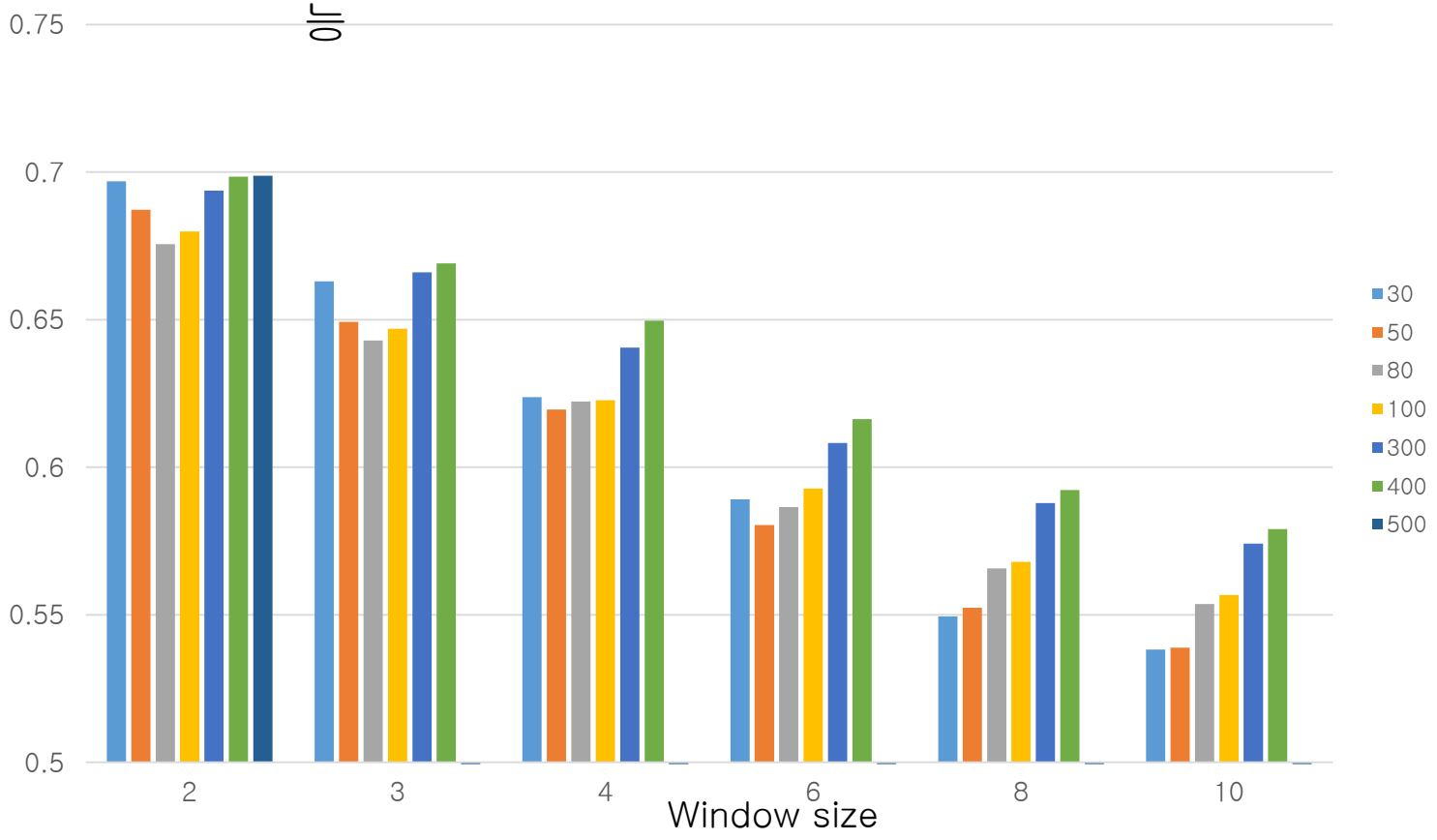
Electronics, Clothing, Beauty 각각 5만개씩 sampling

총 12,434개 브랜드 (각각, 3389, 3894, 5151)

Dimension=400 일 때, epoch에 따른 성능



Window size와 dimension 에 따른 classifier 성능



4 결론

문제점 및 추후 보완사항

- 현재까지 실험의 문제점

1) 문서수가 부족함. 각 브랜드별로 5만개씩 random sampling 해서 총 15만개 사용함.
그러나 dataset이 적어서 그런지 작은 window size, 작은 dimension에서 높은 accuracy 가 나타남

2) 여기서 확인하고 싶은게 Brand Vector 만 보는게 아니라 Brand Vector와 같이 학습되는 단어를 보고 싶은 것
즉, document classification 성능이 아니라, brand와 관련된 '단어'정보를 잘 유지할 필요가 있음

- 추후 보완사항

1) 추가적인 데이터 사용
2) Evaluation 대상을 'brand vector'로 보지 말고, "review dataset"에 대해서 확인해 보자.
document classification 문제를 푸는 것처럼 각각 review 데이터의 document vector 만 학습을 시켜서 성능 확인

-> amazon review dataset 이라는 주어진 데이터 셋의 특징을 반영한 parameter 학습이 될 수 있을 것으로 기대

Window size