



Learning Sentence Similarity via Generating Association Process

2018. 06. 14

고려대학교 산업경영공학부

Data Science & Business Analytics Lab

박경찬, 손규빈, 최희정

AGENDA

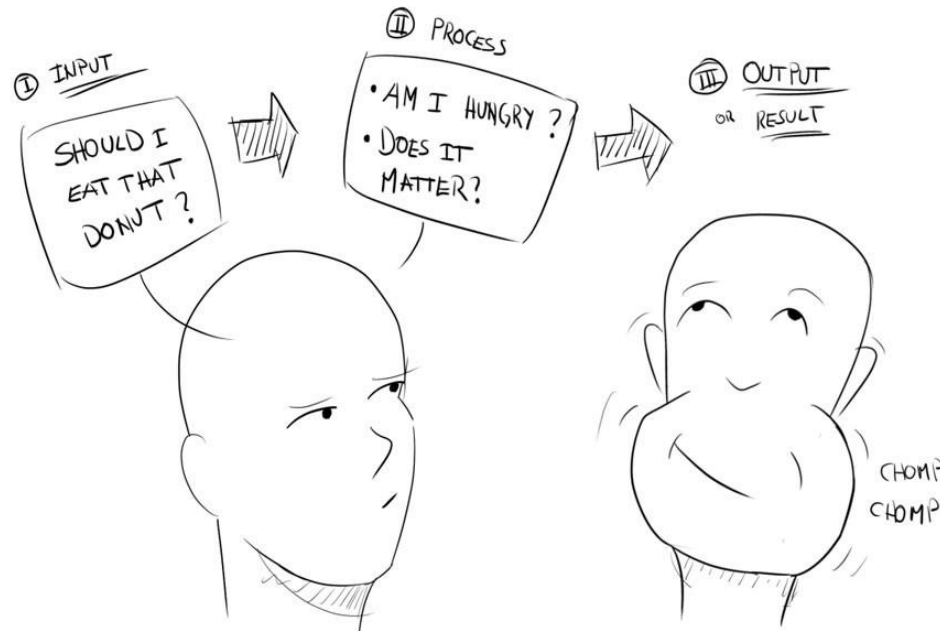
1. 연구 배경
2. 이전 모델
3. 데이터 설명
4. Model 설명
5. 실험 결과

1. 연구 배경

1. 연구 배경

❖ 사람의 문장 유사여부 판단 과정 중 연상작용을 적용한 모델 제안

- 사람은 연상작용을 통해 두 문장의 유사여부를 판단함
- 이러한 연상작용을 모델에 적용해 문장을 이미지에 빗대어 표현한 후, 두 문장의 유사여부를 판단하는 모델을 제안함

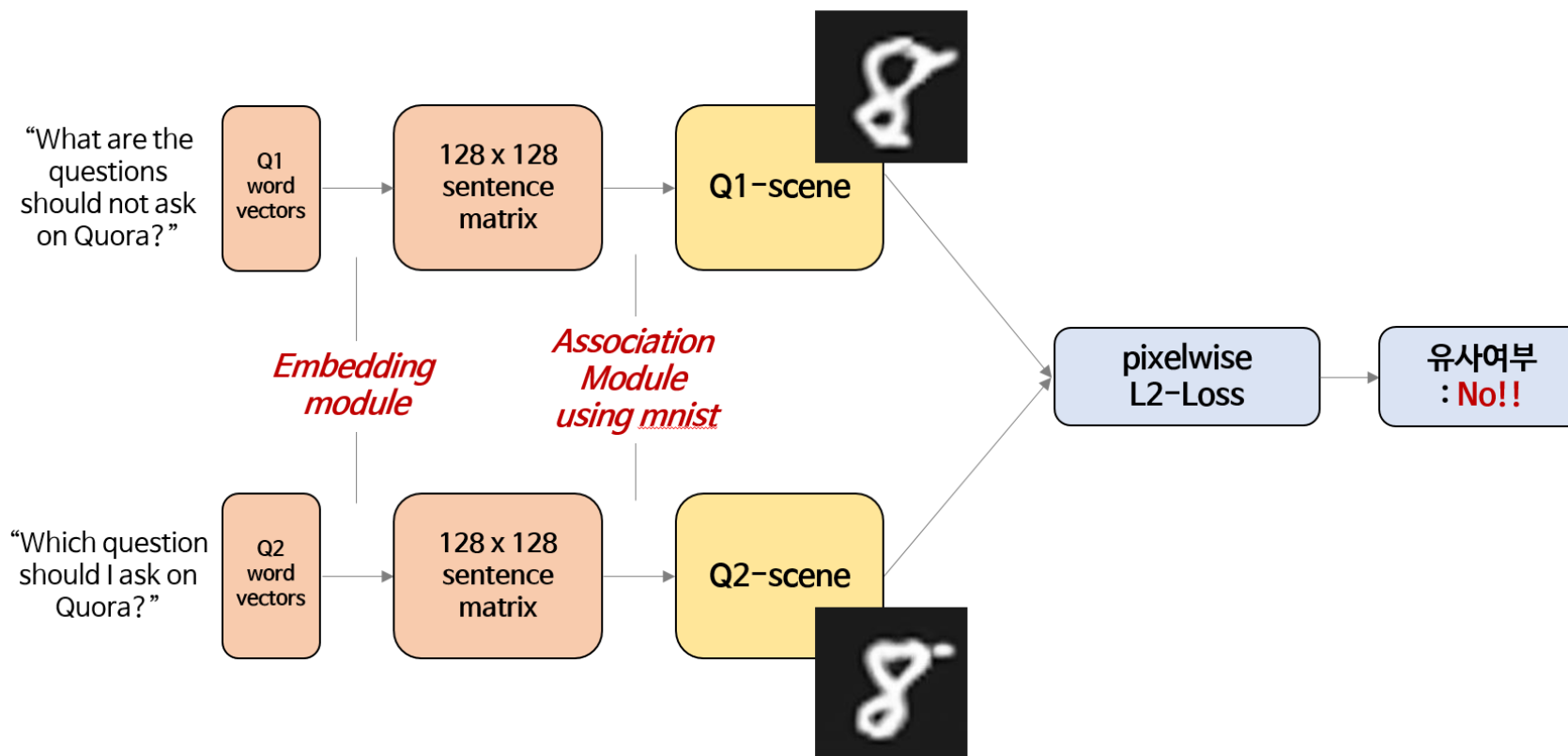


2. 이전 모델

2. 이전 모델

❖ Model Architecture

- 문장을 mnist 데이터에 빗대어 표현한 후, 유사여부를 판단하는 모델



2. 이전 모델

❖ Model Architecture(Cont'd)

1. **Embedding module**: 두 개의 질문에 각각 동일한 self-attention을 적용해 sentence matrix 추출



2. **Association module**: self-attention의 output과 mnist image에 대해 CycleGAN 적용



3. **Pixelwise L2-Loss**: 두 질문을 mnist에 빗대어 표현한 CycleGAN의 output 간의 Loss 계산

2. 이전 모델

❖ 결과 및 문제점

- 기존 CycleGAN의 Loss로 실험을 진행한 결과 문장이 mnist에 빗대어 표현되지만, label에 관계없이 비슷한 mnist 이미지화가 진행됨
- Generator의 loss에 실제 label에 대한 loss term이 존재하지 않기 때문에 발생하는 현상으로 생각됨



2. 이전 모델

❖ 결과 및 문제점 (Cont'd)

- 실제 이미지로 연상이 불가능한 문장을 이미지인 mnist에 빗대어 표현함

Q1: What are the questions should not ask on Quora?

Q2: Which question should I ask on Quora?



실제 이미지로 연상 불가능

2. 이전 모델

❖ 개선 방안

- 연상작용 모듈의 Generator에 유사도에 관한 loss 추가
- 실제 이미지로 연상이 가능한 text와 image로 이루어진 데이터의 사용

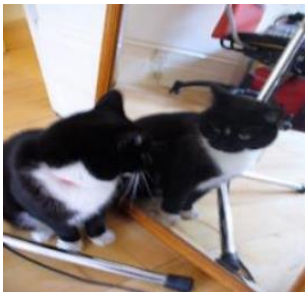
<p>A young boy is playing basketball.</p> 	<p>Two dogs play in the grass.</p> 	<p>A dog swims in the water.</p> 	<p>A little girl in a pink shirt is swinging.</p> 
<p>A group of people walking down a street.</p> 	<p>A group of women dressed in formal attire.</p> 	<p>Two children play in the water.</p> 	<p>A dog jumps over a hurdle.</p> 

3. 데이터 설명

3. 데이터 설명

❖ MS COCO Image Caption Data

- 1개의 이미지와 해당 이미지에 대한 의미가 동일한 서로 다른 5개의 caption pair로 구성된 데이터
- 동일한 이미지에 대한 caption은 유사한 문장, 다른 이미지에 대한 caption은 유사하지 않은 문장으로 가정하고 2문장의 pair로 40000개의 train data 생성






- there is a black tuxedo cat looking in the mirror
- two cats sitting on top of a wooden floor
- a cat looking at itself in the mirror next to a tripod
- a cat and a tripod sitting in front of a mirror
- a close up of a cat in a mirror



- a dog looking at a cat through a glass window
- a cat is outside looking through in at a dog
- the dog wants to go outside with the cat
- a cat sitting outside of a door next to a dog
- a cat sitting at a sliding glass door



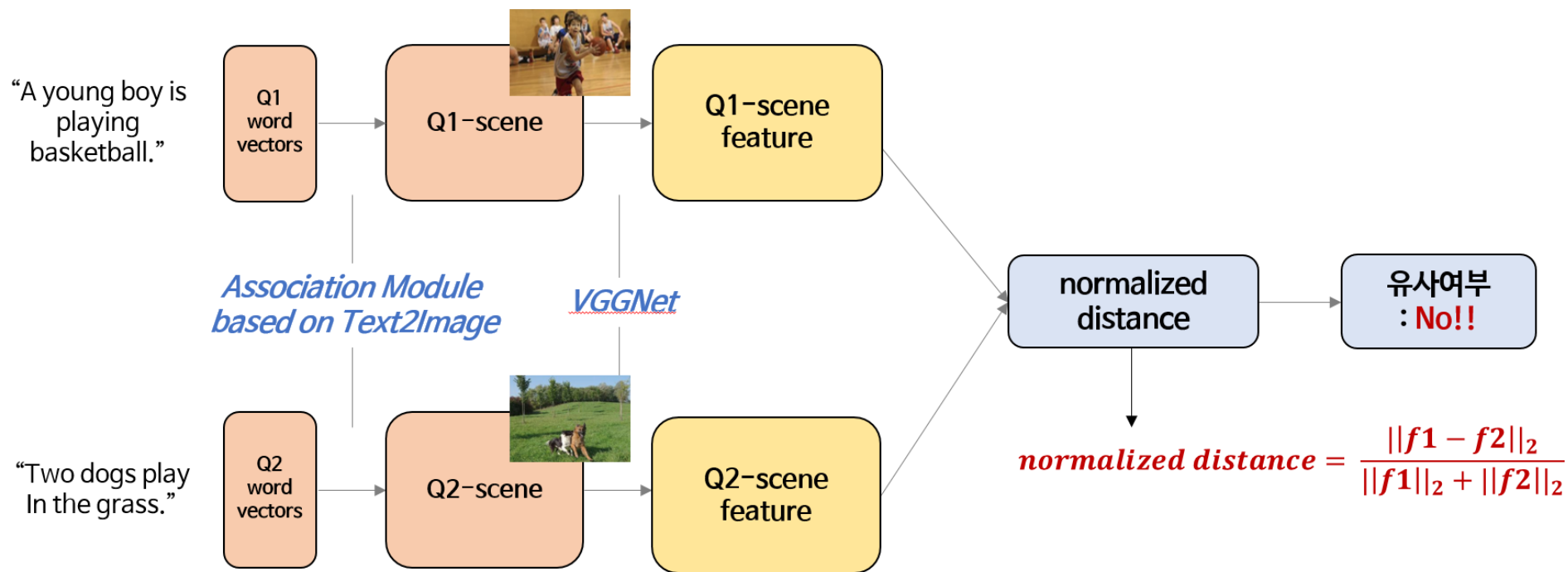
데이터	Input (sentence, image)	Output (유사여부)
Obs1	<div>- there is a black tuxedo cat looking in the mirror</div> <div>- a cat and a tripod sitting in front of a mirror</div> <div></div>	Y
Obs2	<div>- there is a black tuxedo cat looking in the mirror</div> <div>- a cat sitting at a sliding glass door</div> <div> </div>	N

4. 모델 설명

4. 모델 설명

❖ Model Architecture

- 문장을 이미지로 표현한 후, 연상된 이미지에서 뽑은 feature를 이용해 문장간 유사여부를 판단하는 모델

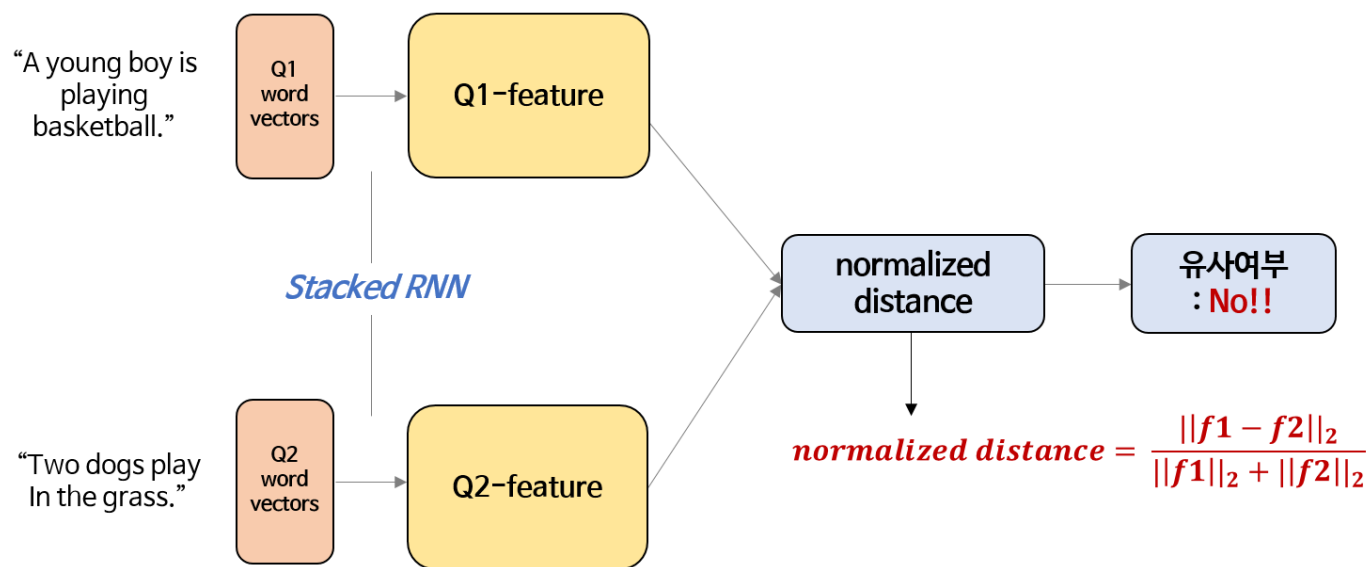


4. 모델 설명

❖ Model Architecture (Cont'd)

- Siamese Network

: **symmetric 구조**를 통해 2개의 input에 **동일한 모델을 적용**하고 해당 output에 대해 **distance metric**을 기준으로 유사여부를 판단하는 모델

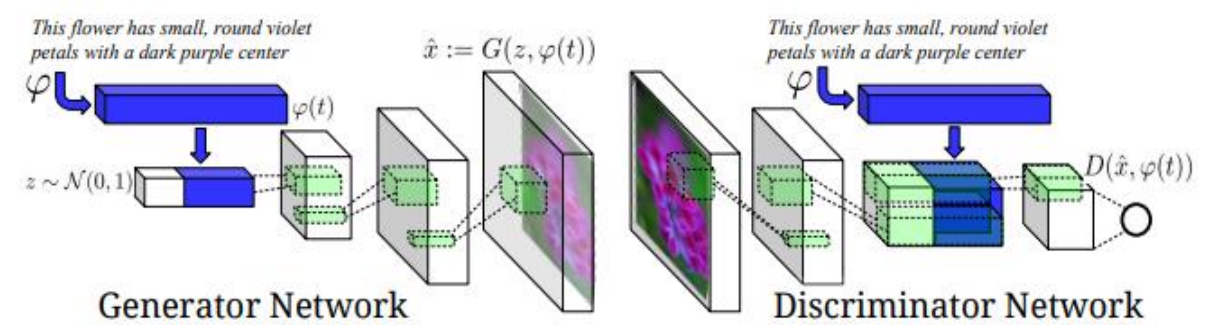


〈 기존 문장 유사도 비교 Siamese network 구조 〉

4. 모델 설명

❖ Model Architecture (Cont'd)

- Text2Image
: Generator와 Discriminator를 이용해 **input text와 매칭되는 이미지를 생성하는 모델**



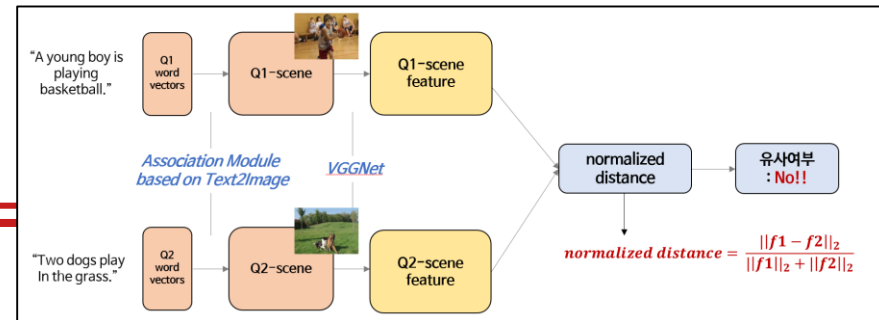
모델	Input	구조	역할
Generator	text, noise	Deconvolution Layer	text와 noise로 text와 매칭되는 이미지를 생성하는 역할
Discriminator	① (real image, right text) ② (real image , wrong text) ③ (fake image, right text)	Convolution Layer	① real image가 real인지 판단하고, ②text와 image가 매칭되는지 판단하고, ③ fake image를 fake인지 판단하는 역할

4. 모델 설명

❖ Model Architecture (Cont'd)

■ Siamese Network using Text2Image

: 문장을 이미지로 표현한 후, 연상된 이미지에서 뽑은 feature를 이용해 문장간 유사여부를 판단하는 모델



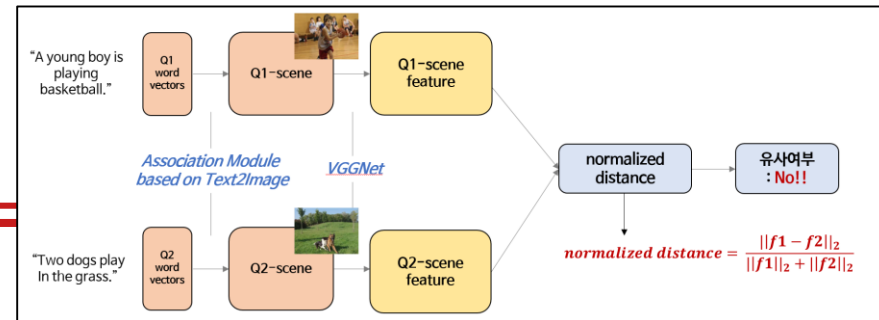
모델	Input	구조	역할
Generator	text, noise, label	Deconvolution Layer	text와 noise로 text와 매치되는 이미지를 생성하되 유사한 text들의 이미지는 더 유사하게 생성하는 역할
Discriminator	① (real image, right text) ② (real image, wrong text) ③ (fake image, right text)	Convolution Layer	① real image가 real인지 판단하고, ② text와 image가 매칭되는지 판단하고, ③ fake image를 fake인지 판단하는 역할
VGGNet	generated image By Generator	Convolution Layer	Generator가 text를 기반으로 생성한 이미지의 feature를 추출하는 역할

4. 모델 설명

❖ Model Architecture (Cont'd)

- Siamese Network using Text2Image

: label=1 (유사함) text pair를 기반으로 생성된 이미지들의 차이를 감소시키고, 반면 label=0 (유사하지 않음) text pair를 기반으로 생성된 이미지들의 차이를 증가시키는 Loss를 기존 Text2Image의 Generator Loss에 추가함



$$Loss_G = \frac{1}{2} \{ (1 - D(Q1 - scene)) + (1 - D(Q2 - scene)) \}$$



$$Loss_G = \frac{1}{2} \{ (1 - D(Q1 - scene)) + (1 - D(Q2 - scene)) \} + \lambda * Loss_{Contrastive}(d(Q1 - scene, Q2 - scene), y)$$

Note) $d(f1, f2) = \frac{\|f1 - f2\|_2}{\|f1\|_2 + \|f2\|_2}$, $Loss_{Contrastive}(d, y) = \frac{y * d + (1 - y) * MAX((1 - d), 0)}{2}$

5. 실험 결과

5. 실험 결과

❖ 모델 비교

- Baseline model

- : 1) Siamese Network using Stacked RNN
- 2) Siamese Network using vanilla Text2Image



$$Loss_G = \frac{1}{2} \{ (1 - D(Q1 - scene)) + (1 - D(Q2 - scene)) \}$$

- Ours

- : Siamese Network using Text2Image



$$Loss_G = \frac{1}{2} \{ (1 - D(Q1 - scene)) + (1 - D(Q2 - scene)) \} + \lambda * Loss_{Contrastive}(d(Q1 - scene, Q2 - scene), y)$$

5. 실험 결과

❖ Test Accuracy

- test data에 대한 accuracy
: 동일한 환경에서 2000개의 test data에 대해 분류를 진행한 결과, 2개의 baseline model보다 좋은 성능을 보임

Model	Stacked RNN	Vanilla Text2Image	Ours
Accuracy	0.6537	0.6882	0.7005

5. 실험 결과

❖ Feature Embedding

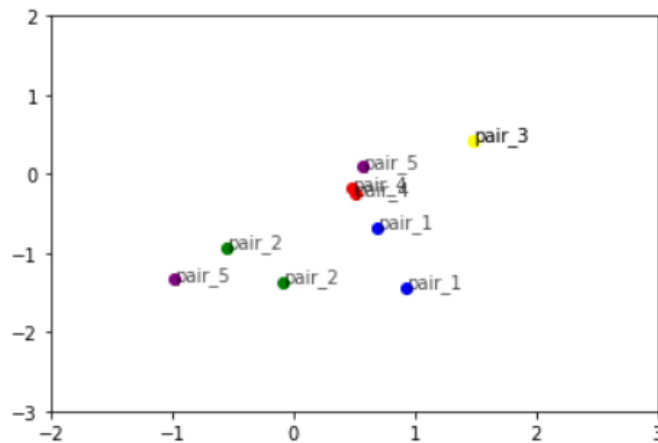
- Label별 feature embedding간 거리의 평균
: 2000개의 test data에 대해 추출한 VGGnet의 feature pair를 tSNE를 통해 차원축소한 결과, **baseline model**보다 **label=1 (유사함)**의 경우 두 문장간 거리가 더 가깝고, **label=0 (유사하지않음)**의 경우 두 문장간 거리가 먼 결과를 보임

Model	Vanilla Text2Image	Ours
distance(label=1)	1.81	1.69
distance(label=0)	3.23	3.46

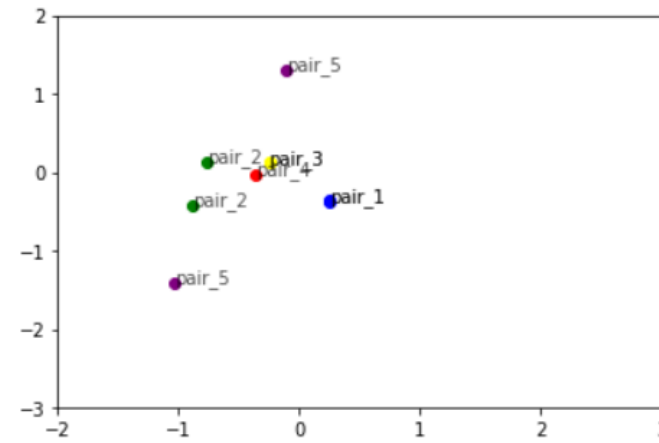
5. 실험 결과

❖ Feature Embedding (Cont'd)

- label=1인 test data의 feature embedding
: label=1 (유사함)의 경우 baseline model보다 두 문장의 feature가 더 가까이 embedding된 결과를 보임



〈Vanilla Text2Image〉

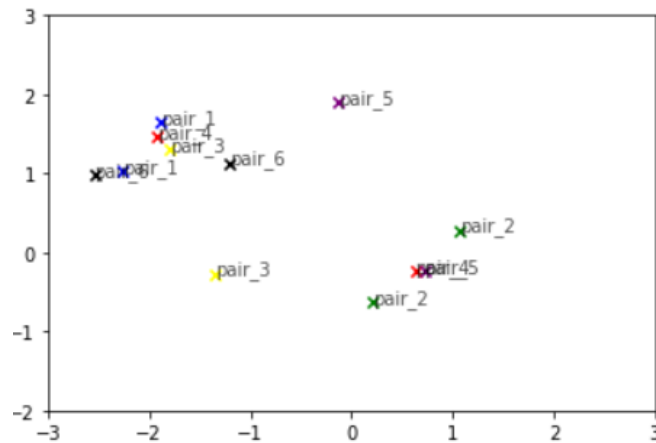


〈Ours〉

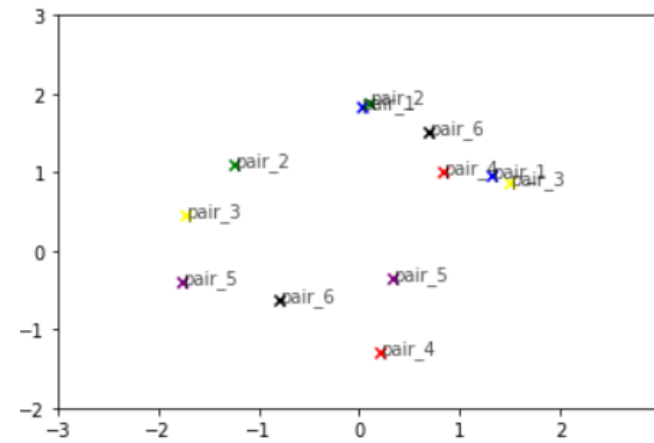
5. 실험 결과

❖ Feature Embedding (Cont'd)

- label=0인 test data의 feature embedding
: label=0(유사하지않음)의 경우 baseline model보다 두 문장의 feature가 더 멀리 embedding된 결과를 보임



〈Vanilla Text2Image〉



〈Ours〉

5. 실험 결과

❖ Contribution

- 사람처럼 연상작용을 통해 유사여부를 판단할 수 있는 모델을 제시함
- end-to-end 학습이 가능한 Association model 구성
- 문장의 유사여부 판단이 아닌 다양한 task로의 연상작용의 적용 가능성

감사합니다