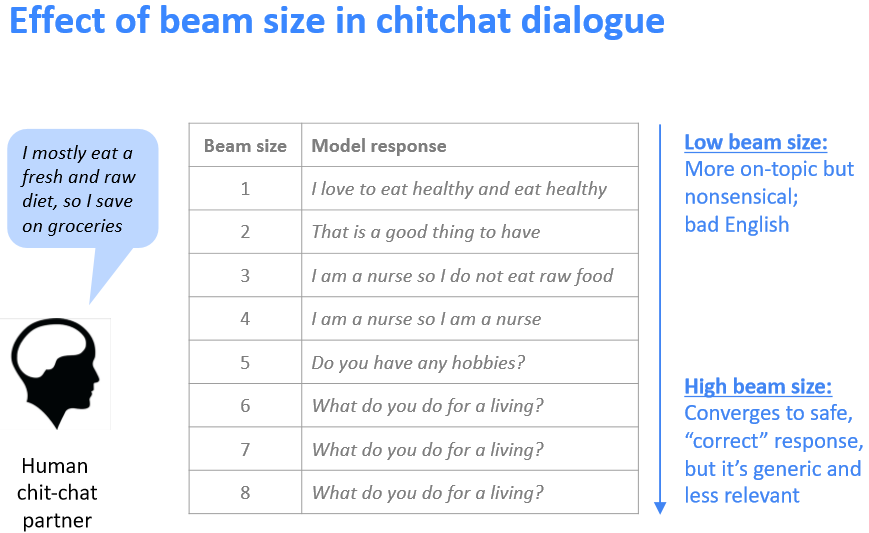


추가적인 Recap으로 우리는 디코딩 시 알고리즘으로 그리디 & 빔서치를 배웠다.

빔 사이즈 k 가 1이면 greedy search와 다를 것이 없고, 크다면 computationally expensive 하고, 너무 크다면 BLEU score를 감소시킨다.(Tu, Koehn 2017)

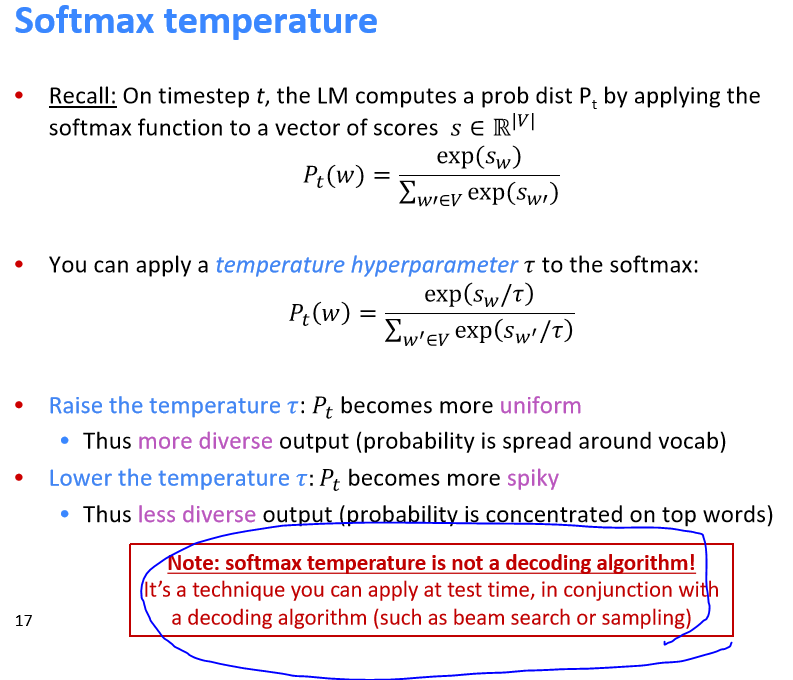
그 이유는 k가 크다면 too short translation을 하려고 하고, 때때로 chit-chat dialogue에서 너무 generic한 답변을 하려하기 때문이다.



따라서 새로운 방법이 제안 됐는데 그것이 Sampling-based decoding 이다.

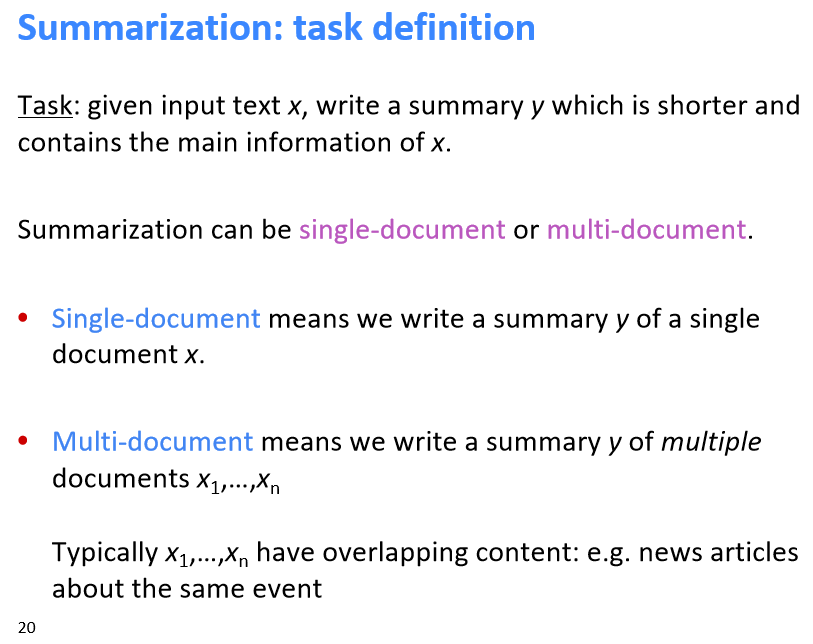
다음 단어를 고를 때 step t에서 확률 분포에서 랜덤하게 1개 고르는 것이다. Argmax의 파라미터가 아닌 sample을 이용하는 방식이다.(그리디한거 같지만 완전 그리디는 아님)

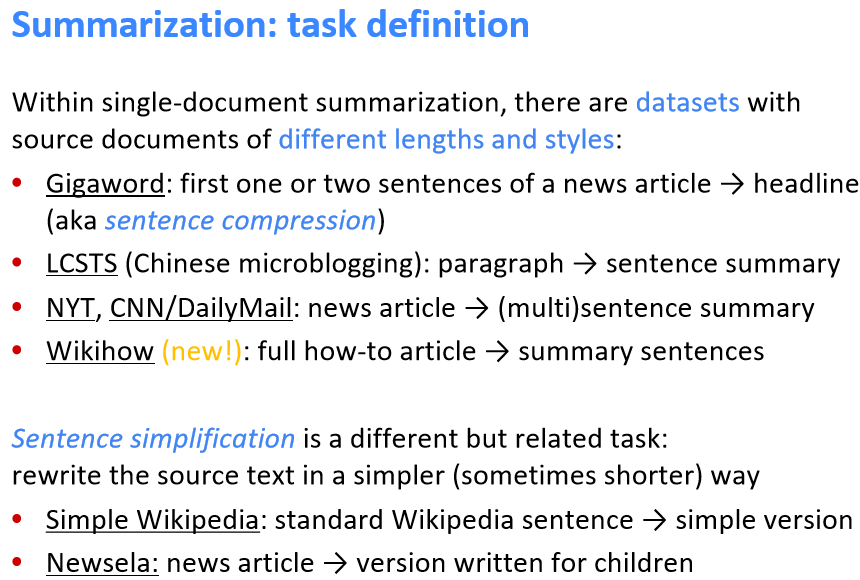
하지만 너무 랜덤하게 진행하면 문제가 될 수 있으니 표본을 top-n으로 줄이는 것이다.(n은 하이퍼 파라미터) 이때 n이 1이면 same greedy 이고 n이 커질수록 diverse하고 risky하며, n이 작을수록 generic하고 safe하다.(V면 pure sample)



Softmax를 변형시켜 사용하는 방법이다.

**Sec2: NLG Tasks & Neural Approaches to them**

라고하는 task이다.



라는 데이터셋들이 있고 비슷한 task인 sentence simplification이 있다.



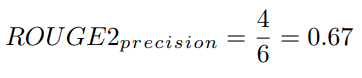
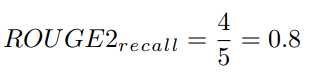
의 두 가지 summarization 방법이 있다.

Pre-Neural Summarization은 mostly extractive 였다.

1. Content selection: 문서에서 문장의 일부를 선택한다. (tf-idf등을 이용해 topic key words가 문장에 존재하는지를 따져서 or 문서에서 문장이 어느 곳에 위치한지를 따져서 // graph based 로 선택할 수도 있다.)
2. Information ordering: 문장마다 순서 매김(multi-documents 일 때 특히 중요)
3. Summary를 위한 마지막 단계: sentence realization

Summarization Evaluation에는 ROUGE가 있다. (Recall-Oriented Understudy for Gisting Evaluation)

BLEU 처럼 n-gram overlap에 기반한다.

“the cat was found under the bed” vs “the cat was under the bed”(gold)  
ROUGE-2를 이용한다면

(bi-gram으로 나누어서 분모, 분자 번갈아 가면서 계산)

ROUGE는 recall에 BLEU는 precision에 기반(둘 다 gold는 사람이 번역한 것)

ROUGE-1,2, L(LCS 기법으로 L을 구함.)이 많이 쓰임

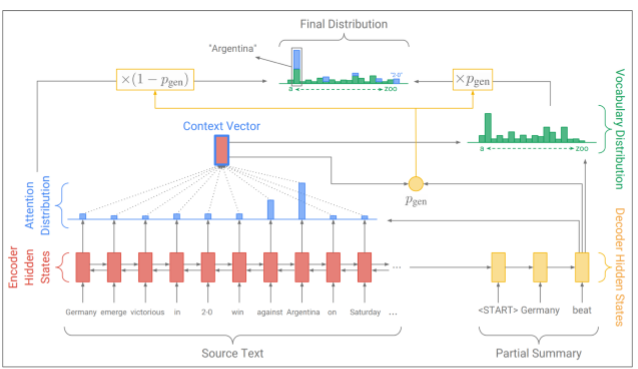
F1 Score로도 많이 쓰임.

Neural Summarization(2015) – seq2seq with attention을 활용한 summarization이 연구되었다. 이후 다양한 연구가 이루어지고 있다.

+취약했다. 따라서 copy mechanism이 seq2seq with attention에 도움을 주었다. (hybrid extractive/abstractive approach의 성능 향상)

아래 그림을 보면 일반 attention 처럼 context vector 생성하면  
1. Attention distribution(encoder states & decoder state) / 2. P-gen(context vector & decoder state) / 3. Vocab distribution(context vector & decoder state)

으로 마지막 그림을 보면 attention 부분은 카피할까? 초록 부분은 새로 생성할까? 의 비율 싸움이다.(그대로 카피가 좋을 경우도 있다.)



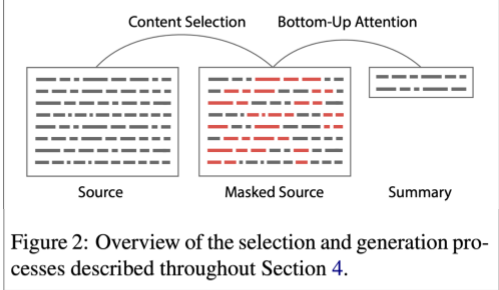
Problems: 긴 source text에 대해 너무 많이 copy 할 수도 있다. 또한 긴 input에 대해 전반적인 content selection(카피 기능 or 생성 기능을 실행하기 위해 일단 인풋 문장을 고르는 것?)이 잘 안된다.

즉, 여기서는 decoder가 surface realization을 attention이 word-level content selection 역할을 해서 summarization이 진행 됐지만 global content selection(문장을 전체적으로 선택 못함.)이 잘 되지 않았다.

따라서 제안된 것이 Bottom-up Summarization이다.

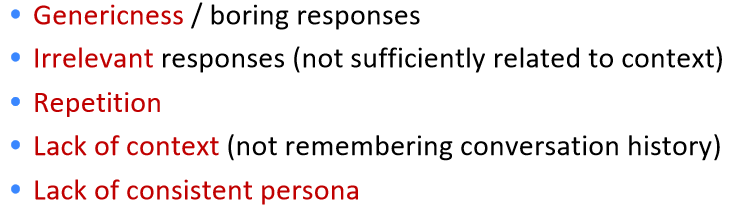
1. Content selection stage: neural sequence-tagging model이 모든 words에 include or don’t-include tag를 한다.
2. Bottom-up attention stage: seq2seq+attention이 don’t-include word는 mask한 뒤에 summary를 한다.

=> 간단한 구조지만 효과적이다: overall content selection이 되며 less copying of long sequences (more abstractive)

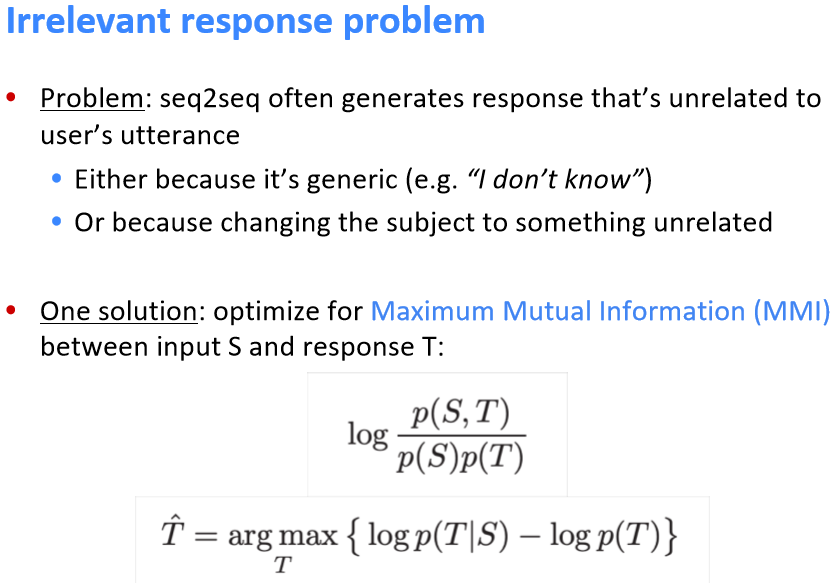


이번엔 NLG(Natural Language Generation)의 다른 Task인 Dialogue이다. (챗봇 같은거)

이 Task에도 역시 seq2seq attention이 적용되었지만 문제가 있었다.(pervasive deficiencies)



첫 번째 문제는 Irrelevant responses(+genericness) 인데



처럼 문제가 발생하고 objective function을 바꿔서 해결했다.

두 번째는 genericness 만의 문제 해결인데, beam search에 rare words를 많이 나오게 하던가 sampling decoding algorithm 을 해서 못 본 단어를 많이 나오게 한다.

또는 디코더에 추가적 내용을 포함하도록 조건을 준다. 추가적인 문장을 줘서 학습을 시킨다는 듯 핵심은 뭔가 새로운 단어를 추가시켜 주는 것이다. => pretty strong 하다.

그 다음은 repetition problem 이다.

반복되는 n-grams를 빔서치 도중에 막는다. (꽤 효과적) 또는 여러 번 같은 단어가 나오지 않도록 어텐션 메커니즘을 예방하는 objective를 설계한다. = coverage mechanism

또는 discourage repetition 하는 objective 설계

계속 깊게는 설명 안하고 간단히 설명한다.

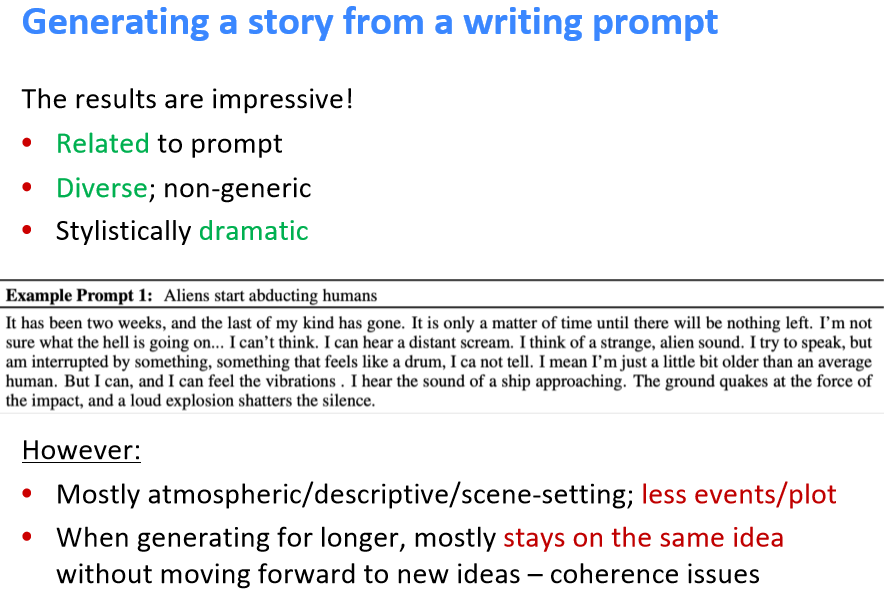
그 다음은 storytelling task 이다.

그림을 보고 이야기를 지어내는 것이다. (하는 일은 like image captioning 하지만 근본 자체는 다르다.)

Sentence encoding space 를 이용하는데 여기서는 skip-thought vectors라 한다.(방법은 word2vec이랑 비슷)

COCO(원래는 image captioning 데이터셋)을 이용해 images를 skip-thought encoding 으로 매핑하는 것을 학습한다. 그리고 디코더 역할인 Taylor Swift Lyrics에 넣어 주기 위해 두 기능을 합치면 된다.

이미지에서 볼 수도 있고 그냥 한 prompt( ex) “the mage, the warrior, and the priest”) 등에서 스토리텔링을 할 수도 있다.(Fan 2018)

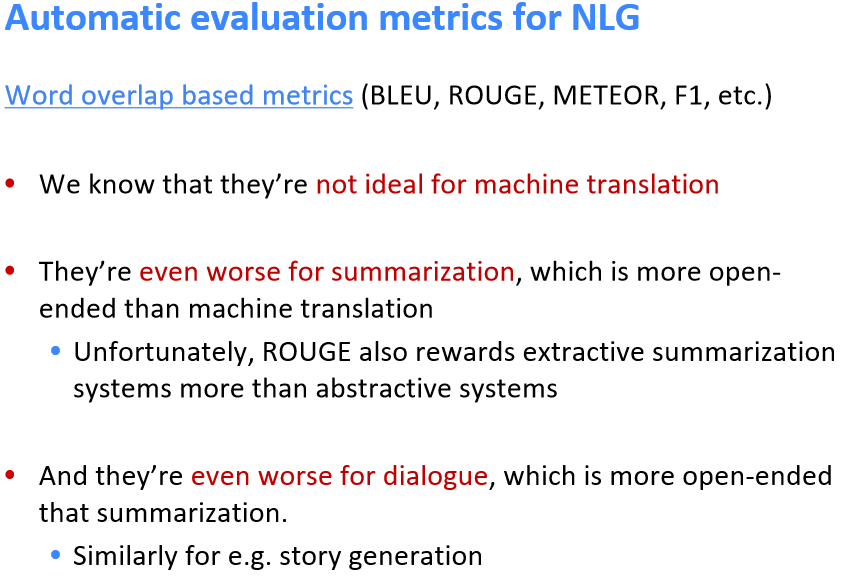


의 장/단점이 있고 결과가 있다.

Poetry generation도 할 수 있지만 스킵!

Sec3: NLG Evaluation(pretty important)

BLEU, ROUGE, METEO), F1 etc



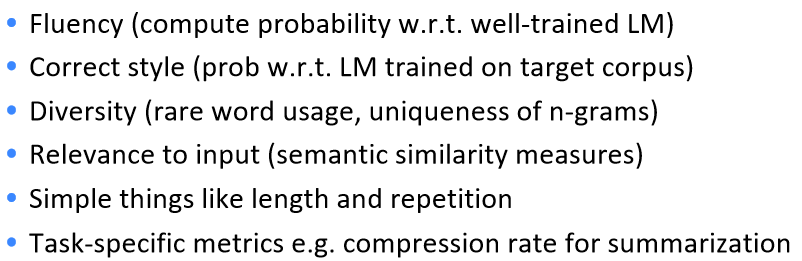
이전에 BLEU와 ROUGE는 n-gram 을 활용한 word overlap method 였다. 하지만 이것은 사실 NMT나 SUMMARIZATION 모두 정확한 것은 아니고 특히 dialogue, storytelling 등 다양한 답변을 도출하는 task에 대해서는 좋은 metric이 될 수 없다.

Perlexity는 LM 모델이 얼마나 강한지(오타 교정 등 약간 뭔가 물리적인 느낌)는 설명 가능 하지만 내용에 알맞게 잘 생성했는지를 말할 수 없다.(의미적인 느낌)

또는 word embedding의 유사도 비교?(word embedding average) => 하지만 이것은 되려 human judgement(gold)에 대한 correlate가 힘들다.

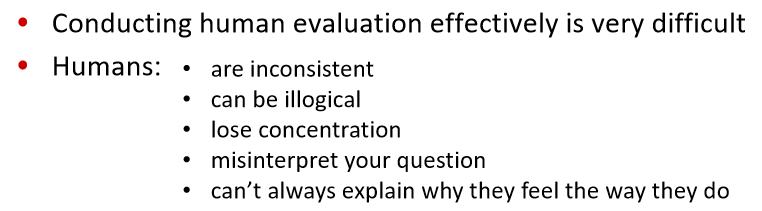
결론적으로 전반적인 quality를 캐치할 automatic metric이 없다.

하지만 이런 것들을 세세히 따질 수는 있으며 100%는 아니지만 어느정도 성능 평가에 반영할 수 있다.

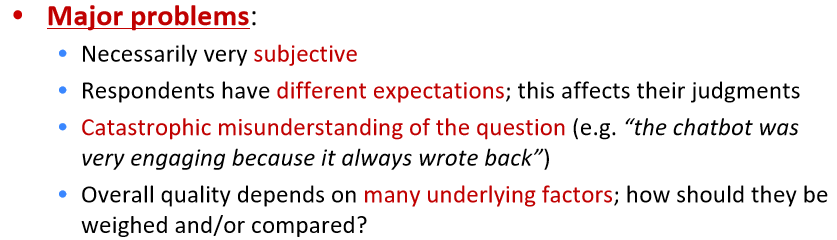


Gold로써 작동하는 human evaluation이 그러면 평가의 상대로 best인가?

아니다. 일단 gold를 구성하기 위해 slow & expensive 하고, 사람마다 매우 주관적이고 한 사람 내부에서도 inconsistent 할 수도 있다. 또한 멍청한 사람은 illogical 할 수도 있다.



또한 ylabel(gold)를 위한 human judgement 뿐만 아니라 평가를 인간이 할 수도 있는데 이것도 마찬가지로 매우 주관적이며 다음과 같은 문제가 있다.



Sec4: Thoughts on LNG research, current trends, and the future

아래와 같은 흥미로운 연구 방법이 있다.

