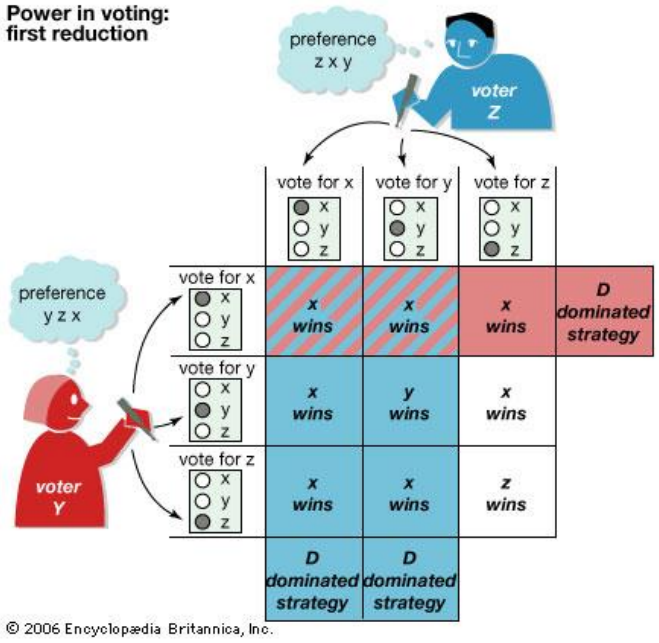
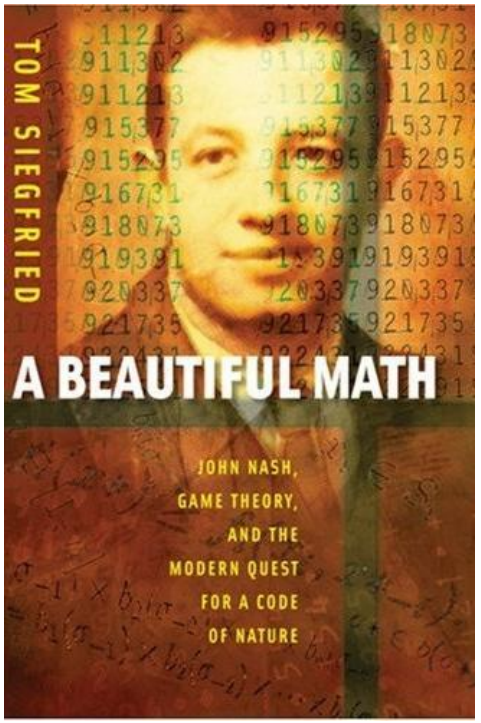
# GAN based Korean summarizer using semi abstractive method

2021.2.20



# Abstract

최근 NLP 분야는 encoder, decoder 및 attention을 거쳐 각종 transformer 모델이 나오면서 비약적인 발전을 이루고 있다. 문서 요약도 transformer에 의해 추상적인 요약이 가능해 졌다. 하지만 한국어의 경우, 양질의 대량 요약 학습셋이 갖추어져 있지 않아, 추상적인 요약 모델 개발 진행이 느리다. 비지도학습의 추출 요약을 사용할 수 있으나, 전반적 내용이 아닌 지협적 요약이 이루어진다는 단점이 있다. 이러한 단점을 극복하기 위해 본 연구에서는 요약 학습셋 없이 원문에서 요약에 필요한 어절을 추출하여 GAN 알고리즘에 의해 새로운 요약문을 생성하는 반추상적 방법 (Semi Abstractive Method, SAM) 을 제안한다. 실험에 의해 비지도학습으로 문서 전반적인 내용의 유사성과 문법의 정합성을 유지하며 요약이 되어짐을 확인하여 본 제안 방법의 우수성을 입증한다.

# Introduction

- 추출 텍스트 요약

먼저 개발 된 전통적인 방법으로서 주요 목표는 텍스트의 중요한 문장을 식별하고 요약에 추가하는 것이다. 얻은 요약에는 원본 텍스트의 정확한 문장이 포함되어 있다는 점에서 많이 활용되고 있다.

- 추상적인 텍스트 요약

그것은 더 진보 된 방법이며, 최근 많은 연구가 진행되고 있다. 접근 방식은 중요한 섹션을 식별하고 컨텍스트를 해석하며 새로운 방식으로 재생산하는 것이다. 이렇게하면 핵심 정보가 가능한 가장 짧은 텍스트를 통해 전달된다. 여기서는 원본 텍스트에서 추출한 문장이 아니라 요약 된 새로운 문장이 생성된다.

# Related Work

- 16 text generation models

| Category | Task Type | Model | Reference |
|---|---|---|---|
| VAE | Unconditional | LSTMVAE | (Bowman et al., 2016) |
| | | CNNVAE | (Yang et al., 2017) |
| | | HybridVAE | (Semeniuta et al., 2017) |
| GAN | | SeqGAN | (Yu et al., 2017) |
| | | TextGAN | (Zhang et al., 2017) |
| | | RankGAN | (Lin et al., 2017) |
| | | MaliGAN | (Che et al., 2017) |
| | | LeakGAN | (Guo et al., 2018) |
| | | MaskGAN | (Fedus et al., 2018) |
| Seq2Seq | Translation<br>Summarization | RNN | (Sutskever et al., 2014) |
| | | Transformer | (Vaswani et al., 2017b) |
| | | GPT-2 | (Radford et al.) |
| | | XLNet | (Yang et al., 2019) |
| | | BERT2BERT | (Rothe et al., 2020) |
| | | BART | (Lewis et al., 2020) |

- Text dataset

| Task | Dataset |
|---|---|
| Unconditional | Image COCO Caption |
| | EMNLP2017 WMT News |
| | IMDB Movie Review |
| Translation | IWSLT2014 German-English |
| | WMT2014 English-German |
| Summarization | GigaWord |

https://github.com/RUCAIBox/TextBox

# Related Work

- 16 text generation models

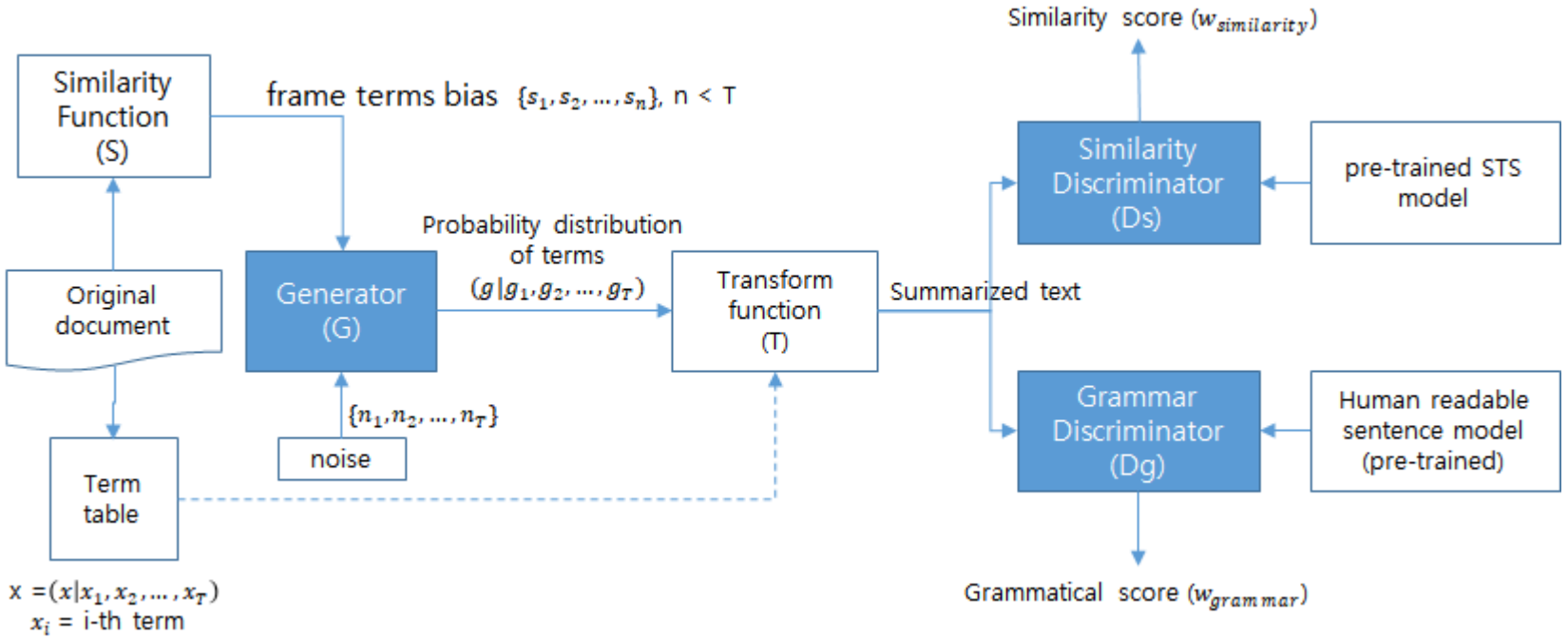
| Category | Task Type | Model | Reference |
|---|---|---|---|
| VAE | Unconditional | LSTMVAE | (Bowman et al., 2016) |
| | | CNNVAE | (Yang et al., 2017) |
| | | HybridVAE | (Semeniuta et al., 2017) |
| GAN | | SeqGAN | (Yu et al., 2017) |
| | | TextGAN | (Zhang et al., 2017) |
| | | RankGAN | (Lin et al., 2017) |
| | | MaliGAN | (Che et al., 2017) |
| | | LeakGAN | (Guo et al., 2018) |
| | | MaskGAN | (Fedus et al., 2018) |
| Seq2Seq | Translation<br>Summarization | RNN | (Sutskever et al., 2014) |
| | | Transformer | (Vaswani et al., 2017b) |
| | | GPT-2 | (Radford et al.) |
| | | XLNet | (Yang et al., 2019) |
| | | BERT2BERT | (Rothe et al., 2020) |
| | | BART | (Lewis et al., 2020) |

- Text dataset

| Task | Dataset |
|---|---|
| Unconditional | Image COCO Caption |
| | EMNLP2017 WMT News |
| | IMDB Movie Review |
| Translation | IWSLT2014 German-English |
| | WMT2014 English-German |
| Summarization | GigaWord |

https://github.com/RUCAIBox/TextBox

# Proposed Method




- Generator는 원문 어절의 개수 만큼의 noise를 input으로 받으며 원문 어절의 개수 만큼의 output
- output은 각 원문의 어절이 요약에 활용되는 확률값
- 특정 확률값 또는 주어진 어절의 개수에 해당하는 상위 순위의 확률값에 해당하는 어절을 선택
- Similarity function에 의해서 원문으로부터 골자 어절들(frame terms)을 추출
- 이를 Generator에 bias로 입력
- Generator는 주요 어절들에 높은 확률값을 우선 할당
- 한국어 문법 discriminator에 의해 생성된 문장이 문법적으로 사람이 읽을 수 있는 문장인지 구분
- 유사도 discriminator에 의해 임의 생성된 문장이 원문 내용과 상이하지 않도록 조정

# Proposed Method

## 3.1 Similarity function을 이용한 frame word의 추출

Document의 terms (or words, tokens), 띄어쓰기 단위의 어절들 = x

$$x = \{x_1, x_2, \ldots, x_T\}$$

Similarity Probability Distribution (SPD)

$$SPD_{for\ one\ term} = \{P_s(x_1), P_s(x_2), \ldots, P_s(x_T)\}$$

$$SPD_{for\ N\ term} = \{P_s(x_1, x_2, \ldots, x_N), P_s(x_2, x_3, \ldots, x_{N+1}), \ldots, P_s(x_{T-N}, x_{T-N+1}, \ldots, x_T)\}$$

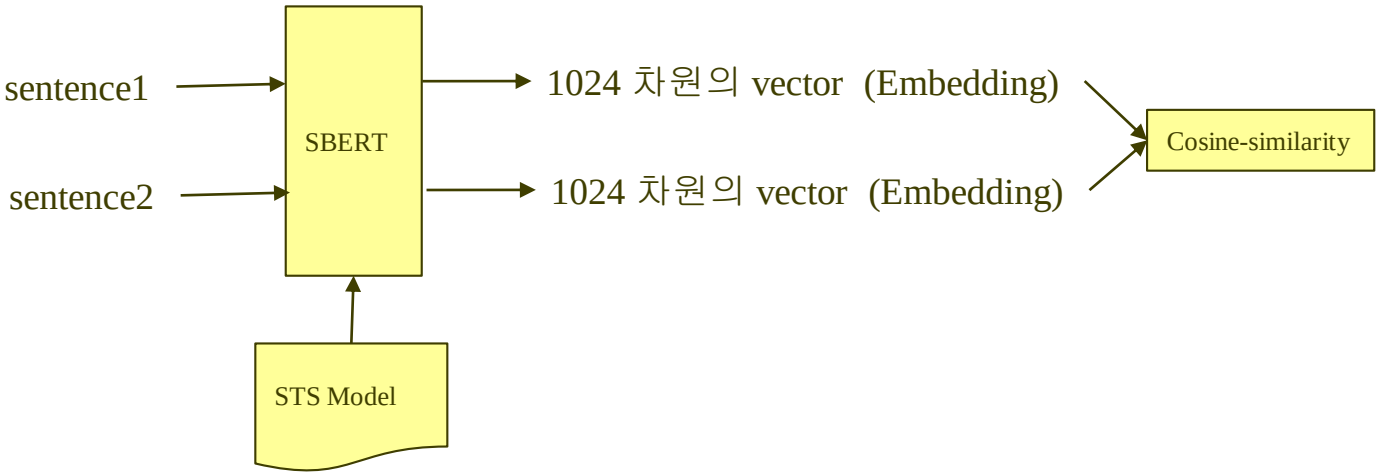
$$s = peak(\sum_{i=1}^{m} SPD_i)$$

$$s = peak(\sum_{i=2}^{3} SPD_i - SPD_1) \quad \Rightarrow \quad s = peak(\sum_{i=2}^{3} SPD_i) - peak(SPD_1)$$

# Proposed Method

## 3.2 Similarity function, S (Sentence-BERT)

https://www.sbert.net/docs/pretrained_models.html




In this paper, the'xlm-r-large-en-ko-nli-ststb' model learned in Korean was used, and the STS Benchmark score of this model was 84.05%.

# Proposed Method

3.3 한국어 문법 Discriminator, D

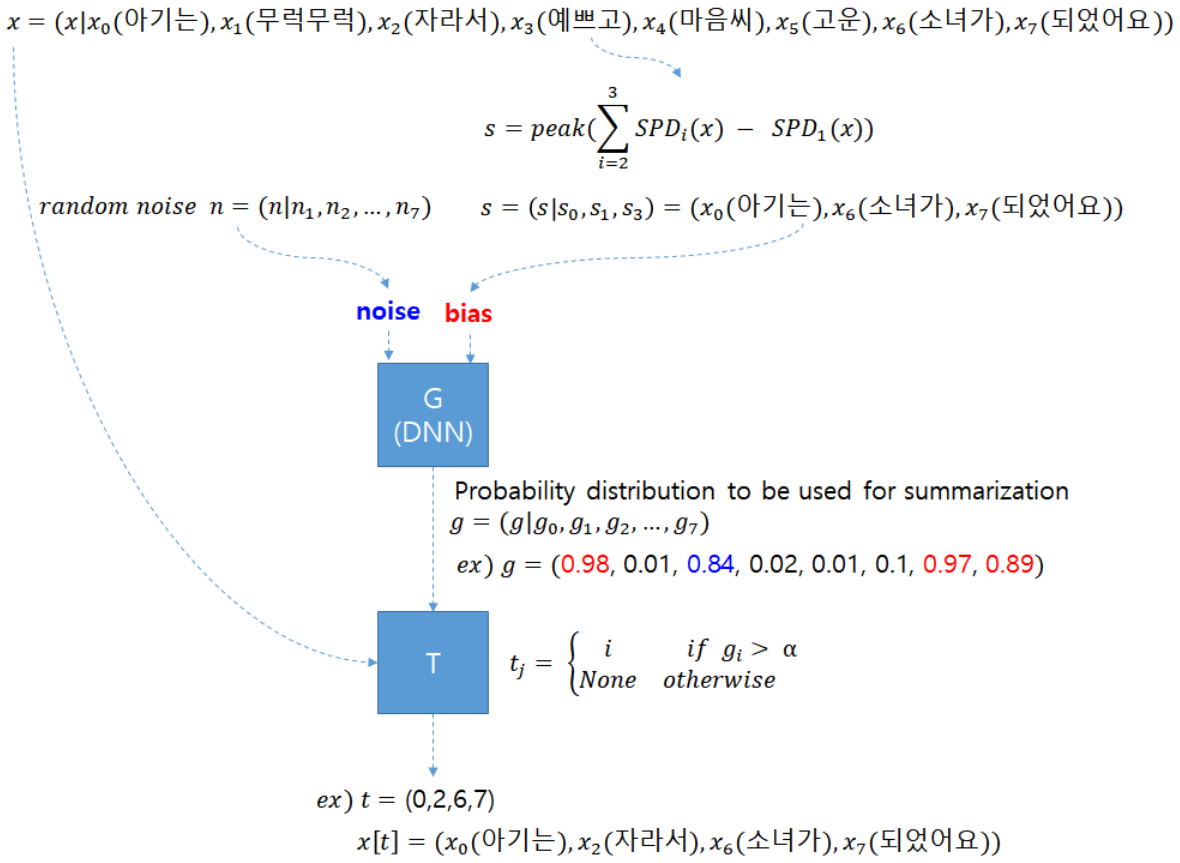
- 단순히 문법적으로 정상문장(1)과 이상문장(0)을 구분하는 Classification
- BERT로 구성

  https://analyticsindiamag.com/how-to-use-bert-transformer-for-grammar-checking/

- 한국어 소설에서 문장 3만개 추출, 이상문장은 정상문장의 어절 순서를 임의 섞어서(shuffling) 생성
- 같은 문장에 대해 정상, 이상이 Dataset에 같이 있다 → 하나의 문장에서 정상 or 이상 같은 비율로 둘 중에 하나만 반영해야 할 듯

# Proposed Method

## 3.4 한국어 Text Generator, G



# Proposed Method

## 3.5 GAN Training

Original GAN의 목적함수 :

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

제안 기법의 GAN 목적함수 :

$$\min_G \max_{D_g,D_s} V(D_g,D_s,G) = \mathrm{E}_{x\sim P_{data(x)}}[logD_g(x)] + \mathrm{E}_{x\sim P_{data(x)}}[logD_s(x)]$$
$$+ \mathrm{E}_{z\sim P_{data(z)}}[log(1 - D_g(T(G(z))))] + \mathrm{E}_{z\sim P_{data(z)}}[log(1 - D_s(T(G(z))))]$$

T함수는 미분 불가능, 이를 해결하기 위해 Wasserstein GAN, https://arxiv.org/pdf/1701.07875.pdf) 을 적용

T(G(z))의 output의 예를 들면 다음과 같다. (t|t1,t2,...tT) ~ (t|g1,0,0,g4,0,g6,...,0,gT)

D(T(G(z))) = w

제안 기법의 GAN 목적함수 :

$$\min_G \max_{D_g,D_s} V(D_g,D_s,G) = \mathrm{E}_{x\sim P_{data(x)}}[logD_g(x)] + \mathrm{E}_{x\sim P_{data(x)}}[logD_s(x)]$$
$$- \frac{1}{T}\sum_i^T (w_{D_g} + w_{D_s}) t_{i,G(z)}$$

그러나, 학습의 과정에서 $\mathrm{E}_x$ 를 정의 할 수 없다. 이미, 요약된 실데이터를 만들수 없기 때문이다. 그래서, 원문의 일부 문장을 대입하였더니, 학습이 진행되면서 원문의 일부로 overfitting 되는 경향이 나타났다. 즉, Extractive 방법의 결과로 수렴되는 것이다. 그래서, 본 논문에서는 Discriminator가 이미 pre-trained model에 의해 학습되어 있으므로 Discriminator를 학습에 참여 시키지 않았다. 따라서 최종 목적함수는 아래와 같이 단순화 하였다.

$$\min_G V(G) = -\frac{1}{T}\sum_i^T (w_{D_g} + w_{D_s}) t_{i,G(z)}$$

# Experiment

## 4.1 Experiment plan

한국어의 abstractive summarizer는 현재 비교 대상이 없기 때문에, 한국어 BERT를 이용하여 Extractive 방법으로 요약한 결과와 비교 검증한다.

1) sentence-transformer에 의해 BERT기반으로 문장별 Embedding한 후 LexRank으로 Extractive 요약한 방법
2) bert-extractive-summarizer 방법
3) bert-extractive-summarizer + koBERT 모델 적용 방법

과 이번 제안 방법의 Similarity 를 비교 한다.
하지만 전체 스토리를 포함여부를 판단하기 위해, 원문을 3등분하여 각 부분과 유사도를 어떻게 갖는지 확인 한다. ***비교적 단문 스토리와 장문 스토리를 구분***하여 실험한다.

# Experiment

## 4.1 Experiment plan

**[단문 스토리] 선녀와 나뭇꾼 이야기**

도입부

나무꾼이 나무를 하다가 숲 속에서 도망치는 사슴을 만났는데, 이 사슴이 사냥꾼이 쫓아오고 있으니 자신을 숨겨달라고 말했다. 말하는 사슴을 신기하게 여긴 나무꾼이 사슴을 숨겨줬고, 뒤쫓아 온 사냥꾼을 다른 방향으로 보내서 구해주었다.

중간

사슴은 은혜를 갚겠다고 하면서, 나무꾼에게 선녀들이 하늘에서 내려와서 목욕하는 선녀탕이라는 샘과 선녀들이 목욕을 하러 오는 시기, 선녀의 옷을 훔쳐 그를 아내로 삼도록 하는 꾀를 나무꾼에게 가르쳐 주었다. 나무꾼은 반신반의 하면서도 사슴이 가르쳐준 시기에 선녀들이 목욕을 하러 내려온다는 샘으로 찾아가 몸을 숨겼다. 그렇게 잠시간 기다리자 과연, 선녀들이 하늘에서 내려와 날개옷을 벗고 선녀탕에서 목욕을 하는 것이었다. 나무꾼은 사슴이 가르쳐준 대로 날개옷을 하나 훔쳤다.

결말부

날개옷이 없어진 탓에 한 명의 선녀는 하늘로 올라가지 못했으며 다른 선녀들은 날개옷이 없는 선녀를 내버려두고 하늘로 돌아갔다. 이때 나무꾼이 홀로 남은 선녀에게 자신의 아내가 되어달라고 하자, 하늘나라로 올라가지 못하게 된 선녀는 할 수 없이 나무꾼에게 의탁하게 되었다.

**[장문 스토리] 신데렐라**

도입부

옛날 어느 집에 귀여운 여자 아기가 태어났어요. 아기는 무럭무럭 자라서, 예쁘고 마음씨 고운 소녀가 되었어요. 그러던 어느날, 소녀의 어머니가 병이들어 그만 세상을 떠나고 말았어요. 소녀의 아버지는 홀로 남은 소녀가 걱정되었어요. 그래서 얼마 후 새어머니를 맞이했어요. 새어머니는 소녀보다 나이가 위인 두 딸을 데리고 왔어요. 그러나 새어머니와 언니들은 성질이 고약한 심술쟁이들이었어요. 새어머니는 소녀가 자기 딸들보다 예쁘고 착한 게 못마땅했어요. 그런데 이번에는 아버지마저 돌아가셨어요. 소녀는 하녀처럼 하루 종일 쓸고, 닦고, 집안일을 도맡아 했어요. 해도 해도 끝이 없는 집안일이 힘들어 지칠때면 난롯가에 앉아서 잠시 쉬곤 했지요. "엄마, 저애를 신데렐라라고 불러야겠어요." "온통 재투성이잖아요. 호호호!" 두 언니는 소녀를 놀려 댔어요.

중간

어느 날, 왕궁에서 무도회가 열렸어요. 신데렐라의 집에도 초대장이 왔어요. 새어머니는 언니들을 데리고 무도회장으로 떠났어요. 신데렐라도 무도회에 가고 싶었어요. 혼자 남은 신데렐라는 훌쩍훌쩍 울기 시작했어요. "신데렐라, 너도 무도회에 가고 싶니?" 신데렐라가 고개를 들어보니, 마법사 할머니가 빙그레 웃고 있었어요. "내가 너를 무도회에 보내주마 호박 한개와 생쥐 두마리, 도마뱀을 구해 오렴." 마법사 할머니가 주문을 외웠어요. 그리고 지팡이로 호박을 건드리자, 호박이 화려한 황금 마차로 변했어요. 이번에는 생쥐와 도마뱀을 건드렸어요. 그랬더니 생쥐는 흰말로, 도마뱀은 멋진 마부로 변했답니다. 신데렐라의 옷도 구슬 장식이 반짝이는 예쁜 드레스로 바뀌웠어요. "신데렐라, 발을 내밀어 보거라." 할머니는 신데렐라에게 반짝반짝 빛나는 유리 구두를 신겨 주었어요. "신데렐라, 밤 열두시가 되면 모든게 처음대로 돌아간단다. 황금 마차는 호박으로, 흰말은 생쥐로, 마부는 도마뱀으로 변하게 돼. 그러니까 반드시 밤 열두 시가 되기 전에 돌아와야 해. 알겠지?" 왕자님도 아름다운 신데렐라에게 마음을 빼았겼어요. 왕자님은 무도회장에 모인 다른 아가씨들은 쳐다보지도 않고,신데렐라하고만 춤을 추었어요. 신데렐라는 왕자님과 춤을 추느라 시간 가는 줄도 몰랐어요. 땡, 땡, 땡..... 벽시계가 열두 시를 알리는 소리에 신데렐라는 화들짝 놀랐어요. 신데렐라가 허둥지둥 왕궁을 빠져나가는데, 유리 구두 한 짝이 벗겨졌어요. 하지만 구두를 주울 틈이 없었어요. 신데렐라를 뛰쫓아오던 왕자님은 층계에서 유리 구두 한 짝을 주웠어요. 왕자님은 유리 구두를 가지고 임금님께 가서 말했어요. "이 유리 구두의 주인과 결혼하겠어요."

결말부

그래서 신하들은 유리 구두의 주인을 찾아 온 나라를 돌아다녔어요. 언니들은 발을 오므려도 보고, 구두를 늘려도 보았지만 한눈에 보기에도 유리 구두는 너무 작았어요. 그때, 신데렐라가 조용히 다가와 말했어요. "저도 한번 신어 볼 수 있나요?" 신데렐라는 신하게 건넨 유리 구두를 신었어요, 유리 구두는 신데렐라의 발에 꼭 맞았어요. 신하들은 신데렐라를 왕궁으로 데리고 갔어요. 그 뒤 신데렐라는 왕자님과 결혼하여 오래오래 행복하게 살았대요.

# Experiment

4.2 실험결과

| method | Comp ratio | Similarity | | | | grammar |
|---|---|---|---|---|---|---|
| | | intro | body | end | total | |
| ***SAM+WGAN*** | ***0.2364*** | ***0.8497*** | ***0.6673*** | ***0.5484*** | ***0.6832*** | ***0.9986*** |
| BERT+LexRank | 0.2309 | 0.2268 | 0.5041 | 0.6728 | 0.4769 | 0.9998 |
| BESM | 0.2709 | 0.9007 | 0.5679 | 0.7900 | 0.7066 | 0.9998 |
| BESM+kobert | 0.2291 | 0.9007 | 0.7162 | 0.4640 | 0.6993 | 0.9998 |

Table 2 Summary comparison results for relatively short story (fairy and woodcutter story)

| method | Comp ratio | Similarity | | | | grammar |
|---|---|---|---|---|---|---|
| | | intro | body | end | total | |
| ***SAM+WGAN*** | ***0.1646*** | ***0.5166*** | ***0.5829*** | ***0.6458*** | ***0.5766*** | ***0.9986*** |
| BERT+LexRank | 0.1508 | 0.2258 | 0.2443 | 0.1812 | 0.2302 | 0.9999 |
| BESM | 0.1632 | 0.4112 | 0.6119 | 0.5997 | 0.5620 | 0.9998 |
| BESM+kobert | 0.1632 | 0.4112 | 0.6119 | 0.5997 | 0.5620 | 0.9998 |

Table 3 Summary comparison results for relatively long story (Cinderella story)

# Experiment

## 4.3 고려사항

**실제 결과 :** 나무를 하다가 숲 속에서 도망치는 이 사슴이 자신을 숨겨달라고 말했다. 사슴을 여긴 사슴은 선녀들이 하늘에서 목욕하는 선녀탕이라는 가르쳐준 선녀들이 하러 내려온다는 몸을 과연, 하늘에서 선녀탕에서 대로 없어진 탓에 하늘나라로 되었다.

- 주요한 단어, 어절들이 포함되므로 Similarity는 유사하게 나오나
- 문법적 학습 효과는 있으나 사람이 읽는 느낌은 틀리다. → 성능 향상이 필요함
- 결과에서 문법적 지표 차이 많지 않아 보이나, 0~1의 지수로 만들기 위해 tanh 를 넣어서 차이 없는 것 처럼 보일 뿐

# Conclution

본 논문에서 제안한 방법은 extractive 방법의 지협성을 해결 할 수 있었고, abstractive 방법에서 필요한 대량의 학습셋 없이도 WGAN의 비지도 학습에 의해서 abstractive의 효과를 얻을 수 있었다. 원문대비 유사성에 있어서는 비교적 짧은 스토리에서 Extractive 방법과 대등소이한 결과를 보였으나, 비교적 긴 스토리에서는 Extractive보다 Similarity가 서론, 본문, 결론 모두 높게 나타났다. 그러나, grammarlity는 가장 낮았다. 이는, 주요 어절을 기반으로 문법에 맞게 적당한 어절을 만들어내는 기본 알고리즘 구조상 나타나는 현상으로 이해할 수 있다. 일반 DNN구조로 만들어진 Generator이기 때문에 문법 구성이 부드럽지 못했다. 향후에는 seq2seq 알고리즘으로 Generator를 구성하면 문법적으로 더 완성도 높은 문장을 생성할 것으로 기대한다.

# 다음 논문

1. Generator의 개선
   seq2seq 알고리즘 or VAE

2. WGAN이외의 Distance 적용 (EM) 또는 새로운 방법 구상