Lecture 15. Natural Language Generation

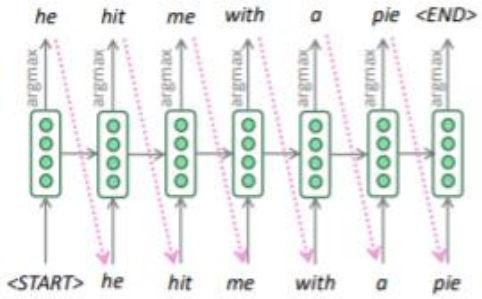
Ⅰ. Decoding Algorithm

- Natural Language Understanding(NLU) : 자연어를 기계가 이해할 수 있는 형태로 변환하는 자연어 처리 기술 분야. 지금까지 대부분의 연구가 NLU에 집중되어 있음.
- Natural Language Generation(NLG) : 시스템 계산 결과를 자연어로 자동으로 생성하는 자연어 처리 기술의 분야. 주어진 input x에 대해 새로운 text를 생성해내는 작업으로 기계번역, 요약, 채팅, 스토리텔링, QA 등이 있다.

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)을 골고루 만족하면 자연어를 잘 생성한다고 말할 수 있다.

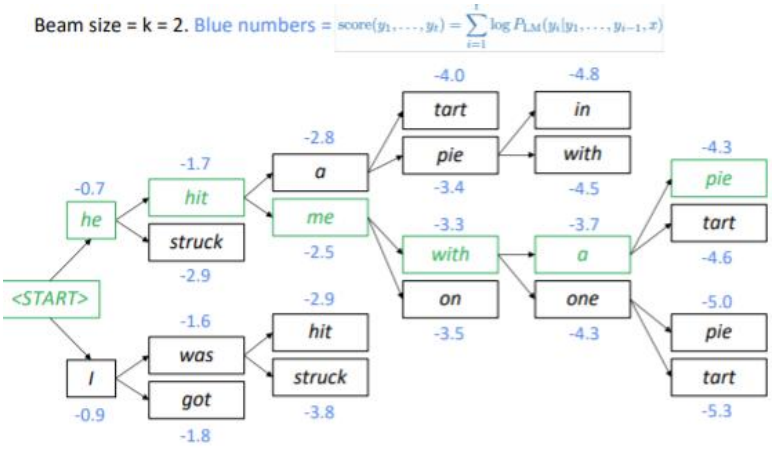
LM을 학습한 후에 NLG를 적용하는 방법으로 Decoding Algorithm이 있다. 가장 가능성이 높은 출력 시퀀스를 디코딩하는 것이 아니라 최대한 가능성이 높은 출력 시퀀스를 디코딩한다. Vocab의 크기가 어마어마하게 큰 경우에는 완전 탐색을 하는 것이 어렵기 때문에 heuristic한 탐색 방법인 decoding algorithm을 사용한다.

1. Greedy Decoding



각 출력을 예측하는데 매 스텝에서 argmax를 사용하여 가장 가능성이 높은 단어 한 개를 선택한다. 매 스탭마다 한 개만 선택하면 되기 때문에 탐색하는데 매우 빠르다. 하지만 어떤 스텝에서 실수를 하는 경우에는 그 실수가 해당 스텝 이후의 단어를 선택하는데 까지 영향을 미치기 때문에 최종 출력 결과가 좋지 않을 수 있다.

2. Beam Search



Greedy Decoding이 확장된 형태. k개의 가능한 가설들을 두고 가장 높은 확률을 갖는 문장을 찾아 나가는 방식이다. k=1인 경우 greedy decoding과 동일하다. 누적 확률을 크게 하는 단어들을 따라서 문장이 생성된다. 누적 확률을 크게 하는 단어들을 따라서 문장이 생성되는데 확률에 로그 값을 취한 형태이기 때문에 음수 값을 가지며 절대값이 작은 단어를 따라서 문장이 생성된다. Beam size가 너무 작으면 주제에 더 가깝지만 말이 안되는 단어를 뱉는다. Beam size가 너무 크면 너무 일반적이고 짧은 답변을 뱉으며 BLEU score를 떨어뜨릴 수 있다.

3. Sampling-based decoding

Beam search에서 큰 beam size를 가지더라도 너무 일반적인 답변을 얻지 않기 위한 방법이다. Pure sampling과 Top-n sampling이 있다.

1) Pure Sampling

Greedy decoding과 비슷하지만 argmax 대신 sampling을 사용한다.

2) Top-n Sampling

Pure Sampling처럼 완전하게 랜덤 샘플링을 하는 것이 아니라 확률이 가장 큰 n개의 단어들 중에서 랜덤 샘플링을 한다.

Ⅱ. Neural Summarization

Summarization은 주어진 input x에 대해, x의 주요 정보를 포함하는 요약된 y를 생성해내는 작업이다. Extractive summarization은 문서 내에서 핵심이 되는 문장을 추출해 비교적 쉽지만 제한적인 요약이 결과로 나온다. Abstractive summarization은 문서의 중요한 내용을 담은 새로운 문장을 생성하여 보다 유연한 결과를 얻을 수 있지만 비교적 어렵다. 이후에 다룰 neural을 적용한 요약

은 abstractive summarization이다. 원문에서 요약된 결과를 보면 문장 그대로를 추출하는 것이 아닌 새로운 문장을 생성해냄을 확인할 수 있다.

1. Evaluation Score : ROUGE

   -BLEU score : 주로 기계 번역 평가를 위해 사용한다. Precision을 기반으로 하고 brevity penalty를 준다.

$$\text{BLEU} = \min\left(1, \frac{\text{output length}}{\text{reference length}}\right)\left(\prod_{i=1}^{4} \text{precision}_i\right)^{\frac{1}{4}}$$

   -ROUGE score : 주로 summarization 평가를 위해 사용한다. Recall 기반으로 하고 brevity penalty를 주지 않는다.

ROUGE-N

$$\frac{\text{number of overlapping n-grams}}{\text{n-grams in reference summary}} = \frac{\sum_{s\in\{\text{ReferenceSummaries}\}} \sum_{gram_n \in s} \text{Count}_{\text{match}}(gram_n)}{\sum_{s\in\{\text{ReferenceSummaries}\}} \sum_{gram_n \in s} \text{Count}(gram_n)}$$

2. Pre-neural summarization

   이전의 요약 시스템은 대부분 추출하는 방식이다.

   -Content Selection : 포함할 중요 문장을 선택한다. 중요 문장은 topic이 해당 문장에 존재하는지, 문장의 위치는 어디에 있는지 등으로 선택한다.

   -Information Ordering : Content Selection에서 선택한 문장들을 중요도에 따라 순서대로 나열한다.

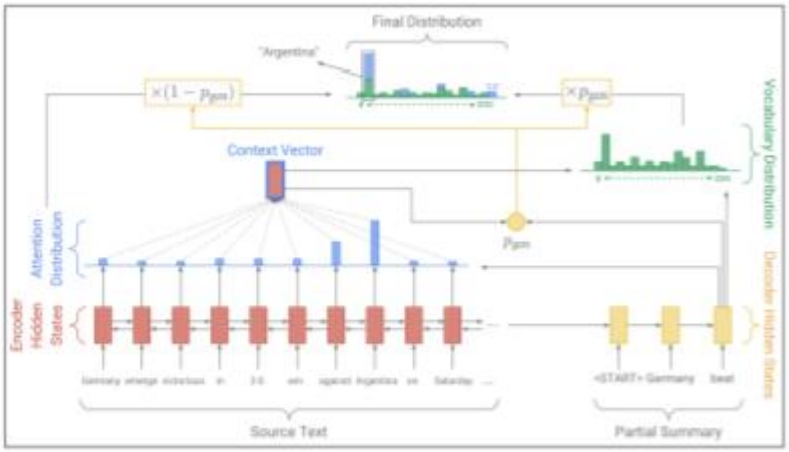
   -Sentence Realization : 요약문을 생성한다.

3. Neural Summarization

   신경망을 이용한 요약은 Seq2Seq과 Attention을 사용해 요약문을 생성한다. 하지만 Seq2Seq과 Attention만을 적용한 모델은 디테일을 잡아내기에 한계점이 있다. 문장을 생성할 때 out-of-vocabulary 문제가 발생할 수 있고 고유명사들의 출력 확률이 낮아지는 문제가 있다.

Ⅲ. Copy Mechanisms

Copy Mechanisms을 Seq2Seq과 함께 사용해서 디테일을 잡아내고자 했다. Attention에서 copy를 더 잘할 수 있도록 처리한 것이다.



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

P_{gen}은 단어를 copy할지 생성할지 결정지어주는 것이 generation probability이다. Context vector와 디코더 상에서의 타임 스텝의 hidden state를 입력 받고 디코더 상에서의 input을 입력 받아서 시그모이드를 통해 계산할 수 있다. P_{vocab}은 Context vector와 디코더 상에서의 타임 스텝의 hidden state를 가지고 softmax를 취해 계산할 수 있다. Attention 분포는 입력된 단어들로부터 얼마나 copy할건지에 대한 분포를 가지고 있는 것이다.

기존의 pre-neural summarization은 중요 문장을 선택하는 부분인 content selection과 요약을 하는 부분인 surface realization으로 나누어 동작한다. 하지만 neural approach는 그런 것 없이 하나로 묶어져 나오기 때문에 전체적인 것을 보지 못하는 문제가 발생한다. 이러한 점을 보완하기 위해 등장한 방법이 bottom-up이다. Word가 포함되었는지 포함되지 않았는지에 따라 0과 1을 태깅하여 모델은 word가 포함되지 않은 부분에는 집중하지 않는다. 간단하지만 매우 효과적이다.

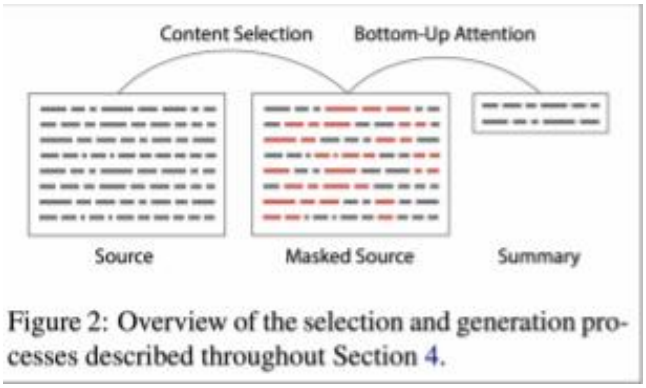


Figure 2: Overview of the selection and generation processes described throughout Section 4.

IV. NLG Using Unpaired Corpus

지금까지는 input에 대응하는 output을 미리 준비한 후에 학습시키는 지도 학습에 기반하고 있다. (ex) 문장 쌍 준비하여 학습) 그러나 이러한 방법의 문제는 입력-출력 쌍의 데이터를 대량으로 요

구한다. 고성능의 자연어 생성 시스템을 만들기 위해 그 훈련에 필요한 말뭉치의 확보부터가 현실적으로 매우 어렵다. 이러한 한계점을 돌파하기 위해 unpaired corpus를 활용한 비지도 학습이 등장한다.

1. 인공 신경망이 반드시 학습해야 하는 것

   -어떤 스타일의 문장이 들어오더라도 그 스타일에 의존적이지 않는 본질적인 의미를 latent vectors의 형태로 인코딩해야 한다.

   -인코딩된 latent vectors가 주어졌을 때, 각 스타일에 해당하는 디코더는 해당 스타일의 문장을 생성할 수 있어야 한다.

2. 인공신경망의 학습 방향

   인공신경망은 autoencoder loss와 cycle loss를 최소화하는 방향으로 학습한다.

   -autoencoder loss : X 스타일의 문장 x를 latent vectors로 변환한 후, 이를 다시 X 디코더를 이용해 문장 x′가 생성되었을 때, x′와 원래의 문장 x가 얼마나 다른지

   -cycle loss : X 스타일 문장 x를 변환 과정을 통해 Y 스타일의 문장 y로 변환하였을 때, 이 문장 y를 다시 X 스타일로 변환한 문장 x″과 원래의 문장 x는 얼마나 다른지