



Week 18_Lecture 15

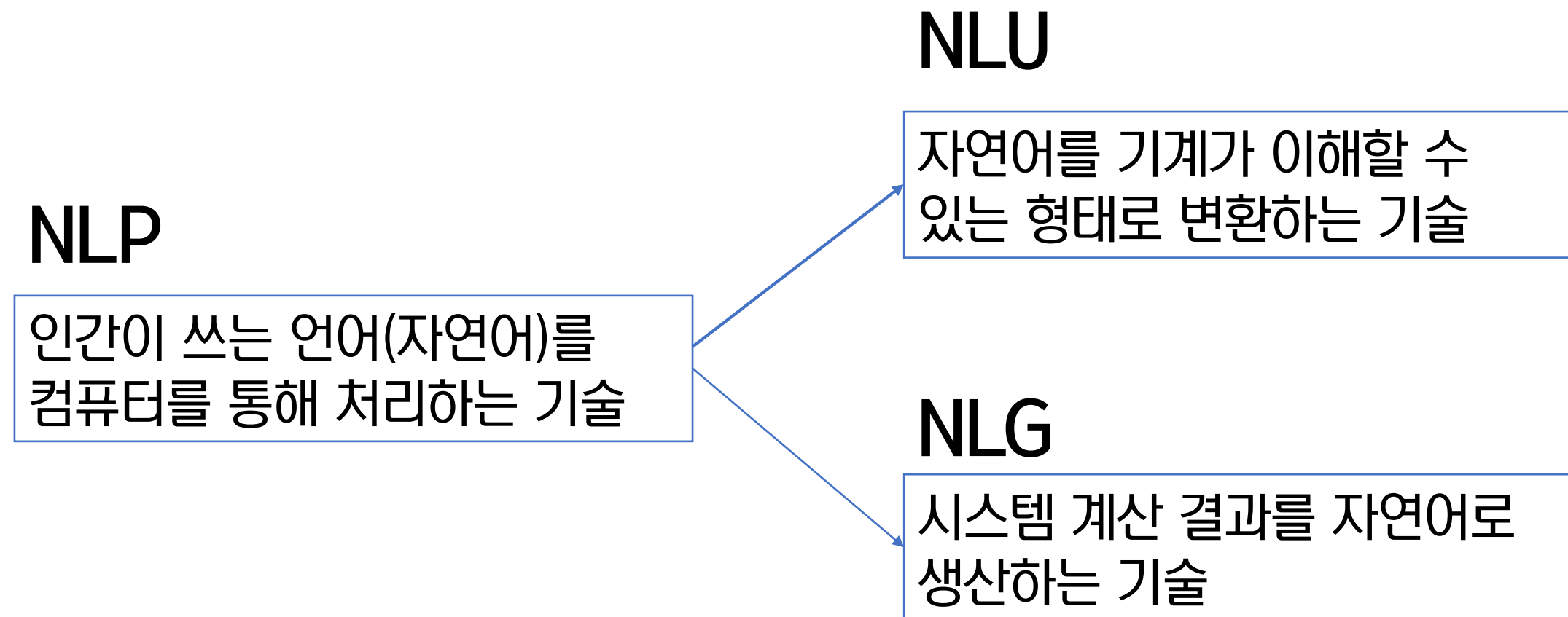
Natural Language Generation

김나현, 황채원

LMs and decoding algorithms



Natural Language Generation (NLG)



Natural Language Generation (NLG)

주어진 input text에 대해 새로운 text를 생성하는 작업

- Machine Translation
- Summarization
- Dialogue(chit-chat)
- Creative Writing : storytelling, poetry-generation
- Freeform Question Answering (not extracted)
- Image Captioning

Natural Language Generation (NLG)

무엇이 적합한 NLP인가?

- **적절성**: 생성된 문장이 모호하지 않고 원래의 input text의 의미와 일치해야 한다.
- **유창성**: 문법이 정확하며 어휘를 적절하게 사용해야 한다.
- **가독성**: 적절한 지시어, 접속사 등을 사용하여 문장의 논리 관계를 고려하여 생성해야 한다.
- **다양성**: 상황과 대상에 따라 다른 문장을 생성해야 한다.

Language Modeling

Recap

- Language Modeling : 주어진 단어로 다음 단어를 예측하며, 이에 대한 확률 분포를 생성하는 시스템

$$P(y_t | y_1, \dots, y_{t-1})$$

- Conditional Language Modeling : input x 를 추가로 고려하여 다음 단어를 예측하는 시스템

$$P(y_t | y_1, \dots, y_{t-1}, x)$$

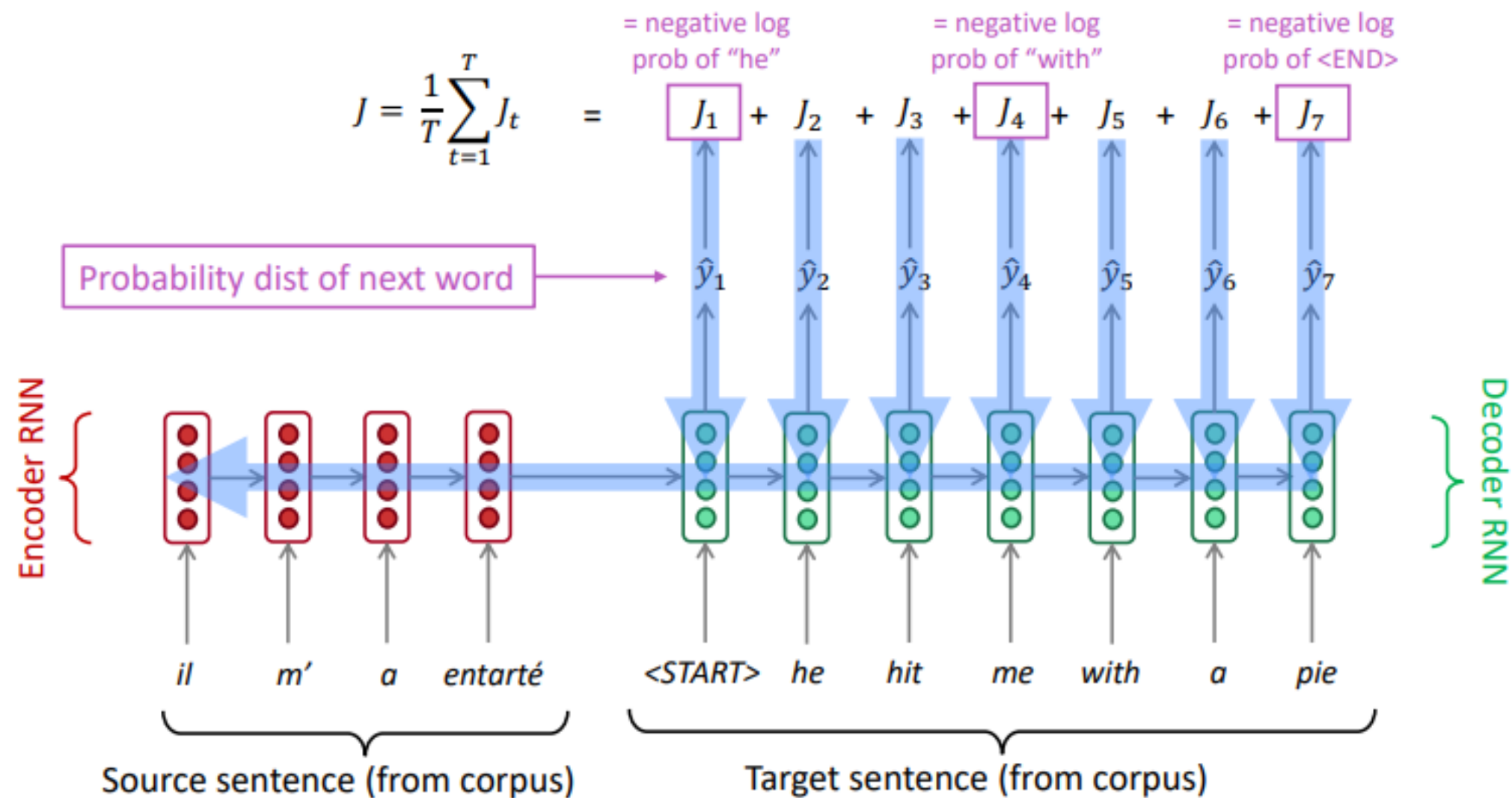
Examples of conditional language modeling tasks:

- Machine Translation (x =source sentence, y =target sentence)
- Summarization (x =input text, y =summarized text)
- Dialogue (x =dialogue history, y =next utterance)
- ...

Language Modeling

Recap : training a (conditional) RNN-LM

This example: Neural Machine Translation



- Teacher Forcing : 시퀀스 생성과정에서 시퀀스 정보를 계속하여 인풋으로 넣어주는 학습방법

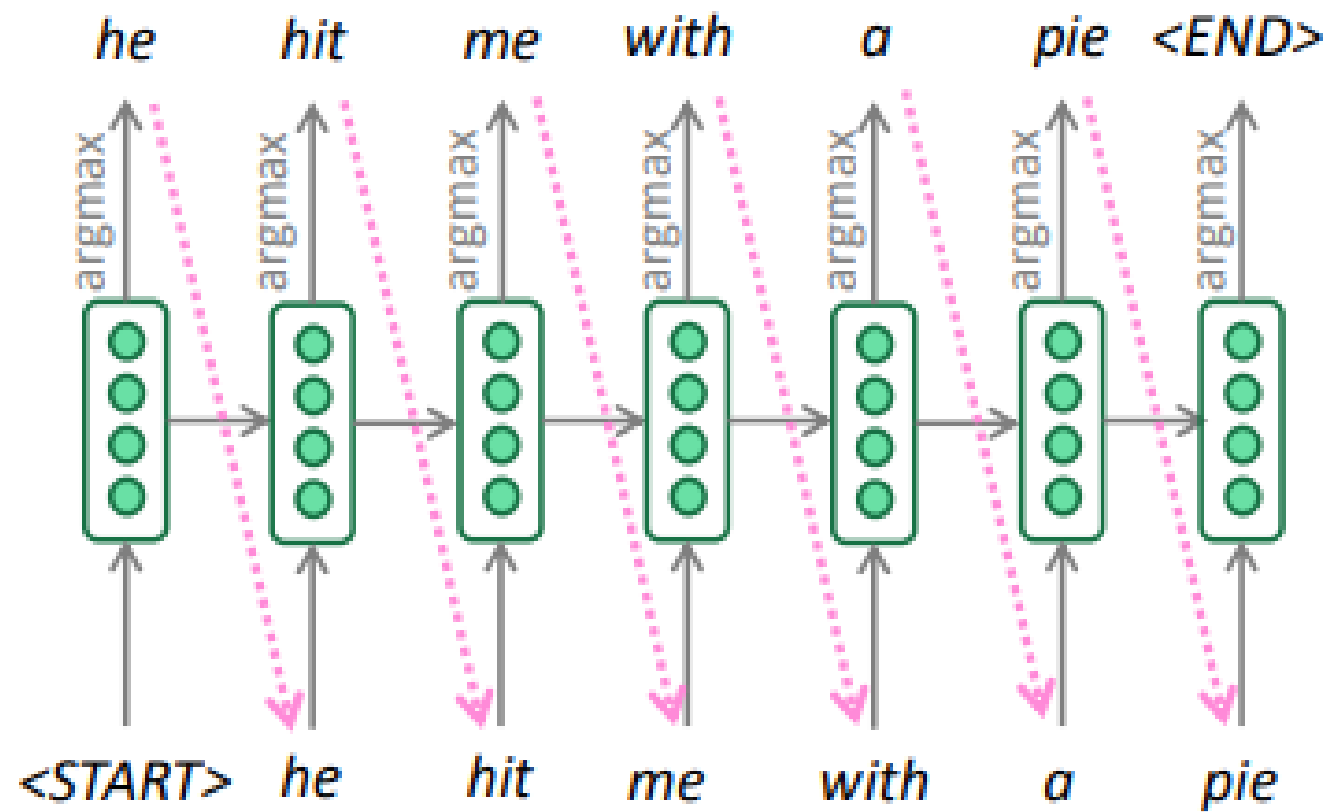
Decoding Algorithm

Decoding Algorithm을 이용한 NLG

- Greedy Decoding
- Beam Search

Decoding Algorithm

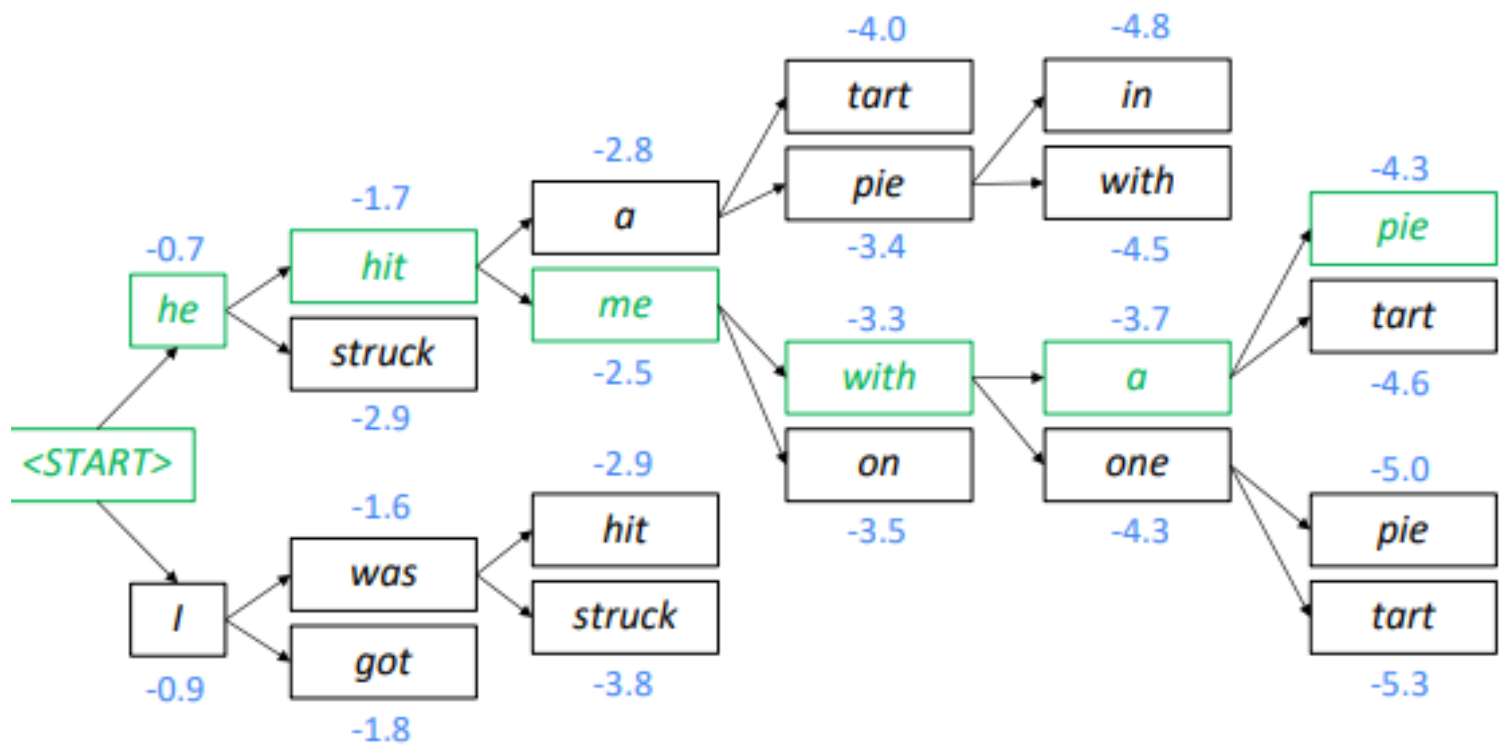
1. Greedy Decoding



- 각 단계에서 most probable한 단어를 선택
- 선택한 단어를 다음 단계의 input으로 feed
- 중간에 잘못된 예측을 할 경우 다음 스텝에 연쇄적으로 영향이 가기 때문에 최종 아웃풋이 좋지 않을 수 있다.

Decoding Algorithm

2. Beam Search



- K개의 후보들 중 가장 높은 확률을 갖는 것을 선택하는 방식
- K=1인 경우 greedy decoding과 같다.

Effect of beam size in chitchat dialogue

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



Human chit-chat partner

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

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

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

- Small k : greedy decoding과 같은 단점을 가진다.
- Larger k : computationally expensive

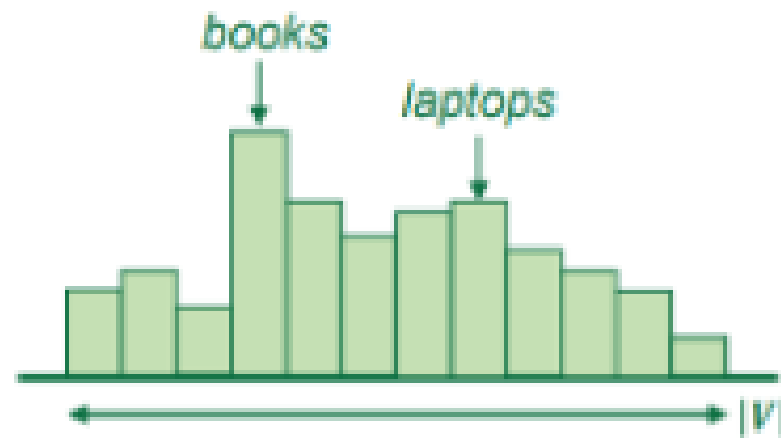
low BLEU score

일반적이고 짧은 답변(=다양성 낮음)

Decoding Algorithm

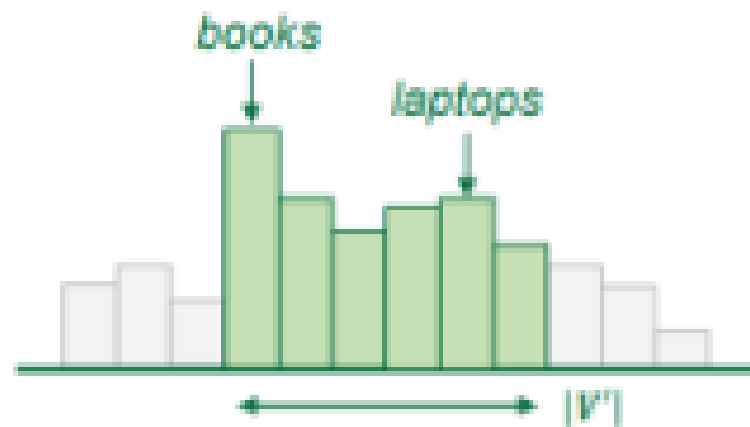
3. Sampling-based decoding

1) Pure Sampling



- Greedy Decoding과 비슷하지만 각 단계에서 random sample을 선정한다.

2) Top-n Sampling



- Random sample 을 뽑지만, 상위 n개 중에서 제한하여 뽑는다.
- Increase n : diverse/risky output
- Decrease n : generic/safe output

Decoding Algorithm

Softmax Temperature (decoding algorithm은 아니지만 함께 사용됨)

$$P_t(w) = \frac{\exp(s_w)}{\sum_{w' \in V} \exp(s_{w'})} \quad \Rightarrow \quad P_t(w) = \frac{\exp(s_w/\tau)}{\sum_{w' \in V} \exp(s_{w'}/\tau)}$$

Raise the temperature τ : P_t becomes more uniform

- Thus more diverse output (probability is spread around vocab)

Lower the temperature τ : P_t becomes more spiky

- Thus less diverse output (probability is concentrated on top words)

- Temperature 증가 : probability distribution이 균등해진다. -> 다양한 output
- Temperature 감소 : probability distribution이 몇 개의 단어에만 집중된다.

NLG tasks and neural approaches to them



Summarization

Task definition

Input text x 가 주어졌을 때, x 의 정보를 포함하며 그것보다 짧은 summary y 를 작성하라.

- Single document : 하나의 문서 x 에 대한 summary y 를 작성한다.
- Multi document : 여러 개의 문서 x 에 대한 summary y 를 작성한다.

일반적으로 x 들에게는 공통의 부분이 존재한다.

Summarization

Main strategies

Extractive summarization

- 원본 텍스트의 일부를 골라내서 summary를 생성한다.
- 생성하기 쉽다.
- Paraphrasing이 없으므로 문장이 제한적이다.

Abstract summarization

- NLG 기술을 이용하여 새로운 텍스트를 생성한다.
- 더 어려운 방법
- 자연어에 더 가까운 방법, 유연한 문장 구사

Summarization

Pre-neural summarization

- 대부분 extractive 방식을 취함
- Pipeline :
 - Content selection – 문장을 선택한다.
 - Information ordering – 문장의 순서를 정한다.
 - Sentence realization – 문장의 시퀀스를 편집한다.

Summarization

Summarization evaluation: ROUGE

- BLEU처럼 n-gram기반의 평가방식이다.
- Recall에 기반한다.
- BLEU는 Machine Translation 평가에 사용하지만, ROUGE는 Summarization에 사용한다.
- BLEU와 달리 Brevity Penalty를 주지 않는다.

*Brevity Penalty : 번역 시 짧은 문장이 높은 정밀도를 갖는 것을 방지하기 위해 짧은 문장에 부여하는 패널티

$$\text{ROUGE-N} = \frac{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} \text{Count}_{match}(gram_n)}{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} \text{Count}(gram_n)} \quad (1)$$

Summarization

Neural Summarization : copy mechanisms

Seq2seq + attention system -> 결과물은 유창했지만 디테일에 약하다 -> copy mechanism

- extractive/abstractive approach의 하이브리드 방식
- 문제점) 너무 많이 copy하게 되는 경우 발생
- 요약할 텍스트가 길 때 overall content selection에 취약하다.

Summarization

Neural Summarization : Bottom-up Summarization

- Content Selection :

sequence-tagging 모델을 활용하여 각 단어들마다 요약에 사용할지, 사용하지 않을지 이진 태깅을 한다.

- Bottom-up Attention :

태깅을 통해 요약에 사용할 단어들만을 이용하여 실제 요약문을 생성하는 인코더-디코더에 넣게 된다.

Dialogue

Pre-neural Dialogue

: 주로 predefined response template 사용하여 반응하거나, 적절한 response 를 검색(retrieve)하여 뽑아 쓰는 방식

이후 seq2seq 기반 모델들을 사용하는 대화시스템이 연구되어왔고, open-ended한 반응을 생성하는 쪽으로 발전됨.

- **Task-oriented dialogue**
 - **Assistive** (e.g. customer service, giving recommendations, question answering, helping user accomplish a task like buying or booking something)
 - **Co-operative** (two agents solve a task together through dialogue)
 - **Adversarial** (two agents compete in a task through dialogue)
- **Social dialogue** (open-domain dialogue)
 - **Chit-chat** (for fun or company)
 - **Therapy** / mental wellbeing

Dialogue

seq2seq models → end-to-end method

- 자연어 이해 모듈에서는 질문의 도메인과 의도 등 파악
- 대화 관리 모듈에서는 대화 맥락을 고려하여 질문에 대한 답변 탐색.
- 자연어 생성 모듈에서는 앞 단계에서 찾은 정보로 문장 구성.

☞ 딥러닝을 이용하여 각 모듈을 구현하거나 모듈 간의 경계 없이 대화를 end-to-end로 처리

[End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning](#) (2016)

[A Network-based End-to-End Trainable Task-oriented Dialogue System](#) (2017)

[End-to-end task completion neural dialogue systems](#) (2017)

Storytelling

- ☞ 이미지가 주어졌을 때 스토리 생성
- ☞ 짧은 키워드들이 주어졌을 때 스토리 생성
- ☞ 현재 시점까지의 스토리가 주어졌을 때, 앞으로의 스토리 생성

학습 가능한 parallel data 부족 → common sentence-encoding space 사용

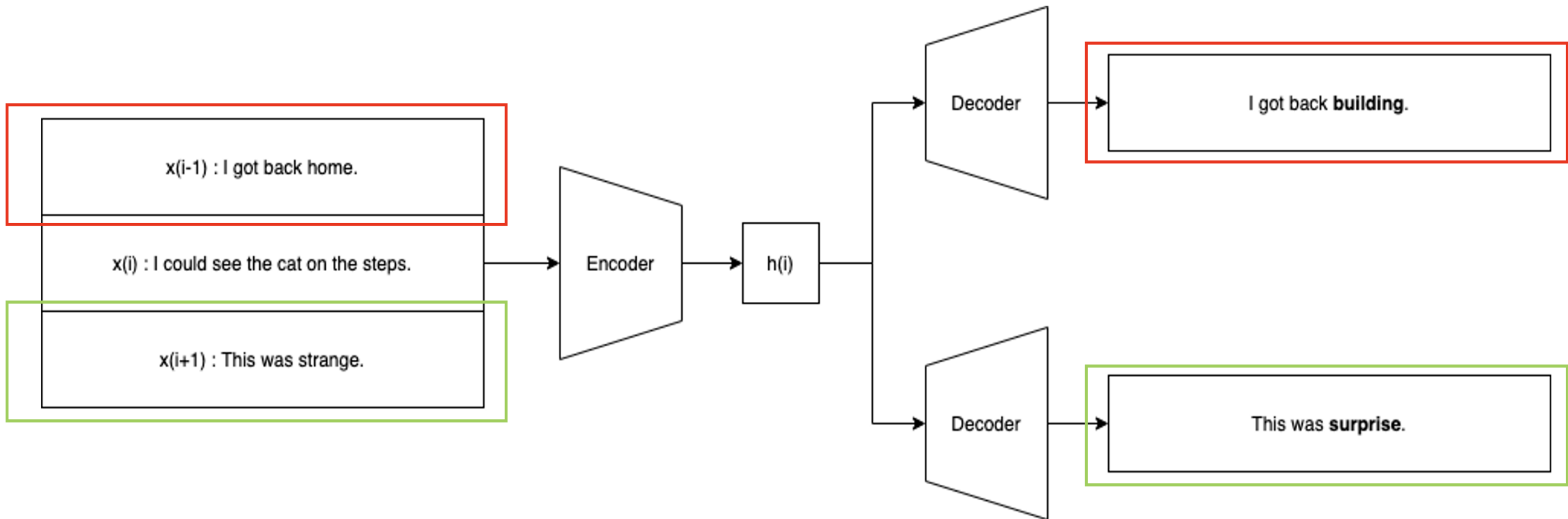
1. skip-thought vector라는 문장 임베딩 방법론을 이용해 sentence representation 학습
2. skip-thought vector를 입력받아 text의 시퀀스로 decoding 하는 언어모델을 학습시킴.
3. 두 components 결합

Storytelling

<https://velog.io/@xuio/NLP-%EB%85%BC%EB%AC%B8%EB%A6%AC%EB%B7%B0-Skip-Thought-Vectors>

skip-thought vector

- 문장을 임베딩,
일반적인 encoder 만들고자 함



NLG Evaluation



Automatic evaluation metrics for NLG

- Machine translation task에서 이 측도들은 ideal하지 않으며,
- Summarization task에서는 심지어 더 worse하며,
- Dialogue task에서는 더욱 worse하다.

(open-ended 한 task 일수록 word overlap기반 측도들은 그 평가에 타당성을 잃어간다.)

- perplexity와 word embedding는??

perplexity는 generated text의 quality에 대한 정보는 평가할 수 없으며,

모델에 의해 생성된 문장의 embedding과 사람이 작성한 정답문장의 embedding 벡터들을 비교하는 방법은 결과적으로 human judgment score와 상관성이 전혀 없었다...

Automatic evaluation metrics for NLG

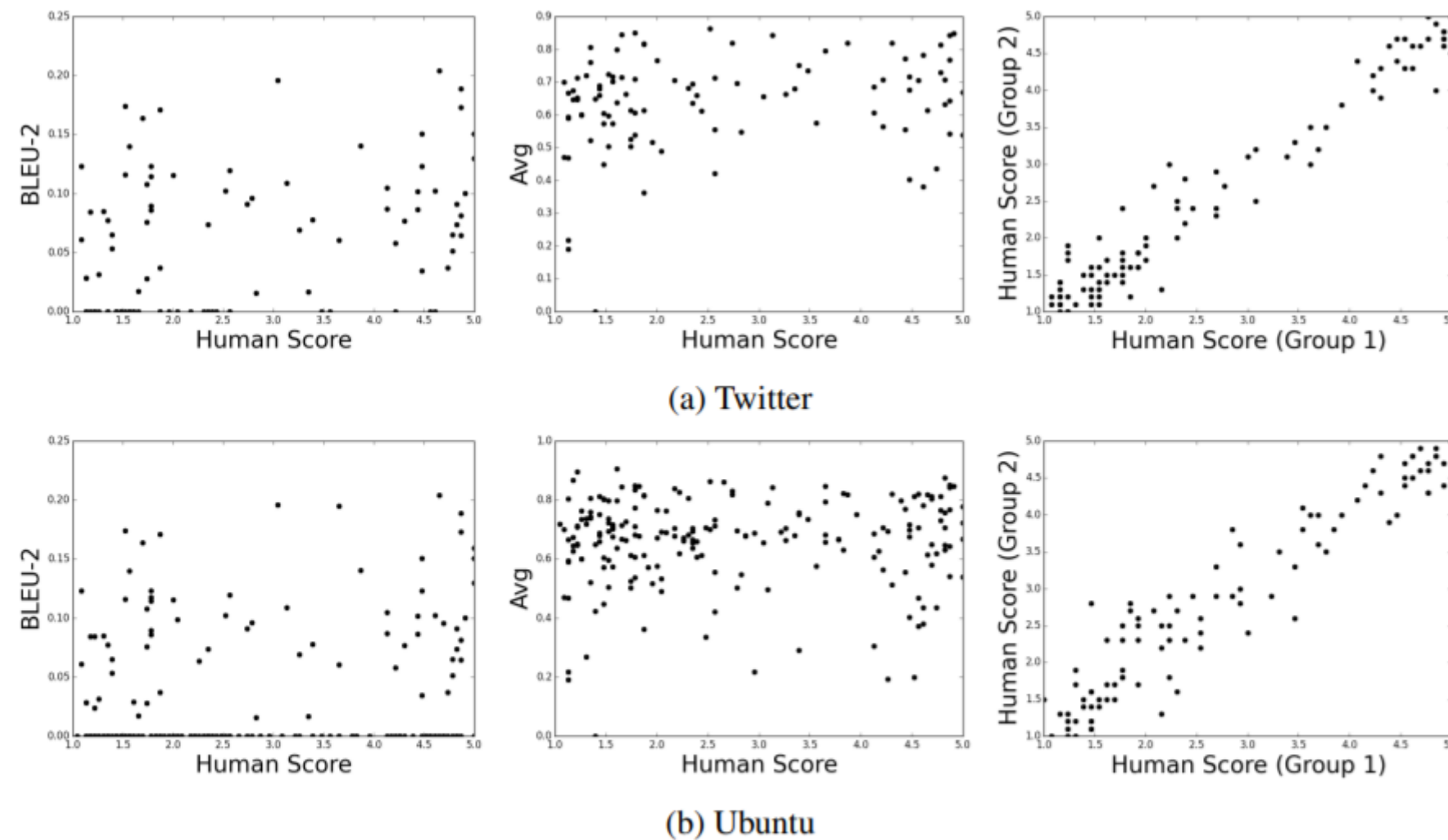


Figure 1: Scatter plots showing the correlation between metrics and human judgements on the Twitter corpus (a) and Ubuntu Dialogue Corpus (b). The plots represent BLEU-2 (left), embedding average (center), and correlation between two randomly selected halves of human respondents (right).

문장 생성의 전반적인 품질(overall quality)을 평가할 수 있는 자동화 standard는 현재까지 없다.

그러나 전반적인 품질이 아닌 조금 더 구체적인 기준을 설정하여 특정한 관점에서 생성 문장을 평가할 수 있는 standard들을 정의할 수 있다.

- 유창성(fluency)
- 다양성(diversity)
- input과의 관련성
- 문장길이 혹은 반복정도
- Task-specific metrics (텍스트 요약 task에서의 압축률 등)

Detailed Human eval of controllable chatbots

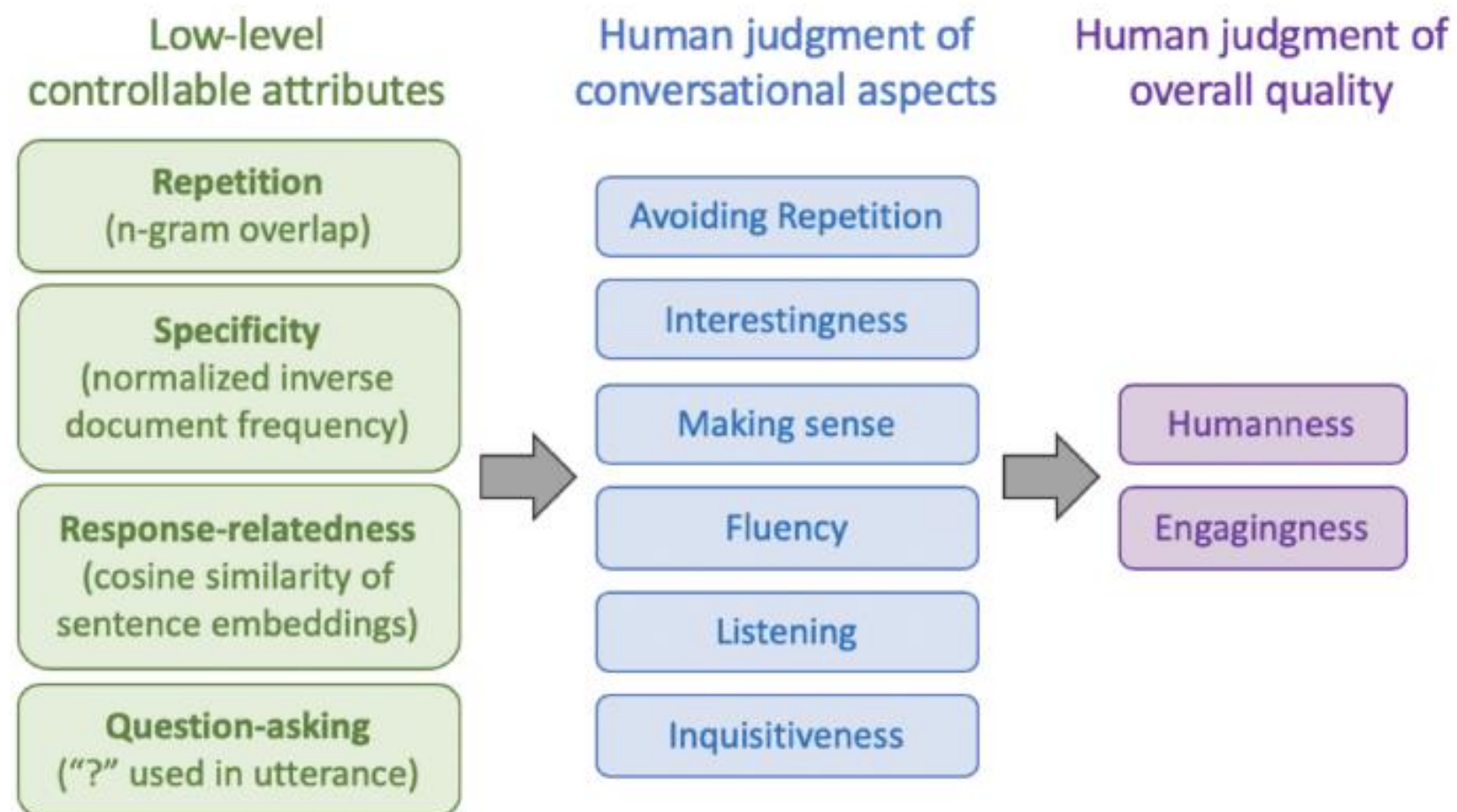
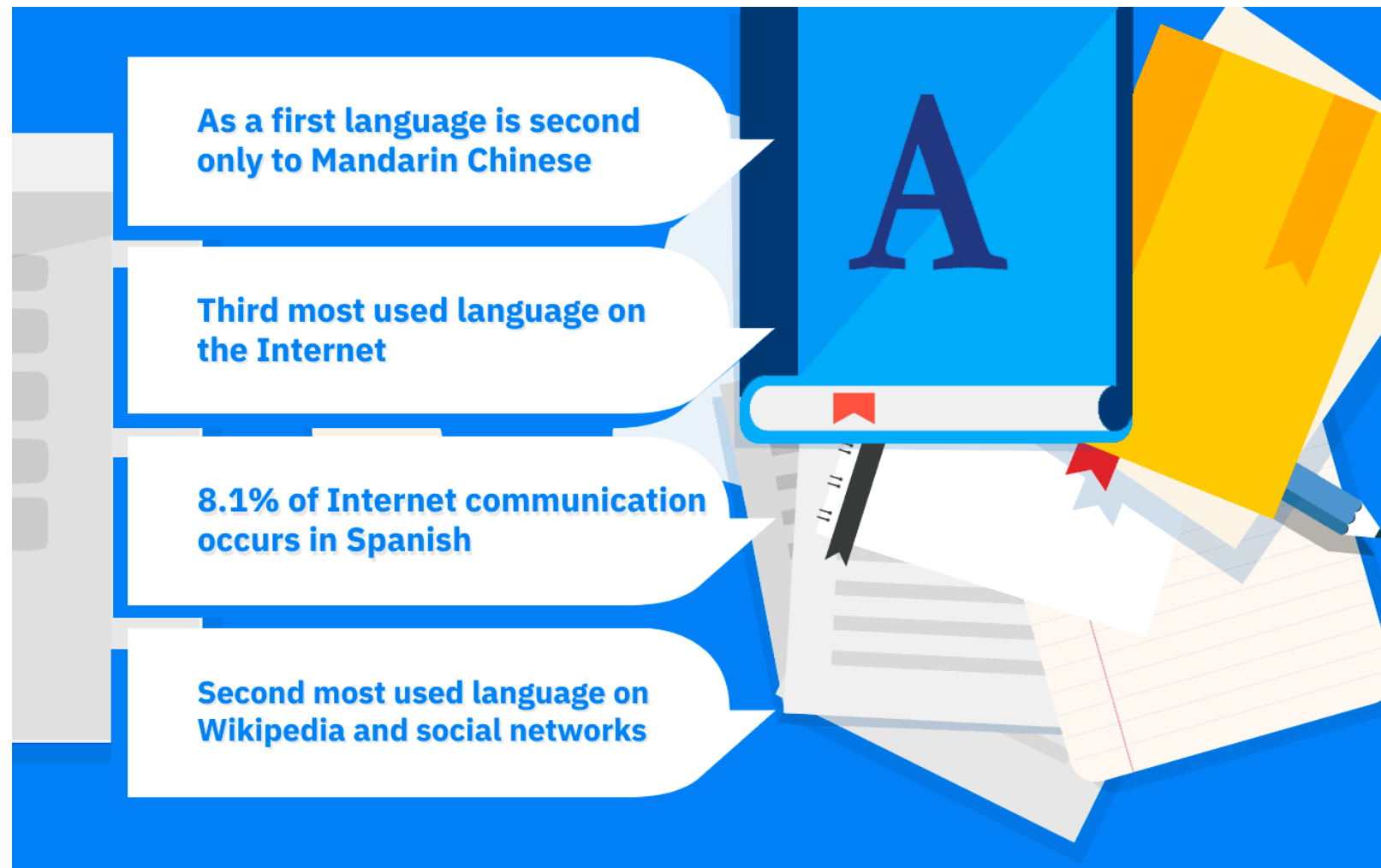


Figure 1: We manipulate four low-level attributes and measure their effect on human judgments of individual conversational aspects, as well as overall quality.

NLP / NLG trend

Polyglot artificial intelligence models

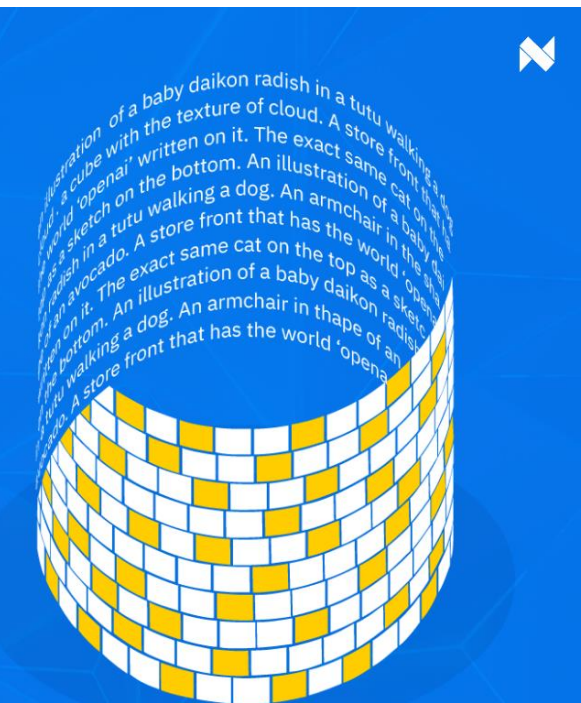


What is a transformer in NLP?

3 MINS

DALL-E

CREATING IMAGES FROM TEXT



PORORO: Platform Of neuRal mOdelS for natuRal language prOcessing

이미지 캡셔닝

OCR

기계번역

감정분석

자연어 추론

zero-shot topic 분류

텍스트 요약

- Abtractive summarization : Model generate a summary in the form of a complete sentence.
- Bullet-point summarization : Model generate multiple summaries in the form of a short phrase.
- Extractive summarization : Model extract 3 important sentences from article.

PORORO: Platform Of neuRal mOdelS for natuRal language prOcessing

먼저 **abstractive** 모델은 하나의 완전한 문장으로 텍스트 내용을 요약합니다. 이 모델은 SKT에서 개발한 KoBART(<https://github.com/SKT-AI/KoBART>) 모델을 사용합니다. 학습에 사용한 데이터는 데이콘(DACON)의 문서 추출요약 경진 대회 데이터(<https://dacon.io/competitions/official/235671/data/>)와 AI 허브에서 공개한 AI 학습용 데이터(<https://www.aihub.or.kr/node/9176>)입니다.

task='summary' 로 지정하여 텍스트 요약 작업을 알리고 **lang='ko'** 로 지정합니다. 현재 텍스트 요약 작업은 한글만 지원합니다. 먼저 **abstractive** 모델을 사용해 보죠.

경고: 최신 **transformers** 패키지에서 나눗셈 에러가 발생할 수 있습니다. 이런 경우 다음 명령으로 **transformers** 4.7.0 버전을 설치해 주세요.

```
!pip install -U transformers==4.7.0
```

```
abs_summ = Pororo(task='summary', lang='ko', model='abstractive')
```

```
abs_summ(text1)
```

'미래의서비어런스는 화성의 생명체 거주 여부, 화성의 고대 환경 조사, 화성 지표의 역사 등을 밝히는 것이 목표이며 미래의 인류가 화성을 유인 탐사할 때 위험한 것이 없는지 탐색하고, 대기의 조성을 알려주어 미래의 기지를 건설하는 데 도움을 주는 역할을 수행한다.'

꽤 잘 요약된 것 같습니다. 다음에는 **bullet** 모델을 사용해 보겠습니다. 이 모델은 짧은 몇 개의 문장으로 텍스트를 요약합니다.

```
bul_summ = Pororo(task='summary', lang='ko', model='bullet')
```

```
bul_summ(text1)
```

['NASA 화성 지표면 로봇 탐사 계획', '퍼서비어런스, 인제뉴어티 드론 헬리콥터 포함']

Transformer architectures - encoder

```
text2 = "알로사우루스(라틴어: Allosaurus)는 후기 쥐라기(1억 5600만 년 전 ~ 1억 4500만 년 전)를 대표하는 큰 육식공룡이다. 알로사우루스라는 학명의 어원은 고대 그리스어 (αλ
```

```
bul_summ([text1, text2])
```

```
[['NASA 화성 지표면 로봇 탐사 계획', ' 퍼서비어런스, 인제뉴어티 드론 헬리콥터 포함'],  
 ['짧은 목에 큰 머리, 긴 꼬리, 짧은 앞다리', ' 후기 쥐라기 대표하는 큰 육식공룡']]
```

이번에는 텍스트에서 가장 중요한 3개의 문장을 추출하는 **extractive** 모델을 사용해 보겠습니다. 이 모델은 페이스북의 RoBERTa(<https://ai.facebook.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/>) 모델을 위에서 언급한 말뭉치에서 훈련시켜 사용합니다.

```
ext_summ = Pororo(task='summary', lang='ko', model='extractive')  
ext_summ(text1)
```

'퍼서비어런스(Perseverance)는 화성 탐사차로 2020년 7월 30일 발사하여 2021년 2월 18일 화성에 착륙하였다. 화성의 생명체 거주 여부, 화성의 고대 환경 조사, 화성 지표의 역사 등을 밝히는 것이 이 탐사선의 목표다. 참고로, 마스 2020(Mars 2020)은 퍼서비어런스와 인제뉴어티 드론 헬리콥터를 포함한, NASA의 화성 지표면 로봇 탐사 계획의 명칭이다.'

마찬가지로 여러 개의 문장을 리스트로 묶어서 전달할 수 있으며 **return_list=True** 로 지정하면 추출한 3개의 문장을 리스트로 만들어 반환합니다.

```
ext_summ([text1, text2], return_list=True)
```

```
[['퍼서비어런스(Perseverance)는 화성 탐사차로 2020년 7월 30일 발사하여 2021년 2월 18일 화성에 착륙하였다.',  
 '화성의 생명체 거주 여부, 화성의 고대 환경 조사, 화성 지표의 역사 등을 밝히는 것이 이 탐사선의 목표다.',  
 '참고로, 마스 2020(Mars 2020)은 퍼서비어런스와 인제뉴어티 드론 헬리콥터를 포함한, NASA의 화성 지표면 로봇 탐사 계획의 명칭이다.'],  
 ['알로사우루스(라틴어: Allosaurus)는 후기 쥐라기(1억 5600만 년 전 ~ 1억 4500만 년 전)를 대표하는 큰 육식공룡이다.',  
 '알로사우루스는 짧은 목에 큰 머리, 긴 꼬리와 짧은 앞다리를 가진 전형적으로 거대한 수각류 공룡이다.',  
 '생존 당시에는 대형 포식자로서 먹이사슬의 최고점에 있었다.']]
```

THANK YOU

