TRAIN AND TEST

[CS224N] Lecture 15
Natural Language Generation

INDEX

1. LMs and decoding algorithms
2. NLG tasks and neural approaches to them
3. NLG evaluation

LMs and decoding algorithms

Topic1: LMs and decoding algorithms

Natural Language Generation (NLG)

: 'new text를 생성'하는 모든 것!

종류)

1) MT (Machine Translation)

2) Summarization

3) Dialogue

4) Creative writing

5) Freeform Question Answering

6) Image captioning

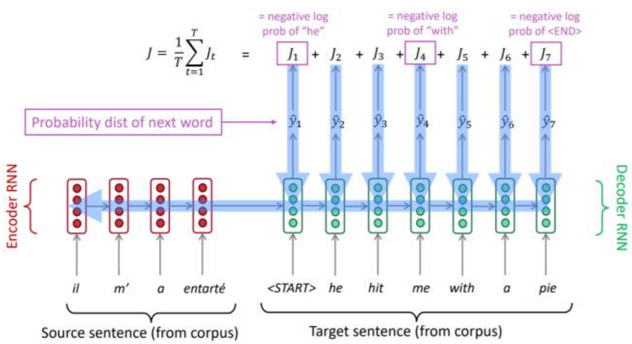
예시)

$$P\left(y_t \middle| y_{1,\ldots,}y_{t-1}\right)$$

: 이전단어들이 주어졌을 때,

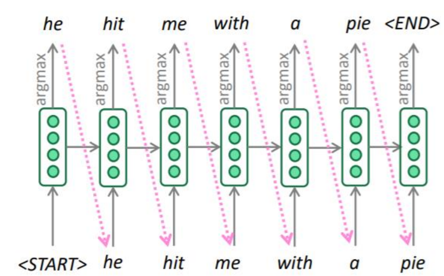
다음 단어를 맞추는 형태 (LM)

Decoding algorithms



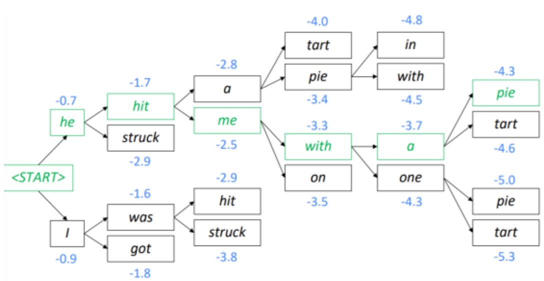
Q) 이러한 모델을 훈련시켰다면 어떻게 사용?

: Decoding algorithms을 사용한다!



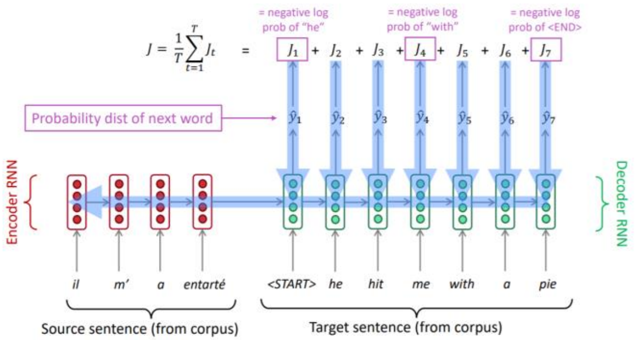
Decoding algorithms의 대표적인 종류

- Greedy decoding: take the most probable word
- Beam Search: find a high-probability sequence



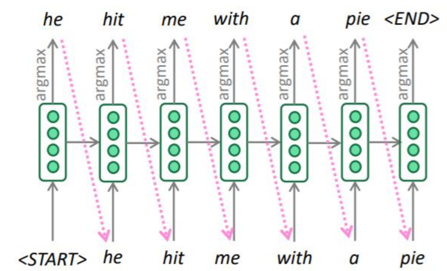
Topic1: LMs and decoding algorithms

Decoding algorithms



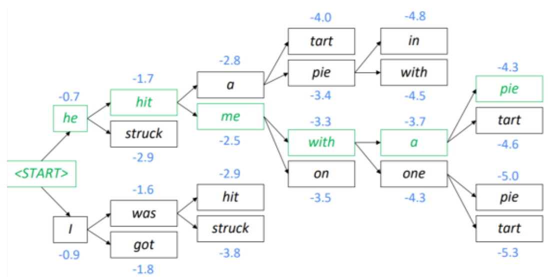
Q) 이러한 모델을 훈련시켰다면 어떻게 사용?

: Decoding algorithms을 사용한다!



Decoding algorithms의 대표적인 종류

- Greedy decoding: take the most probable word
- Beam Search: find a high-probability sequence



Topic1: LMs and decoding algorithms

Beam Search

: k=1인 Beam Search = Greedy decoding

Q) 높은 K는 무조건적으로 좋다?

A1) ㄴㄴ, 연산량이 많아진다는 단점이 있다.

A2) BLEU score를 낮춘다! (짧은 문장을 만드는 경우가 많기 때문에)

A3) 결과가 다소 일반적이다.

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

다른 decoding algorithms

1) Sampling-based decoding

- Pure sampling: 확률분포에서 sampling
- Top-n sampling: 확률분포에서 Top-n만을 sampling
  (n이 커지면 diverse,risky output)
- Softmax temperature: temperature hyperparameter를 적용

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

Param이 커지면 uniform, diverse output

(주의: Softmax temperature 그자체는 not decoding algorithms)

TRAIN AND TEST

NLG tasks and neural approaches

Summarization (요약)

: single-document와 multi-document를 요약하는 작업

- single-document: y|x
- multi-document: $y|x_1,x_2,\cdots x_n$

(주의: $x_1,x_2,\cdots x_n$는 new articles이긴 한데, 같은 주제에 대한 articles)

- Gigaword: first one or two sentences of a news article → headline (aka *sentence compression*)
- LCSTS (Chinese microblogging): paragraph → sentence summary
- NYT, CNN/DailyMail: news article → (multi)sentence summary
- Wikihow: full how-to article → summary sentences
- XSum: (Narayan et al., 2018), Newsroom: (Grusky et al., 2018): article → 1 sentence summary (New datasets!)

* 요약의 두가지 방법

- Extractive summarization

: 핵심문장을 그대로 선택하는 것 (select)

- Abstractive summarization

: 핵심문장을 생성하는 것 (generate)

Pre-Neural Summarization

Q) neural net을 사용하기 이전에는 어떻게 요약을 했을까?

A) Content Selection, Information ordering, Sentence realization

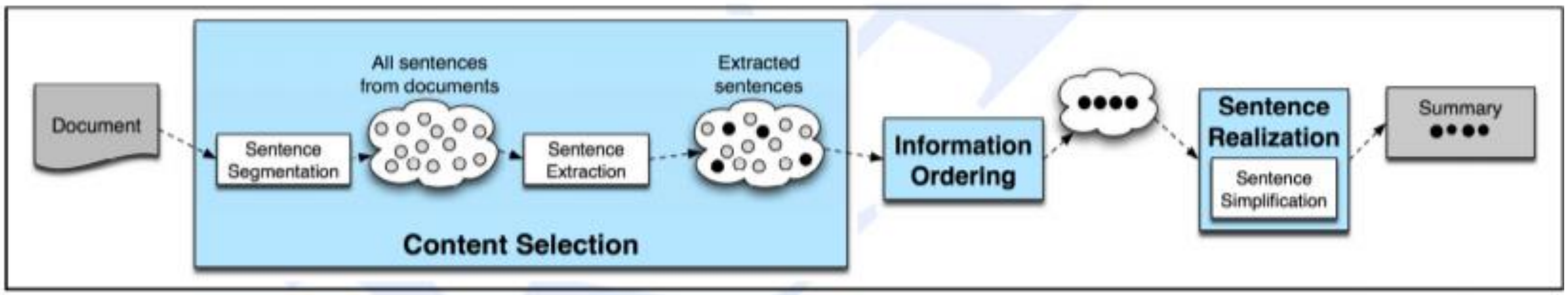


Figure 23.14 The basic architecture of a generic single document summarizer.

* Content Selection Method

- Sentence scoring functions: topic keywords, sentence appearance
- Graph-based algorithms: node & edge

Pre-Neural Summarization

Q) output에 대한 평가는 어떻게 해?

A) ROUGE: scores are reported separately for each n-gram

식)

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

예시) ROUGE는 recall 값을 사용한다.

bigrams 시스템: the cat, cat was, was found, found under, under the, the bed

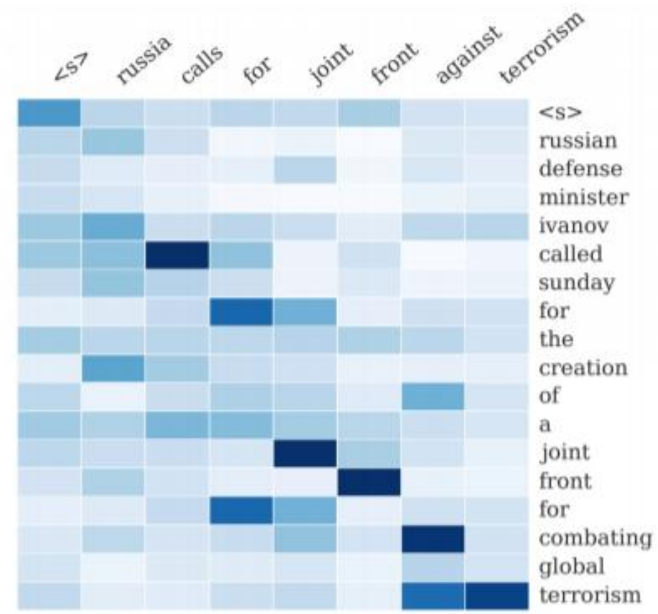
bigrams 참조: the cat, cat was, was under, under the, the bed

$$ROUGE2_{recall} = \frac{4}{5} = 0.8$$

Neural Summarization

: first seq2seq summarization paper

Single-document abstractive summarization is a translation task!



현재는 다양한 발전들이 있었다!

ex) multi-level attention, RL, Graph-algorithms,

high-level content selection

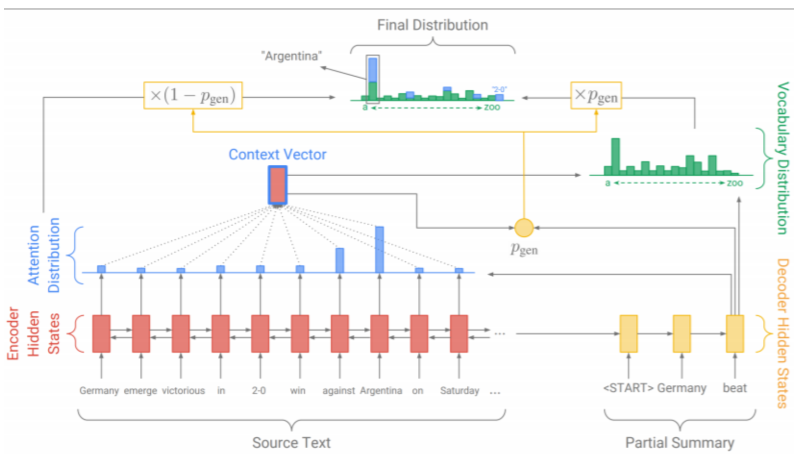
Neural Summarization

* Copy mechanisms

: Summarization 영역에서 유용하게 사용된다.

(Copying & generating) = (extractive & abstractive) 이기 때문에

: seq2seq+attention 같은 경우, 결과는 유창하지만, 세부적인 부분에서는 X



<Copying & Generating >

P_{gen}: Generating

$1-p_{gen}$: Copying

$$P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i:w_i=w} a_i^t$$

Neural Summarization

Q) Copy mechanism에 문제점은 없을까?

A1) 지나치게 너무 많이 Copy를 하는 경우가 있어. 또한, abstractive를 원했는데,

extractive가 결과로 나올 때가 많아..

A2) selecting content을 할 때, 성능이 안 좋게 나타나는 경우가 많다!

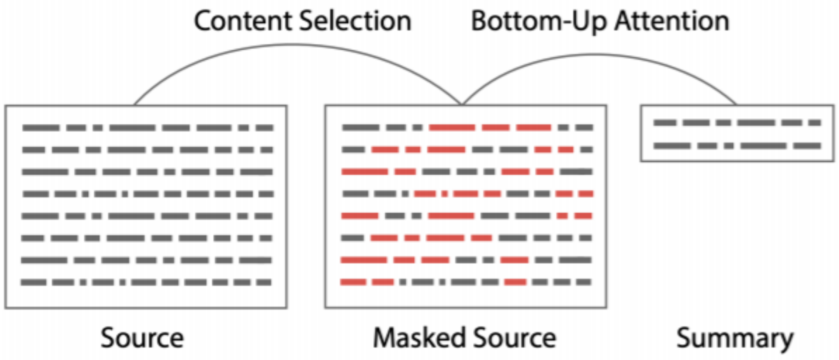
(특히, input이 길 때)

Q) 그럼 해결책은?

A) Bottom-up summarization

(Tagging model &

seq2seq+ attention)



Dialogue (chitchat)

: 요약에서 좋은 성능을 보였던, seq2seq+attention이 해당 주제에 문제점 발생 (Genericness, Irrelevant responses, Repetition, Lack of context, Lack of consistent persona)

(예시)

input(S): 밥 먹었니?

seq2seq(T): 날씨가 너무 좋다!

<Solution>

$$\log \frac{p(S,T)}{p(S)p(T)}$$

* Input: S, response: T

$$\hat{T} = \arg\max_{T} \left\{ \log p(T|S) - \log p(T) \right\}$$

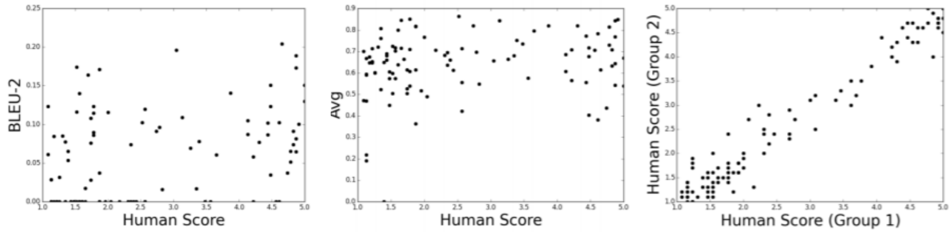
NLG evaluation

NLG evaluation

: 생성된 text에 대해서 평가는 어떻게 진행해야 할까?

Q) 기존의 Word overlap based metrics (BLEU, ROUGE, METEOR, F1, etc.) ?

A) 요약, 대화에서는 평가 방법으로 너무 좋지 않다



(a) Twitter

Q) perplexity, word embedding matrix

A) 모두 인간의 판단과 무관하다! 평가하는 지표로 사용할 수 없다!

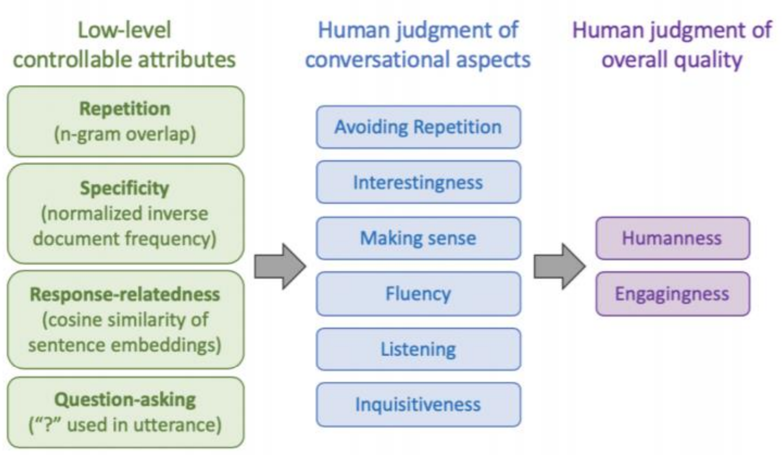
NLG evaluation (con't)

Q) 그렇다면, 사람이 모든 평가를 하는 것이 좋냐?

A) 거의 그렇다. 다만 사람들은 '주관적'이라는 문제점을 가지고 있다.

(Feat. expensive & slow)

Q) 이것은 어떻게 해결할 수 있을까?



TRAIN AND TEST