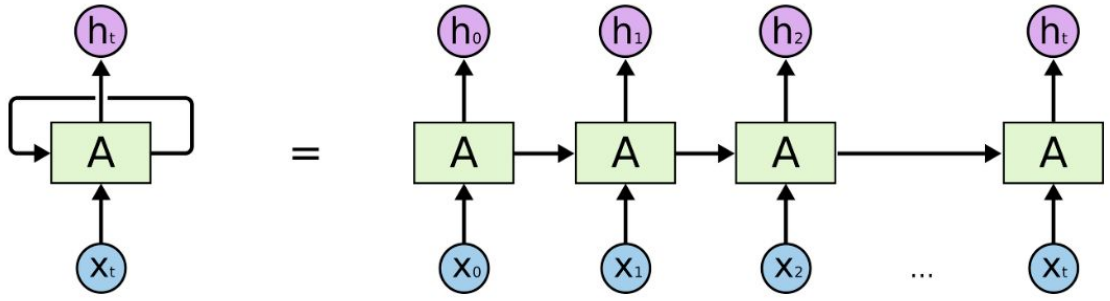
Backtranslation, MixupTraining 파트(조민호)

여러분... 최선을 다해 이것저것 참고해보았으나... 너무너무 어렵습니다... RNN LSTM, seq2seq도 공부 해야할듯... 일단 한번도 안찾아본 것보다는 도움이 될 것 같아 이것저것 정리해봤습니다.

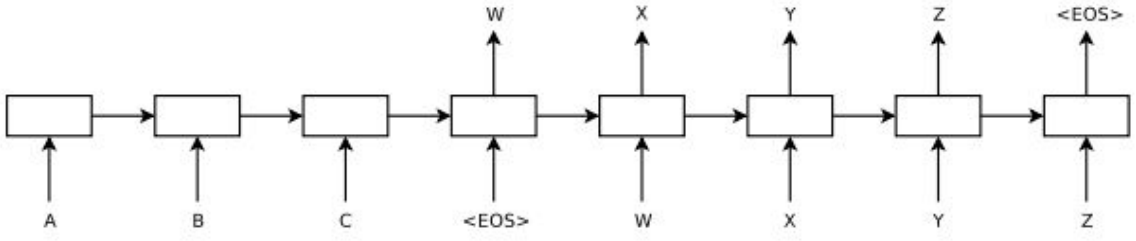
하기전에 RNN 간단히)

인간은 매초 처음부터 생각하는 것이 아니라 바로 전의 문맥에 맞게 다음 단어를 이해한다. RNN은 이를 잘 활용해 루프를 만든다. 이 루프는 과거의 데이터가 미래에 영향을 줄 수 있도록 한다.A는 xt를 입력값을 가지고 다음 단계 뉴럴 네트워크로 이동하게 해준다. 즉, 네트워크를 계속해서 복사해서 다음 네트워크로 넘겨주는 구조와 유사하게 돌아간다.



Seq2Seq도 간단히)

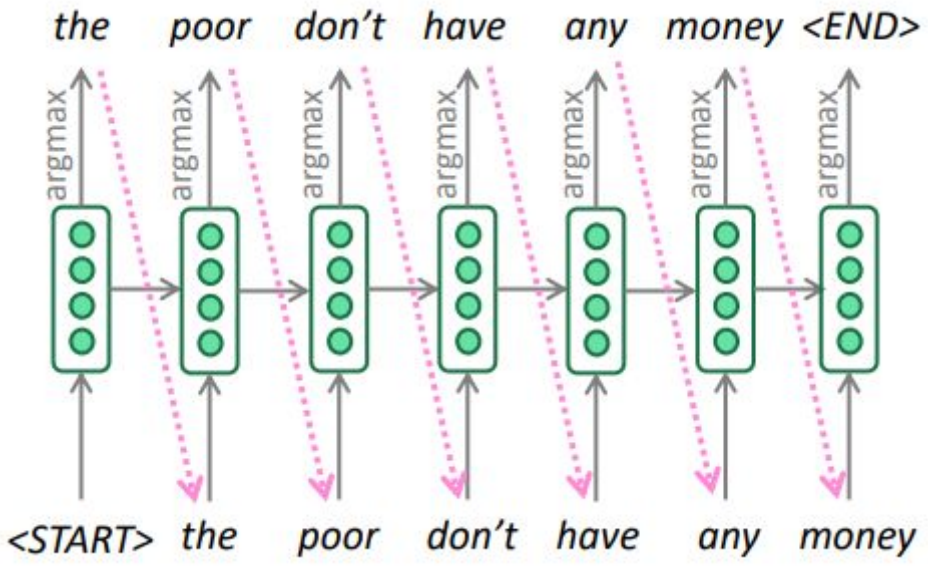
seq2seq는 encoder와 decoder를 사용한다.



위 그림에서 보면, ABC를 입력으로 넣으면 WXYZ를 출력해주는 원리다. 언어로 생각하면 영어를 입력으로 하면 프랑스어가 나오는 것이다.

("the cat sat on the mat" -> [Seq2Seq model] -> "le chat etait assis sur le tapis")

이 때 ABC를 넣는 부분을 Encoder라고 하고 최종적으로 ABC가 끝나고 Decoder에게 hidden state로서 넘기면, 아래와 같은 방식으로 예측한다. 이 때 각 cell을 거칠 때마다 최대 확률값을 가지는 단어를 출력값으로 내보낸다. 즉, the 가 들어갔을 때 poor가 나오겠구나 추측하도록 해준다(여기서 최대 확률값을 가지는 단어를 계산한다는 점을 기억해두도록 하자. backtranslation의 이유가 된다.[정확한지는 모르겠다…])



Text에서의 BackTraslation 원래 개념 살펴보기)

"improving neural machine translation models with monolingual data"(https://www.aclweb.org/anthology/P16-1009/) 논문에서 backtranslation 개념이 등장한다. 원래 생각하는 번역 학습에서는 본래 문서와 번역된 문장이 쌍으로되어있는 pararrel한 데이터를 사용한다(pararrel이라는 단어가 이런 의미였다!!). 하지만 우리가 사는 세상에는 번역된 쌍이 없는 monolingual로 되어있는 데이터는 매우 많고 이를 활용해보고자 하는게 이 논문의 주된 내용이라 할 수 있다. 논문에서는 세가지 방식으로 실험을 해본다.

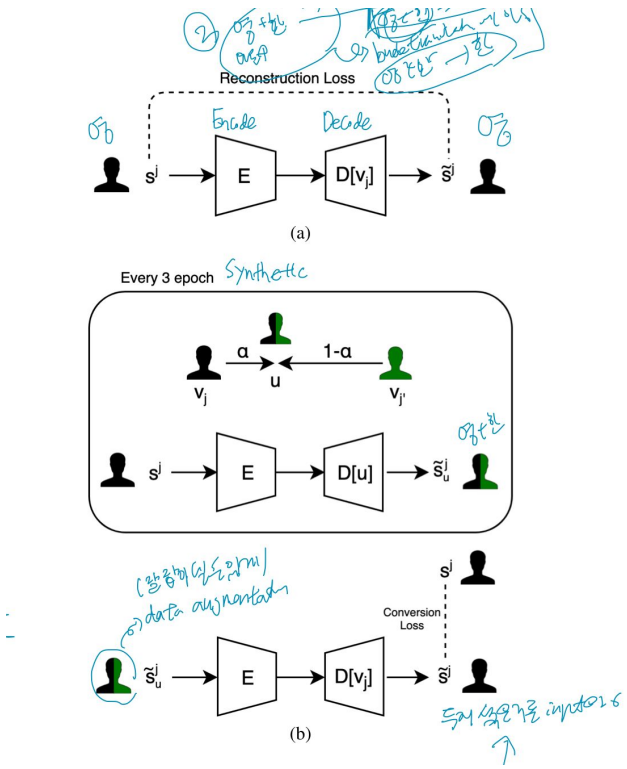
첫번째는 그냥 pararrel한 데이터만 있는 것, 두번째는 dummy로 <null>과 monolingual한 데이터를 연결한 데이터와 pararrel한 데이터를 섞어서 실험하는 것(예를 들어, 한-영 번역이 잘 되어있는 데이터에 한-null

쌍으로 되어 있는 데이터를 더하는 것)이다. 이는 소스문장인 X를 빈입력 null로 둠으로써 인코더로 전달되는 것들을 끊고 decoder에 활용할 수 있는 단어 수를 늘릴수 있도록 해준다.

세번째는 backtranslation 한 데이터와 pararrel한 데이터를 섞는 방식이다. 여기서 backtranslation이란 앞에서 pararrel한 데이터로 학습을 적당히 했다고 가정하는 model로 monolingual한 데이터를 번역한 것이다.(즉, 영<->한 번역한 machine이 있을 때 이를 활용해서 원래 가지고 있는 한 데이터를 영으로 바꾼다. 퀄리티는 안좋지만 바꾸는 것이다. 이건 언어라는건 한글-> 영어, 영어-> 한글로 바꿀수 있듯 symmetry하기 때문이다. 이미지에서 classification을 하는 경우라면 class로 이미지를 되살릴수 없으므로 symmetry 하지 않다.)

이렇게 실험을 해보니, 세번째 backtranslation 한 데이터가 가장 성능이 좋았다고 한다. 논문에서 이유를 밝히기 위하여 다시 또 실험을 하는데 이를 정확히 이해하지는 못했다.(논문 4.2.5절). 내가 소화한 바로는, 앞에서 말했듯이 decoder에서는 가장 높은 확률값을 내뱉아 번역을 하게 되는데 이 데이터에 monolingual한 데이터가 decoder 데이터에 들어가면 확률 계산할 때 고려해볼 수 있는 단어들이 훨씬 많아진다. 즉, 들어본적은 있지만 무슨 뜻인지 모르는 단어들을 학습함으로써 단어 폭을 넓히고 아닌 단어들을 쉽게 쳐낼 수 있도록 도와준다. 때문에 논문에서는 backtranslation이나 dummy data를 넣는 것은 dropout하는 것과같은 오버피팅을 막는 역할을 한다고도 말한다,

논문에서의 Backtranslation과 Mixup Training)



앞 text 논문에서 사용한 데이터는 영-> 한 처럼 명확한 pararrel data가 있는 데이터이지만, 우리가 만들고 싶은 데이터는 아이유->이적 같은 데이터는 존재하지 않는다. 그래서 논문에서는 1단계에서는 아이유를 encode decode 해서 decode한 아이유와 실제 아이유를 비교해서 학습을 한다.

2단계에서 backtranslation과 mixup training이 활용되는 것 같은데, text 데이터 backtranslation에서 언어를 예시로 들었으므로 여기서도 Asinger,B singer를 한 영으로 대체해 설명해보겠다. 첫번째 단계에서 영->영, 한->한 처럼 다시 내가 원하는 단어로 복구시키는 학습을 함으로써 영어(아이유)와 한글(이적)에 대해서 조금 알게 되었다. 이 지식을 이용해 두번째 단계에서는 내가 얻은 영어를 만들게 해주는 vector와 한글을 만들게 해주는 vector를 잘 섞어준다.(y' = By1 + (1-B)y2 [B에 대한 논의는 잘 이해를 못 해 밑에 모르겠는 부분에 적어두었다.]) 이를 활용하여 한글과 영어가 적절히 섞인 새로운 데이터를 만들 수 있다.

우리는 이 데이터를 비율에 맞게 한글이 많이 섞였다면 (한,한영 융합한 데이터) 영어가 많이 섞였다면 (영,한영 융합한 데이터) 쌍을 만들 수 있다. 이를 다시 활용하여 새로운 데이터셋처럼 활용해서 나를 복구해내는 용도로 활용할 수 있다. 이는 앞 논문에서 backtranslation이

영-한 공부를 조금 하고 한 데이터를 정확하진 않지만 번역해서 새로운 쌍을 만들어내는 방법과 유사하다.

모르겠는 부분1. 0.2 를 베타 distribution으로 넣어주면 B가 0이나 1에 가까워지고 sample이 고루 섞인 데이터보다 한쪽에 치우친 데이터가 더 많이 나올거라는 부분이 이해가 안감. 0.2면 엄청 중앙에 몰려있을텐데…?
모르겠는 부분2. 이 논문에서 backtranslation을 이용해도 결국 얻는 것은 내가 나를 잘 복구해내는 것인데 어떻게 다시 활용할 수 있는걸까? 학습한 아이유(A) 벡터를 이적(B) 벡터에 넣어주면 바뀔수 있다는걸 얘기하는 걸까?

참고한 URL)

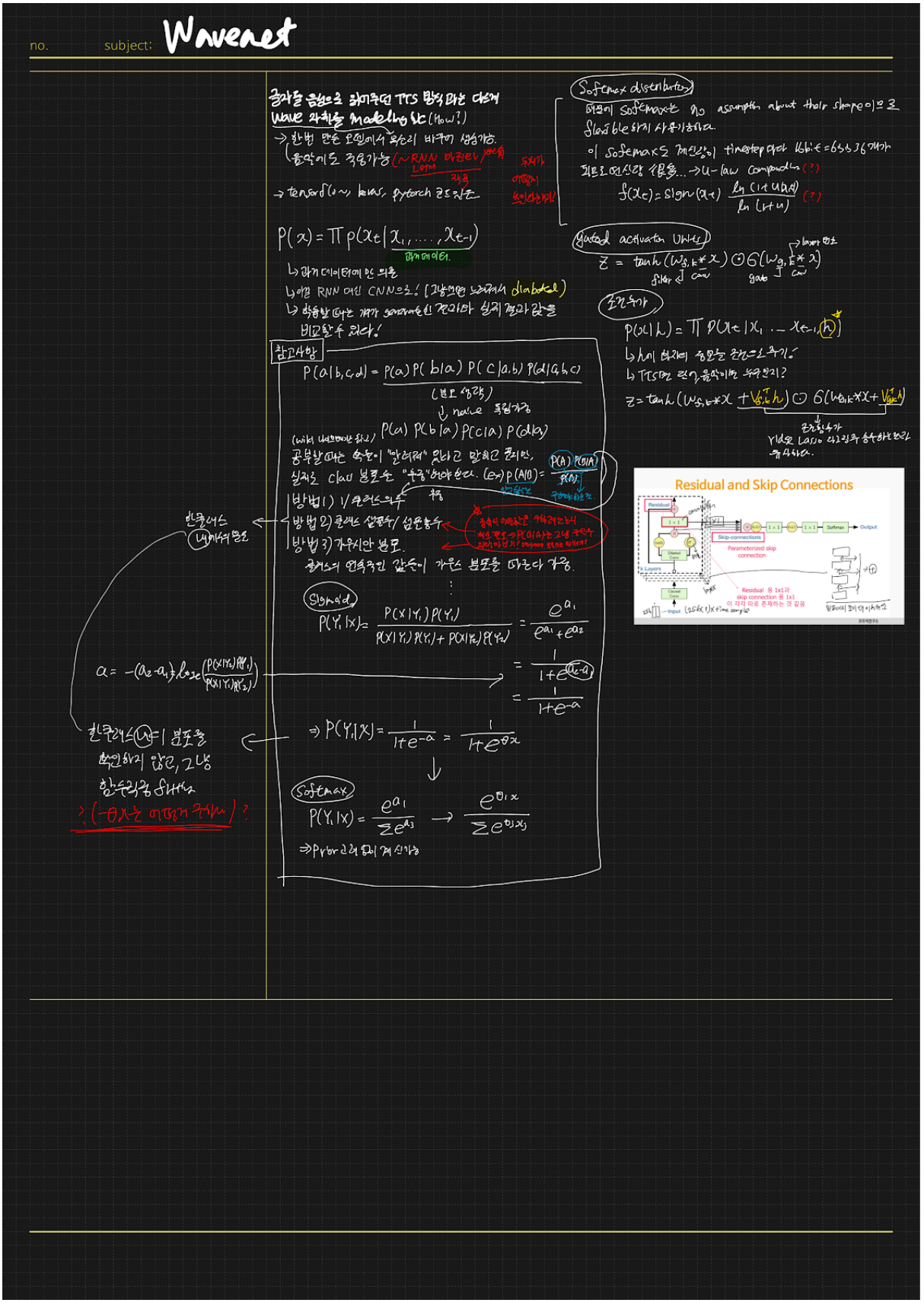
RNN LSTM 설명 자료)
https://brunch.co.kr/@chris-song/9

Seq2seq 설명 자료)
https://hyunkyung12.github.io/2018/10/19/seq2seq/

https://tykimos.github.io/2018/09/14/ten-minute_introduction_to_sequence-to-sequence_learning_in_Keras/

text에서의 backtranslation)
https://kh-kim.gitbook.io/natural-language-processing-with-pytorch/00-cover-10/02-monolingual-corpus

추가적으로 wavenet도 일부 공부하고 정리한 자료다. 이건 더 모르겠다..



참고한 자료)

Wavenet 논문
https://arxiv.org/abs/1609.03499

개괄
https://tensorflow.blog/tag/wavenet/

좀 자세한 설명
http://www.secmem.org/blog/2019/08/18/wavenet

softmax를 사용하는 이유에 대해 조금 추측해 볼 수 있는 자료.
https://taeoh-kim.github.io/blog/bayes-theorem과-sigmoid와-softmax사이의-관계/

Wavenet pytorch
https://github.com/vincentherrmann/pytorch-wavenet

GitHub 한글로 구현해놓은 데이터- 손석희 문재인 소리 만들기 예시
https://github.com/hccho2/Tacotron-Wavenet-Vocoder

- Wavenet 구현은 ibab(https://github.com/ibab/tensorflow-wavenet)의 구현이 대표적이다.
- ibab은 local condition을 구현하지 않았다. 그래서 train 후, 소리를 생성하면 알아들을 수 있는 말이 아니고, '옹알거리는 소리'만 들을 수 있다. 의미 있는 소리를 들을 수 있기 위해서는 local condition이 적용해서 구현해야 한다.