

텍스트 세미나

---

ToBig's 13기 김민정

Chapter 15

# NLG

## Natural Language Generation

# Contents

---

Unit 01 | Decoding Algorithms

---

Unit 02 | Neural Summarization

---

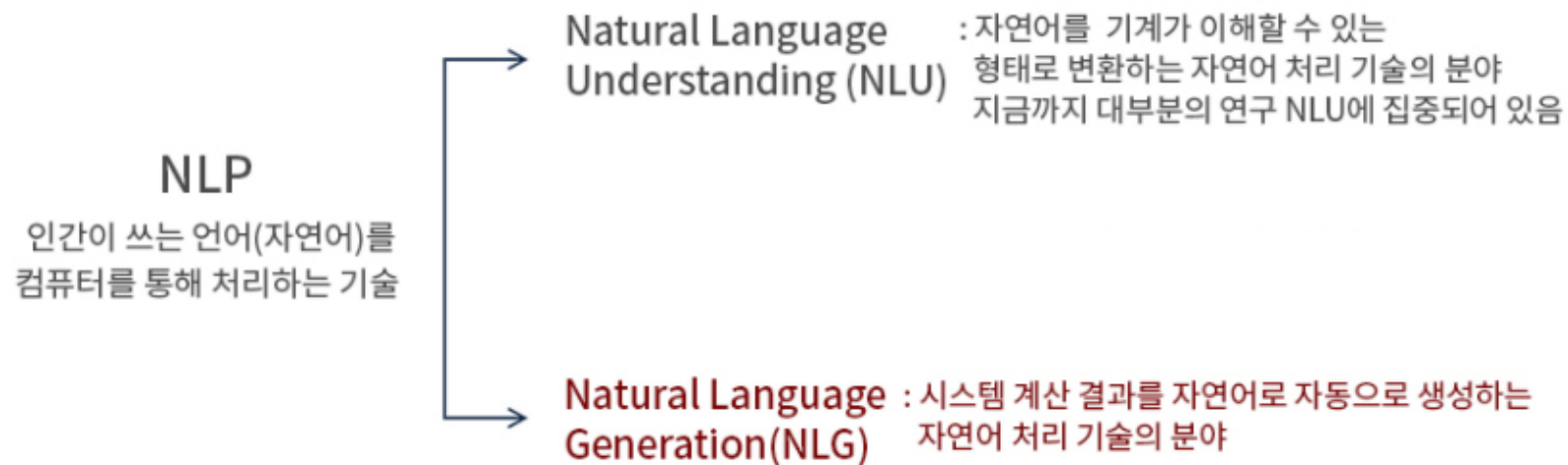
Unit 03 | Copy Mechanisms

---

Unit 04 | NLG using unpaired corpus

---

## 시작하기 전에...



## Unit 01 | Decoding Algorithms

### NLG (Natural Language Generation)

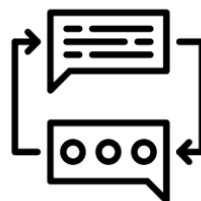
주어진 input x에 대해 새로운 text를 생성해내는 작업



Machine  
Translation



(Abstractive)  
Summarization



Dialogue  
(chit-chat)



(Creative)  
storytelling



QA

## Unit 01 | Decoding Algorithms

### 자연어를 잘 생성한다는 것의 의미

#### 적절성

생성된 문장이 모호하지 않고 원래의 input text의 의미와 일치해야 함

#### 유창성

문법이 정확하며 어휘를 적절하게 사용해야 함

#### 가독성

적절한 지시어, 접속사 등을 사용하여 문장의 논리 관계를 고려하여 생성해야 함

#### 다양성

상황에 따라 혹은 대상에 따라 표현을 다르게 생성해야 함

## Unit 01 | Decoding Algorithm

Q. LM을 학습한 후에 어떻게 NLG에 적용하지?

A) Decoding Algorithm

- 가장 가능성이 높은 출력 시퀀스를 디코딩하기 (x)  
: vocab의 크기가 어마어마한 경우에는 완전 탐색을 하기가 어렵기 때문
- 최대한 가능성이 높은 출력 시퀀스를 디코딩하기 (o)  
: 휴리스틱한 탐색 방법을 사용하자!

## Unit 01 | Decoding Algorithm

Q. LM을 학습한 후에 어떻게 NLG에 적용하지?

A) Decoding Algorithm

1. Greedy Decoding
2. Beam Search
3. Pure Sampling
4. Top-n Sampling

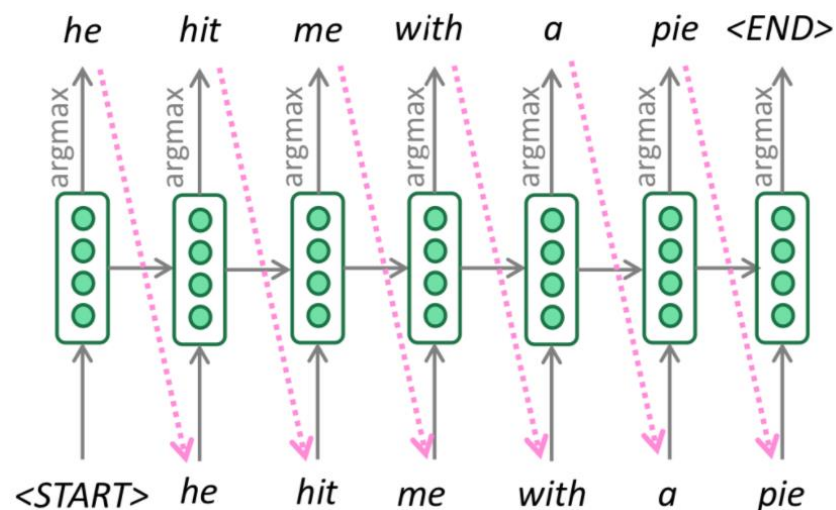
## Unit 01 | Decoding Algorithms

### 1. Greedy Decoding

각 출력을 예측하는데 매 스텝에서 가장 가능성이 높은 단어 **한 개**를 선택 (argmax)

장점) 탐색하는데 매우 빠름

단점) 최종 출력 결과가 좋지 않음





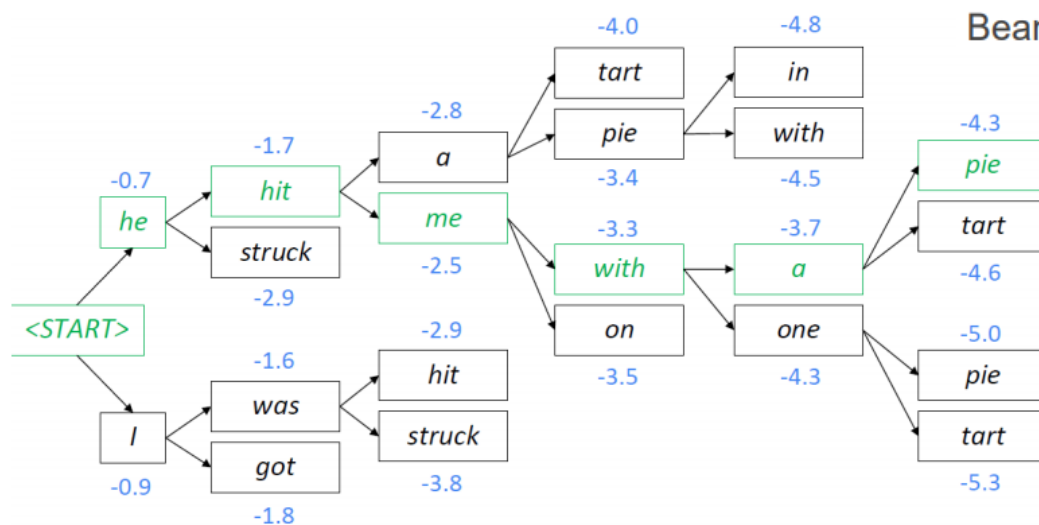
## Unit 01 | Decoding Algorithms

### 2. Beam Search

Greedy Decoding의 확장된 형태

$k$  개의 가능한 가설들을 두고 가장 높은 확률을 갖는 문장을 찾아 나가는 방식

beam size (hyper parameter)



Blue numbers =  $\text{score}(y_1, \dots, y_t)$

$$= \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$$

## Unit 01 | Decoding Algorithms

### 2. Beam Search

그러면 Beam size는 어떻게 고르지?

*I mostly eat a fresh and raw diet, so I save on groceries*



Human  
chit-chat  
partner

Beam size	Model response
1	<i>I love to eat healthy and eat healthy</i>
2	<i>That is a good thing to have</i>
3	<i>I am a nurse so I do not eat raw food</i>
4	<i>I am a nurse so I am a nurse</i>
5	<i>Do you have any hobbies?</i>
6	<i>What do you do for a living?</i>
7	<i>What do you do for a living?</i>
8	<i>What do you do for a living?</i>

Low beam size:  
More on-topic but  
nonsensical;  
bad English

beam size가 **너무 작으면**  
주제에 더 가깝지만 말이  
안되는 답변을 뱉음

High beam size:  
Converges to safe,  
"correct" response,  
but it's generic and  
less relevant

beam size가 **너무 크면**  
너무 generic하고 짧은 답변을 뱉으며  
BLEU score를 떨어뜨릴 수 있음

## Unit 01 | Decoding Algorithms

### 3. Sampling-based decoding

Q. 큰  $k$ 를 가지더라도 너무 generic하지 않도록 할 수는 없을까?

A) Sampling-based decoding

## Unit 01 | Decoding Algorithms

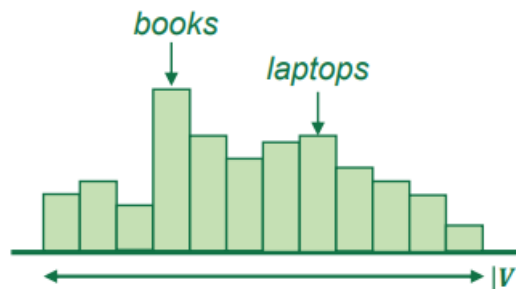
### 3. Sampling-based decoding

Q. 큰  $k$ 를 가지더라도 너무 generic하지 않도록 할 수는 없을까?

A) Sampling-based decoding

#### 1) Pure Sampling

Greedy Decoding과 비슷하지만,  $\text{argmax}$  대신 sampling을 사용함



## Unit 01 | Decoding Algorithms

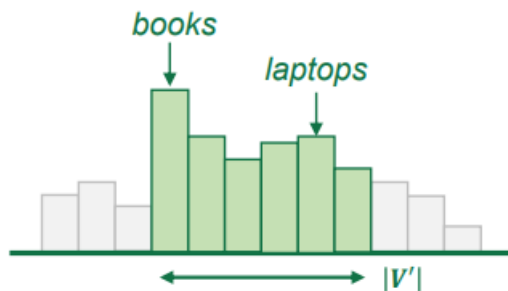
### 3. Sampling-based decoding

Q. 큰  $k$ 를 가지더라도 너무 generic하지 않도록 할 수는 없을까?

A) Sampling-based decoding

#### 2) Top-n Sampling

Pure Sampling처럼 완전하게 랜덤 샘플링을 하는 것이 아니라 확률이 가장 큰  $n$ 개의 단어들 중에서 랜덤 샘플링을 함



# Contents

---

Unit 01 | Decoding Algorithms

---

Unit 02 | Neural Summarization

---

Unit 03 | Copy Mechanisms

---

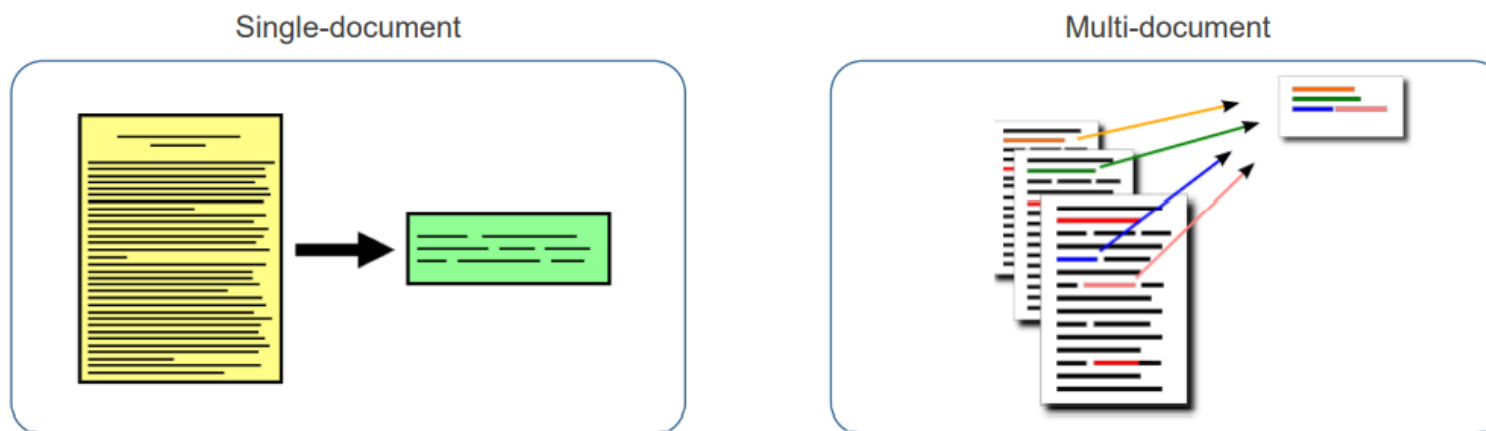
Unit 04 | NLG using unpaired corpus

---

## Unit 02 | Neural Summarization

### Summarization



주어진 input  $x$ 에 대해,  $x$ 의 주요 정보를 포함하는 요약된  $y$ 를 생성해내는 작업



## Unit 02 | Neural Summarization

<https://summariz3.herokuapp.com/>

### Summarization

Extractive Summarization	Abstractive Summarization
	
문서 내에서 핵심이 되는 문장을 추출 비교적 쉽지만 제한적인 요약 결과	문서의 중요한 내용을 담은 새로운 문장을 생성 보다 유연한 결과를 얻을 수 있지만 비교적 어려움



## Unit 02 | Neural Summarization

### Summarization

#### Extractive Summarization

원문 : delighted food looks nice smells really great smaller size kibble old dog teeth comes ziplock pouch importantly though old dog getting little suddenly gotten since starting chow also noticed terrible daily g one fair time switched wet food harmony foods may also factor extremely happy food **continue** buy

실제 요약문 : highly recommend this

예측 요약문 : my dog loves this

원문 : dog loves stuff ground sprinkled dry food gobbles additives fillers carbs also use treat best price amazon quick delivery

실제 요약문 : great

예측 요약문 : great dog food

원문 : yuck worst chocolate ever save money brand find another even taste taste like chocolate threw rest a way

실제 요약문 : horrible chocolate

예측 요약문 : awful

비교적 쉽지만 제한적인 요약 결과

#### Abstractive Summarization



문서의 중요한 내용을 담은 새로운 문장을 생성  
보다 유연한 결과를 얻을 수 있지만 비교적 어려움

## Unit 02 | Neural Summarization

### Evaluation Score : ROUGE

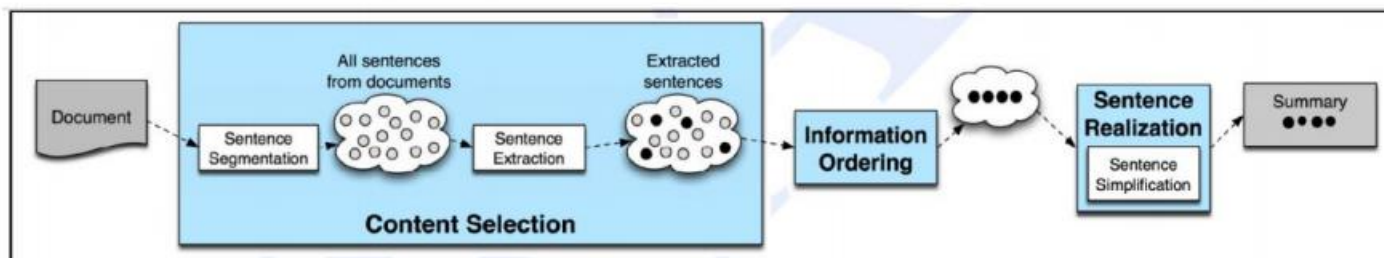
BLEU score	ROUGE score
$\text{BLEU} = \min\left(1, \frac{\text{output length}}{\text{reference length}}\right) \left(\prod_{i=1}^4 \text{precision}_i\right)^{\frac{1}{4}}$	<p>ROUGE-N</p> $\frac{\text{number of overlapping n-grams}}{\text{n-grams in reference summary}} = \frac{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} \text{Count}_{\text{match}}(gram_n)}{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} \text{Count}(gram_n)}$
<ul style="list-style-type: none"> <li>- 주로 Machine Translation 평가를 위해 사용</li> <li>- Precision 기반</li> <li>- Brevity Penalty를 줌</li> </ul>	<ul style="list-style-type: none"> <li>- 주로 Summarization 평가를 위해 사용</li> <li>- Recall 기반</li> <li>- Brevity Penalty를 주지 않음</li> </ul>

둘 다 얼마나 맞았는지에 대해 관심이 있으나 바라보고자 하는 관점만 다른 것!

## Unit 02 | Neural Summarization

### Pre-neural summarization

이전의 요약 시스템은 대부분 **추출(extractive)** 방식



**Figure 23.14** The basic architecture of a generic single document summarizer.

① Content Selection: 포함할 **중요 문장**을 선택함

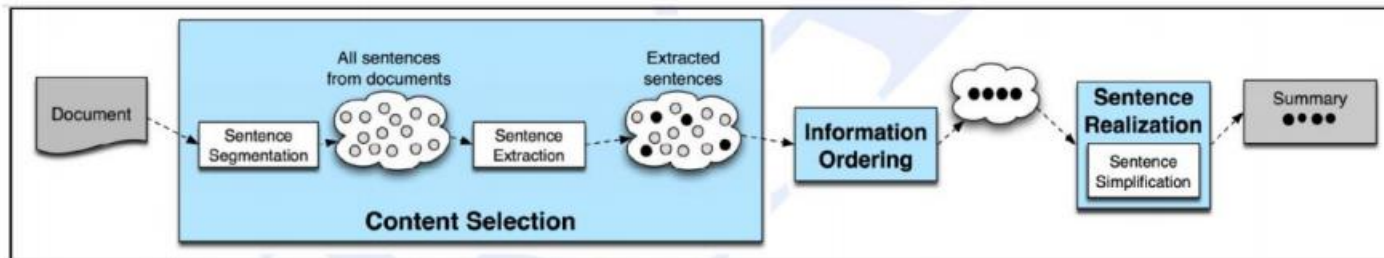
-> topic이 해당 문장에 존재하는지

-> 문장의 위치

## Unit 02 | Neural Summarization

### Pre-neural summarization

이전의 요약 시스템은 대부분 **추출(extractive)** 방식



**Figure 23.14** The basic architecture of a generic single document summarizer.

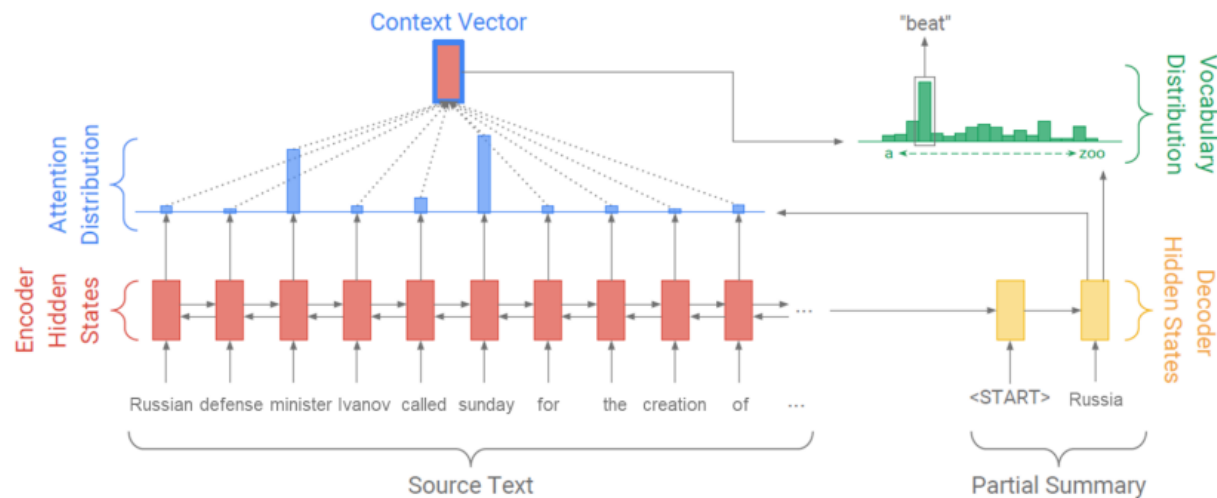
- ① Content Selection: 포함할 중요 문장을 선택함
- ② Information Ordering: 이 문장들을 중요도에 따라 순서대로 나열
- ③ Sentence Realization: 요약문 만들기

## Unit 02 | Neural Summarization

### Neural Summarization

single-document abstractive summarization task로 보고, seq2seq으로 풀어봐도 좋지 않을까?

seq2seq + attention 을 적용해보자!



## Unit 02 | Neural Summarization

seq2seq + attention

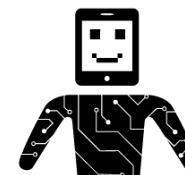


안녕 난 투빅이야

안녕 반가워 투빅아



안녕 반가워 (이름은 모름)



(detail이 떨어짐)

### [문제점]

1. 문장을 생성할 때 out-of-vocabulary 문제
2. 고유명사들의 출력 확률이 낮아지는 문제

# Contents

---

Unit 01 | Decoding Algorithms

---

Unit 02 | Neural Summarization

---

**Unit 03 | Copy Mechanisms**

---

Unit 04 | NLG using unpaired corpus

---

## Unit 03 | Copy Mechanisms

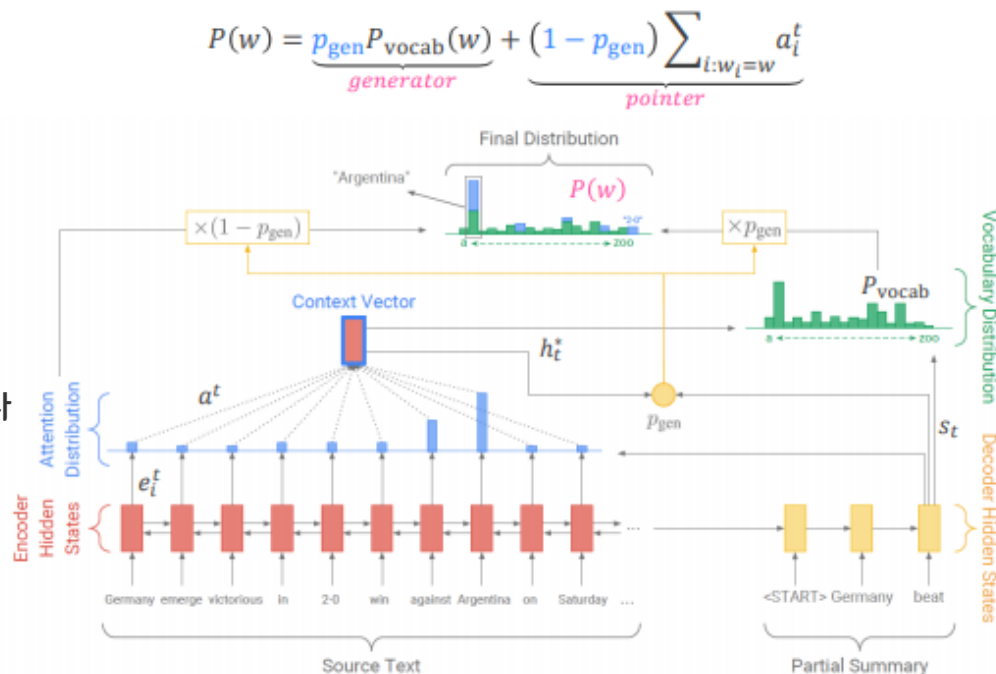
### Copy Mechanisms

Copy Mechanism을 seq2seq와 함께 사용해서 디테일을 잡아내자!  
Attention에서 copy를 더 잘할 수 있도록 처리함

$$p_{\text{gen}} = \sigma \left( \underbrace{w_h^T h_t^*}_{\text{context}} + \underbrace{w_s^T s_t}_{\text{dec state}} + \underbrace{w_x^T x_t}_{\text{dec input}} + b_{\text{ptr}} \right)$$

(단어를 copy할 지 생성할 지 결정지어주는 것)

(입력된 단어 들로부터 얼마나  
copy 할지에 대한 분포)





## Unit 03 | Copy Mechanisms

기존의 pre-neural summarization은 content selection과 surface realization으로 나누어져 동작함

중요 문장을 선택하는 부분

요약을 하는 부분

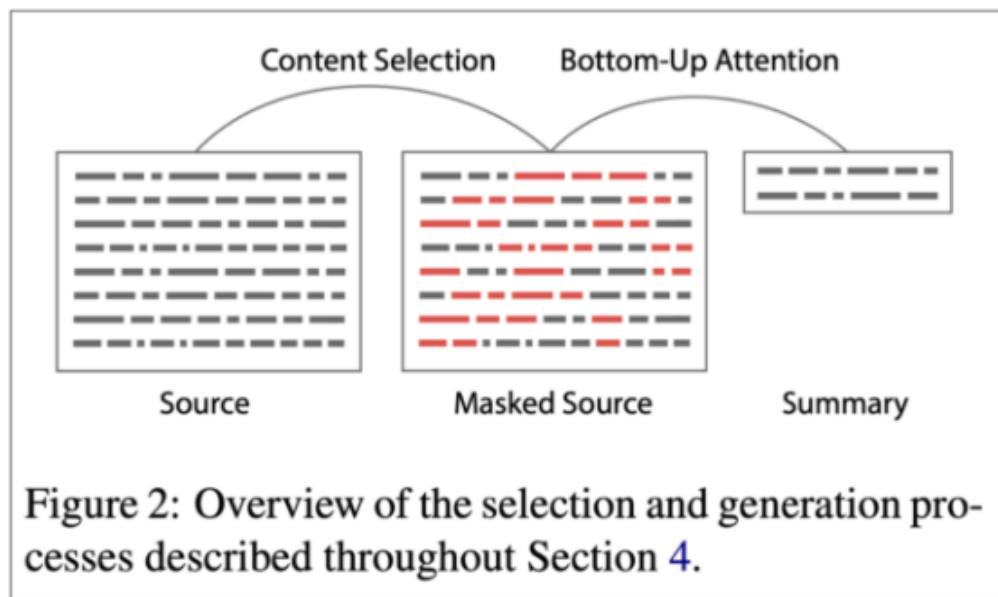
하지만 neural approach는 그런 것없이 하나로 묶여서 나오기 때문에 전체적인 것을 보지 못하는 문제가 발생함

## Unit 03 | Copy Mechanisms

<https://arxiv.org/pdf/1808.10792.pdf>

### Bottom-up summarization

이러한 점을 보완하기 위해 등장한 방법이 Bottom-up !



Word가 포함되었는지 포함되지 않았는지에 따라 0과 1을 태깅  
모델은 word가 포함되지 않은 부분에는 집중하지 않음

간단하지만 매우 효과적인 방법!

# Contents

---

Unit 01 | Decoding Algorithms

---

Unit 02 | Neural Summarization

---

Unit 03 | Copy Mechanisms

---

Unit 04 | NLG using unpaired corpus

---

## Unit 04 | NLG using unpaired corpus

지금까지는 input에 대응하는 output을 미리 준비한 후에 학습시키는 지도 학습에 기반하고 있음  
예를 들어 '오늘 날씨 알려줘'에 대응한 출력 문장인 '오늘은 비가 올 것 같아요'와 같은 쌍을 준비하여 학습함

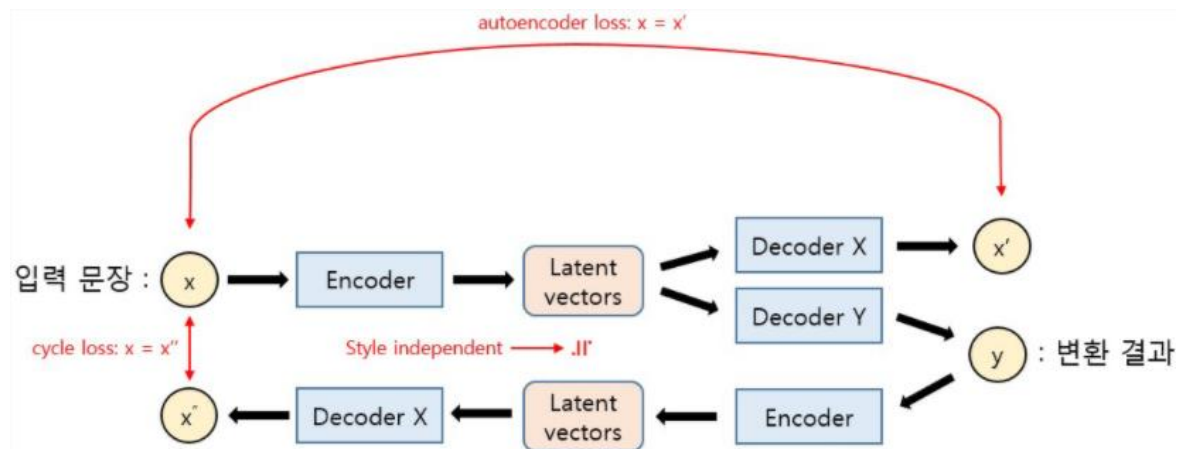
이러한 방법의 문제는 입력-출력 쌍의 데이터를 대량으로 요구한다는 것  
고성능의 자연어 생성 시스템을 만들기 위해 그 훈련에 필요한 말뭉치의 확보부터 현실적으로 매우 어려움

이러한 한계점을 돌파하기 위해 Unpaired corpus를 활용한 비지도 학습 등장

## Unit 04 | NLG using unpaired corpus

이는 비교적 최근 시작된 연구 방향이기에 더욱 고난이도의 기술을 요구함

자세한 설명은 어려우니...ㅎㅎ 인공지능망이 반드시 학습해야 하는 핵심적인 포인트들을 정리해보자 !

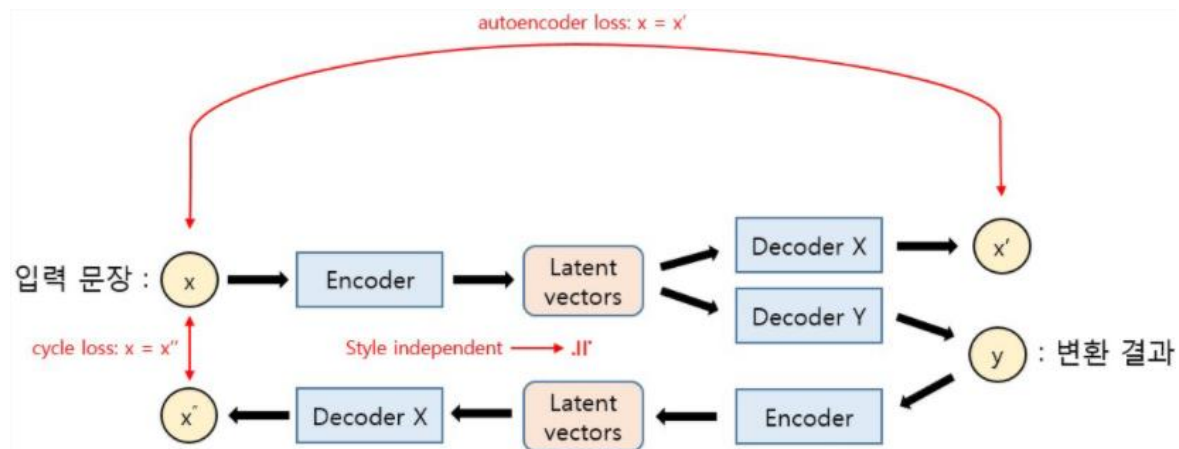


1) 어떤 스타일의 문장이 들어오더라도 그 **스타일에 의존적이지 않는** 본질적인 의미를 Latent vectors의 형태로 인코딩해야 한다.

2) 인코딩된 Latent vectors가 주어졌을 때, **각 스타일에 해당되는** 디코더는 해당 스타일의 문장을 생성할 수 있어야 한다.

## Unit 04 | NLG using unpaired corpus

인공 신경망은 Autoencoder loss, Cycle loss를 최소화하는 방향으로 학습



### Autoencoder loss

: X 스타일의 문장  $x$ 를 latent vectors로 변환한 후, 이를 다시 X 디코더를 이용해 문장  $x'$ 가 생성되었을 때,  $x'$ 과 원래의 문장  $x$ 와 얼마나 다른지

### Cycle loss

: X 스타일 문장  $x$ 를 변환 과정을 통해 Y 스타일의 문장  $y$ 로 변환하였을 때, 이 문장  $y$ 를 다시 X 스타일로 변환한 문장  $x''$ 과 원래의 문장  $x$ 는 얼마나 다른지

## Unit 04 | NLG using unpaired corpus

### Unpaired corpus를 적용한 예시

#### 부정적 <-> 긍정적

##### 부정적

1486. avoid it if you can .  
1487. worst management & staff imaginable and all follows from that .  
1488. what kind of mcdonalds does n't have that .  
1489. i would give then zero stars if i could !  
1490. worst mcdonalds ever , horrible service .  
1491. dirty and rude employees .  
1492. do not recommend !  
1493. worst mcdonald 's ever .

##### 긍정적

1486. love it if you can .  
1487. best management & staff , and all follows from that .  
1488. what kind of mcdonalds does n't have that .  
1489. i would give then five stars if i could !  
1490. best mcdonalds ever , great service .  
1491. nice and friendly employees .  
1492. do definitely recommend !  
1493. best mcdonald 's ever .

#### 구어체 <-> 문어체

##### 구어체

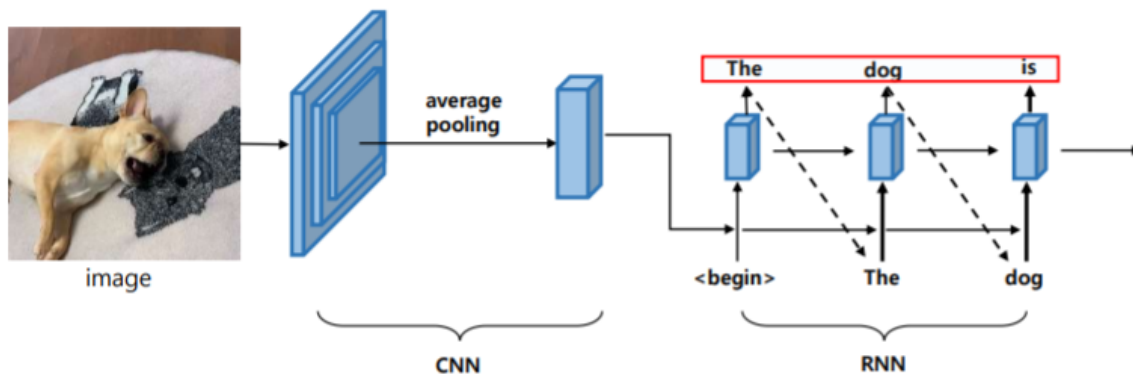
24. 롯데 타선이 무려 24안타를 터뜨리며 뜨거운 화력을 뽐냈다.  
25. NC 다이노스는 11일 잠실구장에서 열린 2014 한국야구리그 세븐 프로야구 LG 트윈스와의 시즌 1차전에서 12-11로 승리했다.  
26. 마무리 투수 손승락은 팀의 1점차 승리를 지켜내며 세이브를 챙겼다.  
27. NC 선발 이재학은 7.2이닝 1실점 호투로 시즌 첫 승을 거뒀다.  
28. SK는 이날 승리로 9승 3패를 기록하며 선두 자리를 굳게 지켰다.  
29. 지난해 KIA전 10승 6패로 무위를 점했던 롯데는 올해도 그 무세를 이어갔다.  
30. 하지만 다음 이닝에 팀이 2점을 몰려 역전하면서 구원승을 거뒀다.  
31. 한화는 3연패에 빠졌다.  
32. 반면 4연패에 빠진 한화는 시즌 10패(4승)를 당했다.

##### 문어체

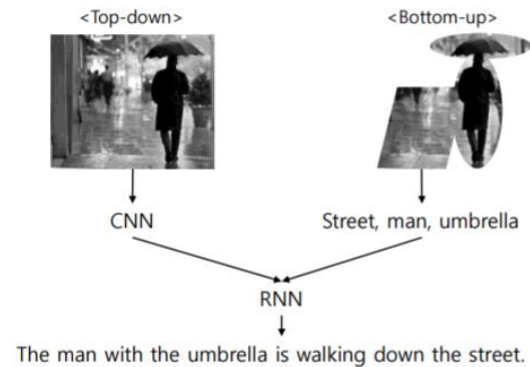
24. 롯데 타선이 무려 24안타를 터뜨리면서 뜨거운 화력을 뽐냈습니다.  
25. NC가 LG와의 경기에서 12-11로 승리했습니다.  
26. 마무리 투수 손승락은 팀의 1점차 승리를 지켜내면서 세이브를 챙겼어요.  
27. NC 선발 이재학은 7.2이닝 1실점 호투로 시즌 첫 승을 거뒀습니다.  
28. SK는 이날 승리로 9승 3패를 기록하면서 선두 자리를 굳게 지켰어요.  
29. 지난해 KIA전 10승 6패로 무위를 점했던 롯데는 올해도 그 무세를 이어갔어요.  
30. 하지만 다음 이닝에 팀이 2점을 몰려서 역전하면서 구원승을 거뒀어요.  
31. 한화는 3연패에 빠졌어요.  
32. 반면 4연패에 빠진 한화는 시즌 10패(4승)를 당했어요.

## Image Captioning

이미지를 설명하는 문장을 생성하는 알고리즘  
CNN과 RNN이 결합된 구조



Top-down approach: 이미지의 전체적인 특징을 확인  
Bottom-up approach: 이미지의 자세한 부분을 확인



- 1) CNN을 통해 이미지의 전체적인 feature를 추출
- 2) 이미지로부터 Bottom-up 방식을 통해 Attribute 추출  
이때 추출 방식은 KNN을 이용
- 3) RNN을 통해 문장의 각 단어를 순서대로 출력



## 출처

- CS224n Lecture 15 강의 & 강의 자료
- 기타 블로그들
  - <https://velog.io/@nawnoes/%EC%9E%90%EC%97%B0%EC%96%B4%EC%B2%98%EB%A6%AC-Beam-Search>
  - <https://jeongukjae.github.io/posts/cs224n-lecture-15-natural-language-generation/>
  - <https://www.slideshare.net/ThoPhan27/abstractive-text-summarization>