한양대학교 인공지능연구실

StarSpace: Embed All The Things!

Facebook AI Research, 2017, AAAI

조건희

개요
StarSpace

- 개요
  - Word2Vec, GloVe, fast text 등과 같은 워드임베딩에 대한 연구
  - 여러 태스크에 적용할 수 있는 General-purpose neural embedding model 제안
    1) Text classification
    2) Content-based document recommendation
    3) Link prediction in knowledge bases
    4) Wikipedia article search & sentence matching
    5) Learning sentence embedding
  - StarSpace 라는 이름의 “Star”는 정규표현식에서 임의의 텍스트를 뜻하는 ‘*’ (와일드카드)를 의미
    - 태스크에 따라서 다양한 타입의 데이터를 임베딩하는 모델이라는 의미
    - E.g., TagSpace, PageSpace, DocSpace, GraphSpace, SentenceSpace

개요
StarSpace

- 핵심 아이디어
  - “음식 맛이 좋다” 라는 **문장**의 임베딩 벡터를 “음식” “맛이” “좋다” 각 **단어**의 임베딩 벡터의 합으로 표현하면, **문장**과 **단어**라는 서로 다른 타입의 정보를 하나의 벡터공간 상에 임베딩할 수 있음.
  - 여기서 각 **단어**를 문장의 feature (F)라고 부르고, feature 벡터의 집합을 dictionary 라고 부름
  - 임의의 문장의 임베딩 벡터는 그 문장을 이루는 단어 벡터(feature)의 합으로 표현될 수 있음.

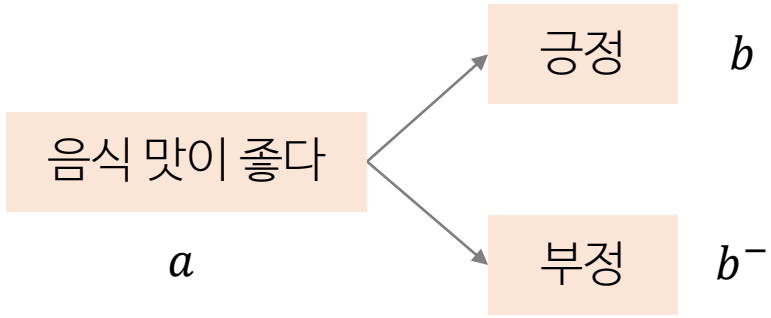
$$a = \sum_{i \in a} F_i \,,$$

음식 맛이 좋다		음식	맛이	좋다
a		F_0	F_1	F_2

학습
StarSpace

$$loss = \sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \dots, sim(a,b_k^-))$$

- 학습
  - (a, b) : Positive pair (down stream 태스크에 따라서 a, b 는 전부 달라짐)
  - b^- : k 개의 negative 샘플 (word2vec 기법)
    - text classification task 예시
      - a는 문장, b는 문장의 라벨.
      - 라벨 b 도 문장 a 와 동일한 벡터공간 상에 임베딩

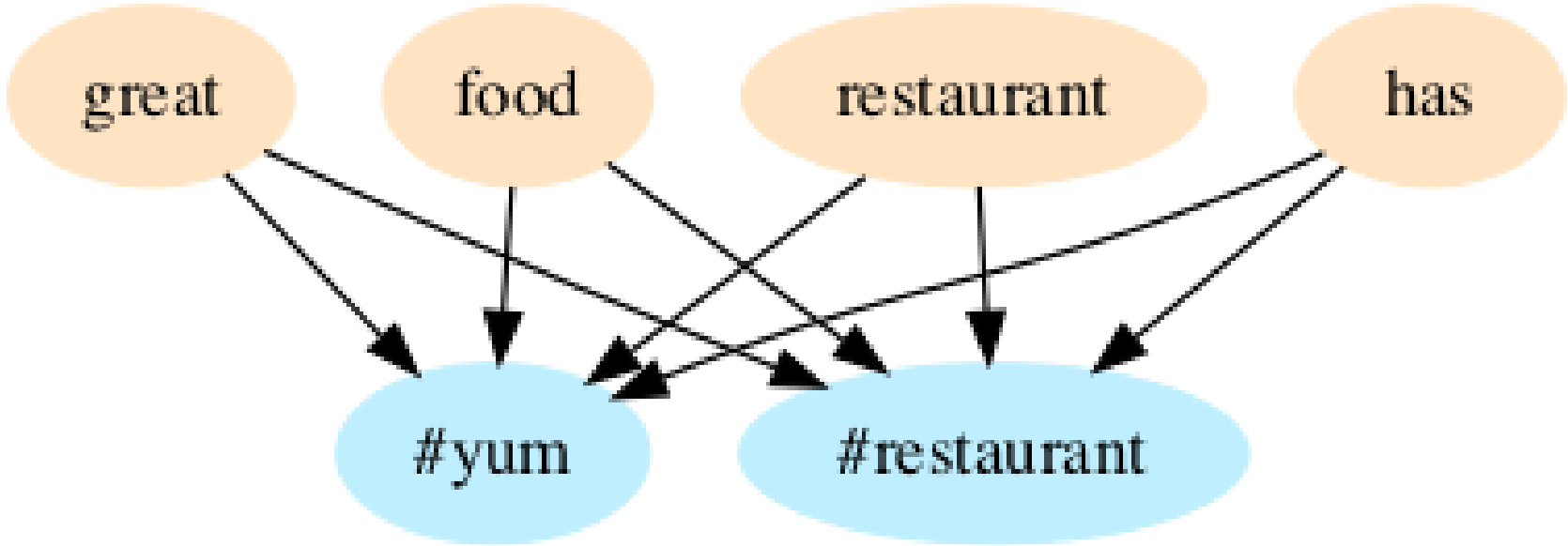


  - $sim(\cdot,\cdot)$: cosine similarity 또는 inner product
    - 라벨 개수가 적은 태스크에서는 성능 비슷
    - 라벨 개수가 많은 경우에는 코사인 유사도가 더 좋음
  - L_{batch} : positive pair와 negative pair 비교
    - 이 논문에서는 margin ranking loss 로 구현 $\max(0, \mu - sim(a,b) + sim(a,b^-))$

학습
StarSpace

$$loss = \sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \dots, sim(a,b_k^-))$$

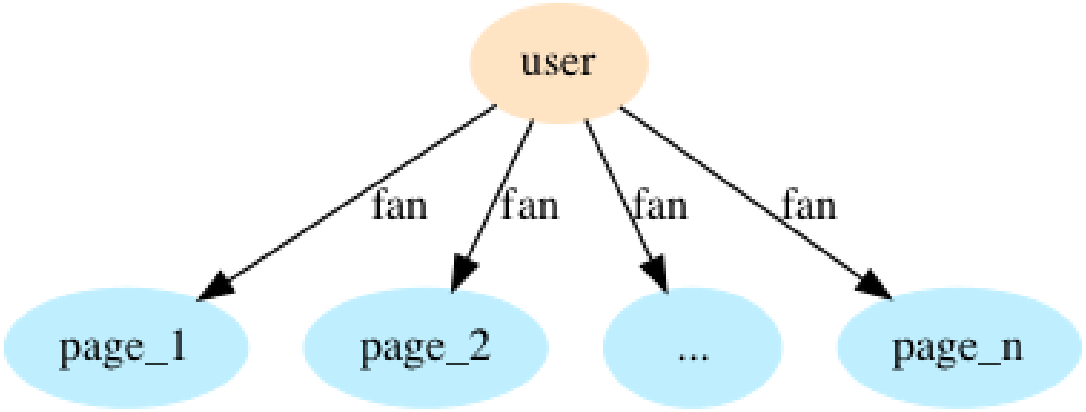
- 태스크에 따른 (a, b)
  - Text classification
    - a 는 문장이나 문서(bag-of-words), b는 라벨



학습
StarSpace

$$loss = \sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \dots, sim(a,b_k^-))$$

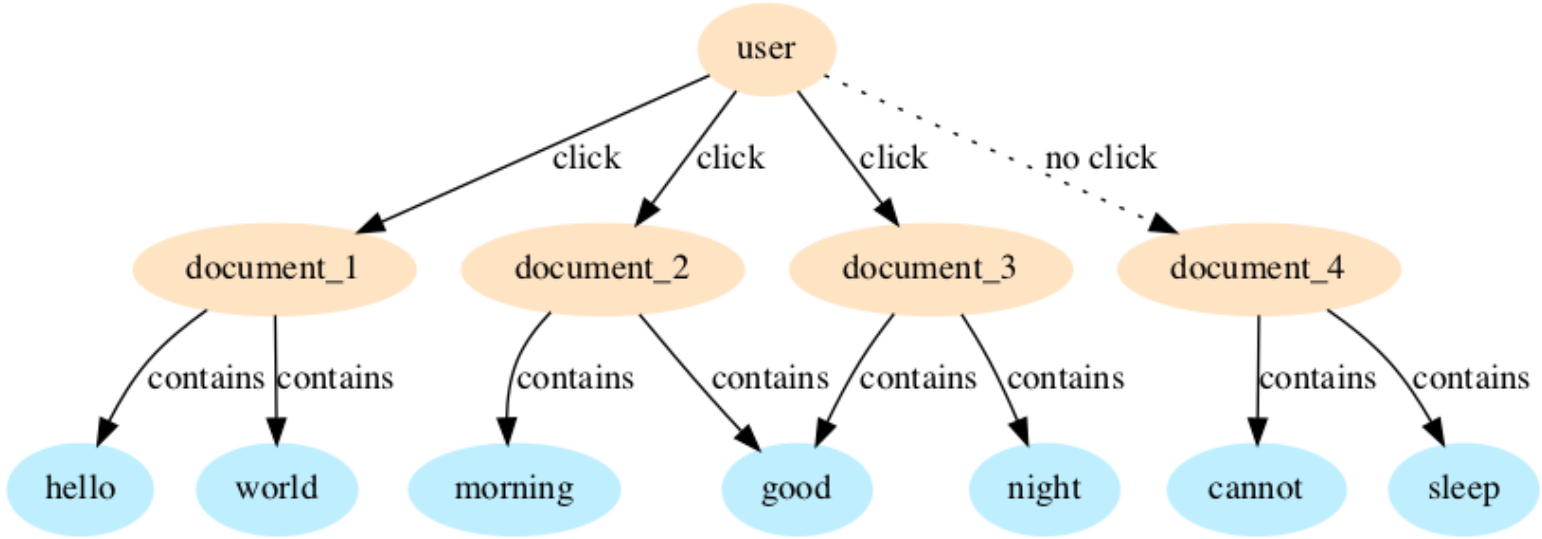
- 태스크에 따른 (a, b)
  - Collaborative filtering-based recommendation
    - a 는 user ID, b 는 user 의 선호 아이템
    - 새로운 user 에 대한 것은 새로 학습 해야함
  - Collaborative filtering-based recommendation with out-of-sample user extension
    - a 는 user가 선호하는 아이템 중 1개를 제외한 모든 아이템의 합, b 는 그 나머지 1개 아이템
    - user ID 직접 임베딩하지 않고 선호 아이템 벡터의 합으로 표현 (a bag-of-items)



학습
StarSpace

$$loss = \sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \dots, sim(a,b_k^-))$$

- 태스크에 따른 (a, b)
  - Content-based document recommendation
    - user 는 bag-of-items 로 표현, 아이템은 bag-of-features 로 표현
    - Document recommendation 예시
      - user 는 선호 document 벡터 합, document 는 포함 word 벡터 합



학습
StarSpace

$$loss = \sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \dots, sim(a,b_k^-))$$

- 태스크에 따른 (a, b)
  - Link prediction in knowledge bases
    - Graph 데이터 (h, r, t)
    - 예시 (h, r, t) = (*Obama, born-in, Hawaii*)
    - (a, b)는 다음 중 선택

      1) a 는 h, r 벡터 합으로 표현, b 는 t

      2) a 는 h 로 표현, b 는 r, t 벡터 합
    - 학습이 완료되면 (*Obama, born-in, ?*) 라는 문제 해결을 위해 *Obama 와 born-in* 두 벡터의 합과 가장 유사도가 높은 벡터를 찾으면 됨

학습
StarSpace

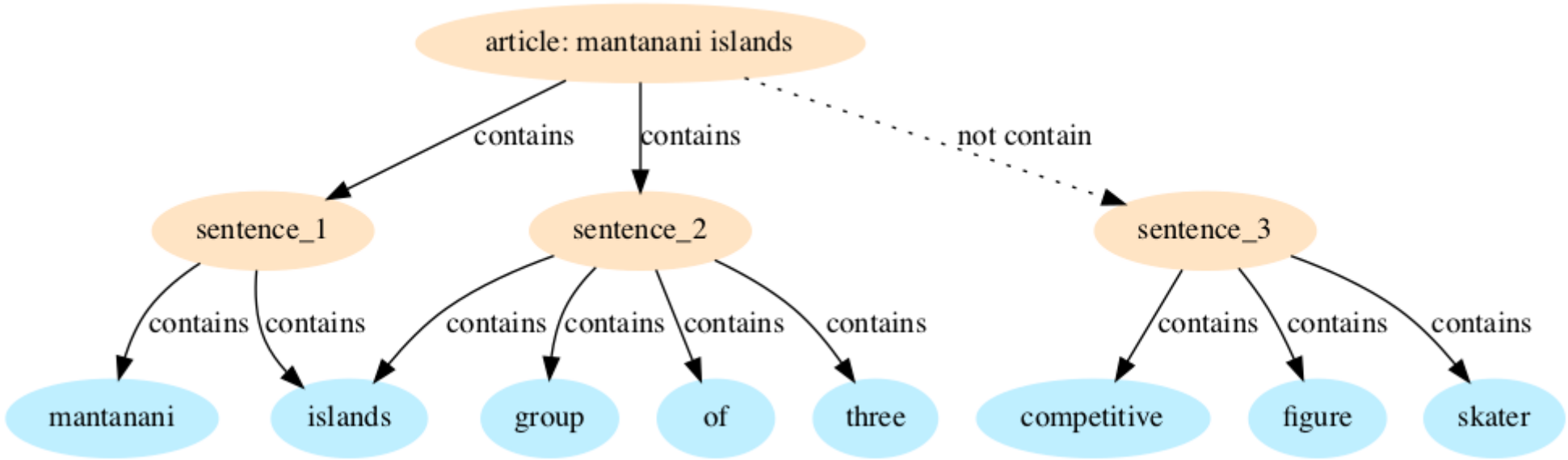
$$loss = \sum_{\substack{(a,b)\in E^{+} \\ b^{-}\in E^{-}}} L^{batch}(sim(a,b), sim(a,b_1^{-}), \dots, sim(a,b_k^{-}))$$

- 태스크에 따른 (a, b)
  - Information retrieval (Document search)
    - 라벨이 있는 데이터셋 : (키워드, 관련 문서)
      - a 는 키워드, b 는 관련 문서, b-는 관련 없는 문서
    - 라벨이 없는 데이터셋 : (라벨 없는 문서)
      - a 는 문서에서 등장한 임의의 단어, b 는 그 문서의 나머지 단어 벡터의 합

학습
StarSpace

$$loss = \sum_{\substack{(a,b)\in E^+ \\ b^-\in E^-}} L^{batch}(sim(a,b), sim(a,b_1^-), \dots, sim(a,b_k^-))$$

- 태스크에 따른 (a, b)
  - Learning sentence embedding
    - (a, b) : 같은 문서 내 문장
    - b− : 다른 문서 내 문장



학습
StarSpace

$$loss = \sum_{\substack{(a,b)\in E^{+} \\ b^{-}\in E^{-}}} L^{batch}(sim(a,b), sim(a,b_1^{-}), \ldots, sim(a,b_k^{-}))$$

- 태스크에 따른 (a, b)
  - ImageSpace: Learning Image and Label embeddings (in github)
    - CIFAR-10 데이터셋
    - Image feature는 pre-trained ResNeXt 모델(정확도 96.34%)의 last layer feature
    - 10개의 클래스 라벨 벡터를 이미지 feature 벡터와 같은 벡터공간 상에 임베딩
    - 이미지 라벨 prediction task에 활용할 수 있음
    - 5번 실험 평균 정확도 96.40%

성능 평가
StarSpace

Metric	Hits@1	Hits@10	Hits@20	Mean Rank	Training Time
Unsupervised methods					
TFIDF	0.97%	3.3%	4.3%	3921.9	-
word2vec	0.5%	1.2%	1.7%	4161.3	-
fastText (public Wikipedia model)	0.5%	1.7%	2.5%	4154.4	-
fastText (our dataset)	0.79%	2.5%	3.7%	3910.9	4h30m
Tagspace†	1.1%	2.7%	4.1%	3455.6	-
Supervised methods					
SVM Ranker: BoW features	0.99%	3.3%	4.6%	2440.1	-
SVM Ranker: fastText features (our dataset)	0.92%	3.3%	4.2%	3833.8	-
StarSpace	3.1%	12.6%	17.6%	1704.2	12h18m

Table 1: Test metrics and training time on the Content-based Document Recommendation task. † Tagspace training is supervised but for another task (hashtag prediction) not our task of interest here.

성능 평가
StarSpace

Model	AG news	DBpedia	Yelp15
BoW*	88.8	96.6	-
ngrams*	92.0	98.6	-
ngrams TFIDF*	92.4	98.7	-
char-CNN*	87.2	98.3	-
char-CRNN⋆	91.4	98.6	-
VDCNN◇	91.3	98.7	-
SVM+TF†	-	-	62.4
CNN†	-	-	61.5
Conv-GRNN†	-	-	66.0
LSTM-GRNN†	-	-	67.6
fastText (ngrams=1)‡	91.5	98.1	**62.2
StarSpace (ngrams=1)	91.6	98.3	62.4
fastText (ngrams=2)‡	92.5	98.6	-
StarSpace (ngrams=2)	92.7	98.6	-
fastText (ngrams=5)‡	-	-	66.6
StarSpace (ngrams=5)	-	-	65.3

Table 2: Text classification test accuracy. * indicates models from (Zhang and LeCun 2015); ⋆ from (Xiao and Cho 2016); ◇ from (Conneau et al. 2016); † from (Tang, Qin, and Liu 2015); ‡ from (Joulin et al. 2016); ** we ran ourselves.

Training time	ag news	dbpedia	Yelp15
fastText (ngrams=2)	2s	10s	
StarSpace (ngrams=2)	4s	34s	
fastText (ngrams=5)			2m01s
StarSpace (ngrams=5)			3m38s

Table 3: Training speed on the text classification tasks.

정리
StarSpace

- 여러 태스크에 적용 가능한 새로운 임베딩 학습 방법론 제안
- 페이스북 리서치
- 학습에 걸리는 시간도 성능평가에 포함
- 실험의 완고함

참고

- 논문 링크 : https://arxiv.org/abs/1709.03856
- Github : https://github.com/facebookresearch/StarSpace.git