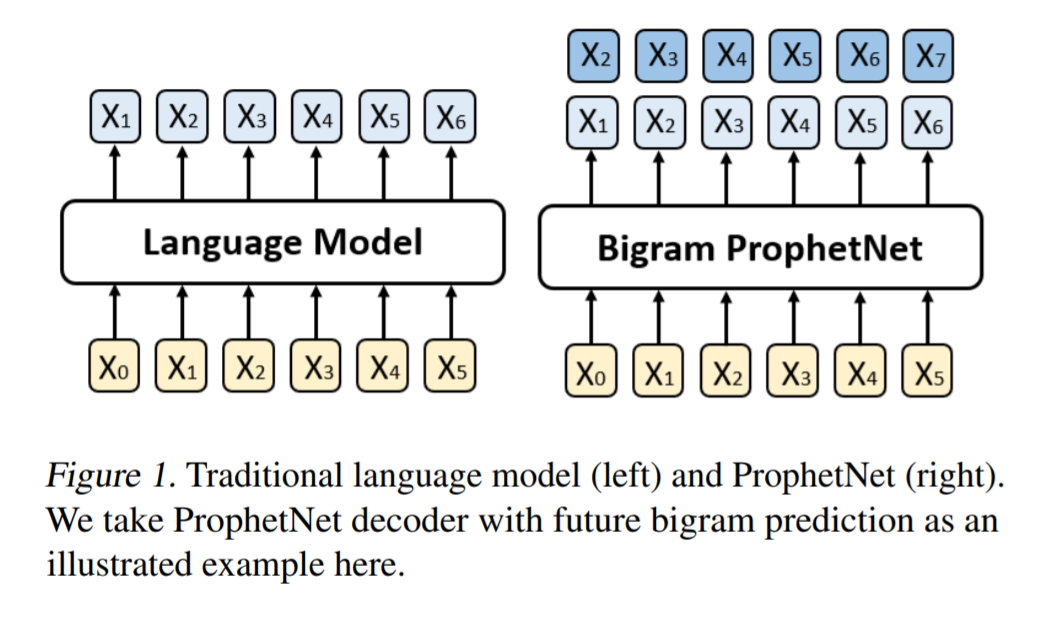
**[ Prophetnet : Predicting Future N gram for Sequence-to-Sequence Pre-training . 2019]**

1. 핵심 정리

Future n gram prediction (by n stream self attention mechanism<-XLnet에서 나옴)

그로 인해서 local에 집중하는 overfitting을 방지하고, 전체적인 문맥에 의존



**2. Introduction**

AR based model의 경우엔 가장 최신의 token에 집중함

이유로는, Local correlation이 long term dependencies보다 강하기 때문이고, Decoder 학습시 Teacher Forcing을 쓰기 때문에 future token planning과 modeling 사이의 bias가 없다. ( 후자의 말의 경우, 잘 이해가 되지는 않지만, 짐작으로는 실제 test시로 생각을 해보면 이전 시점의 output을 다음 시점의 input으로 쓰는 데, 이전 시점의 output을 planning이라고 하고, 그냥 teacher forcing으로 이전 시점의 실제 결과를 주는 것을 modeling이라고 하면, 이 둘에 대한 차이가 없다는 것을 나타내는 말 같음 )

추가로 Greedy Decoding하는 것이 beam search보다 local에 더 집중하게 만든다고 한다.

그래서 ProphetNet은 future n gram prediction을 쓴다. Future n gram이 extra guidance로 활용되고, 또한 local에 치중하는 overfitting을 방지하게 해준다.

Training 時 future n gram prediction을 하고, Test(Inference) 時 1개의 token(unigram)을 prediction한다. (+ main stream의 parameter는 모든 predicting stream과 공유된다.)

이를 위해서 XLNet에선 2 stream self attention에서 n stream self attention으로 확장했다.

# ProphetNet

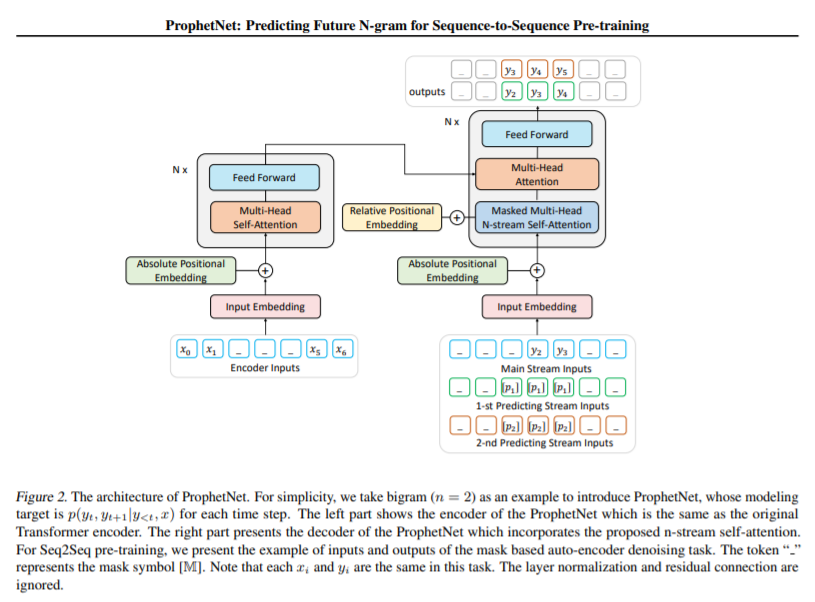
1. future n gram prediction
2. n stream self attention mechanism
3. modified positional embedding
4. mask based auto encoder denoising task for seq 2 seq

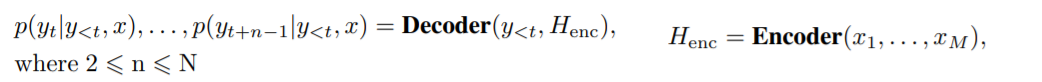
**2.2) Future n gram prediction**



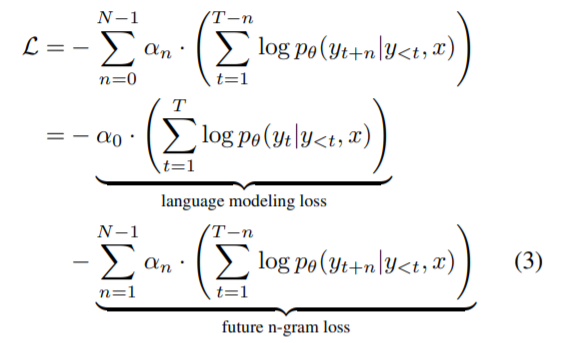
Conditional language modeling에서는 이번 시점(t)의 token만 예측했는 데, 여기서는 n개(y\_t … y\_t+n-1)의 token을 예측한다.

Prophet net은 encoder부분은 transformer encoder와 동일(multi layer and multi head attention) 하지만 decoder부분은 multi head n-steam self attention mechanism이 새롭다.(Figure 2 Masked Multi Head N stream Self Attention 참조)

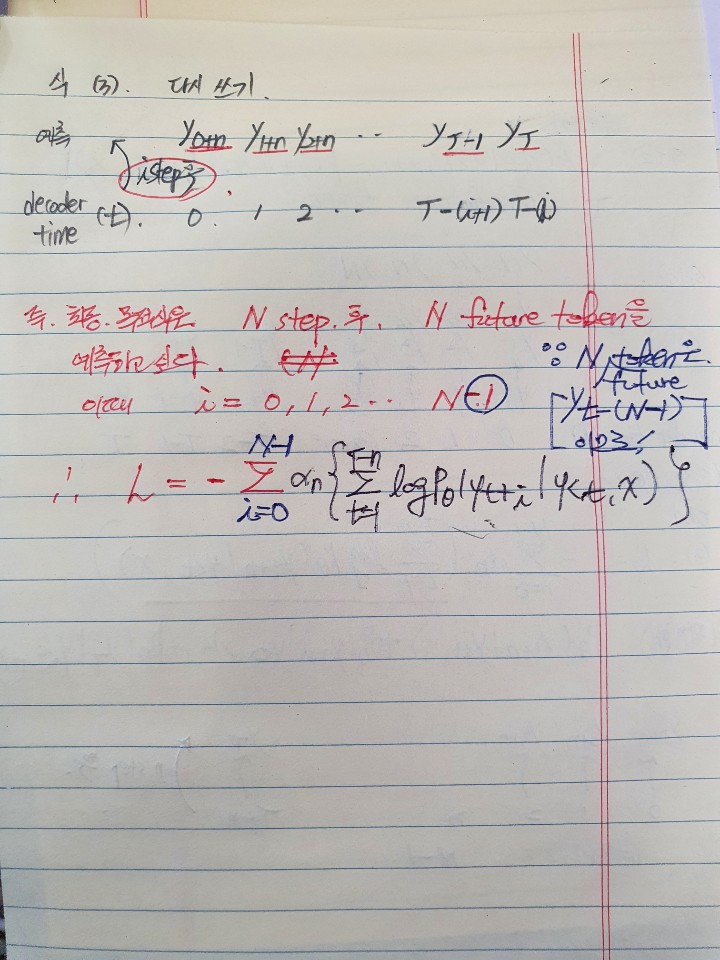




Decoder가 n future tokens을 동시에 예측함. ( 여기서 n은 n future tokens에서의 n이고 T는 decoder의 seq len, Large N은 ??\_크게 의미부여 하지 말자 , 그리고 M은 encoder의 seq\_len이다. 저 부등호의 의미로 볼 때, Large N의 max 값은 T가 될 것이다. )

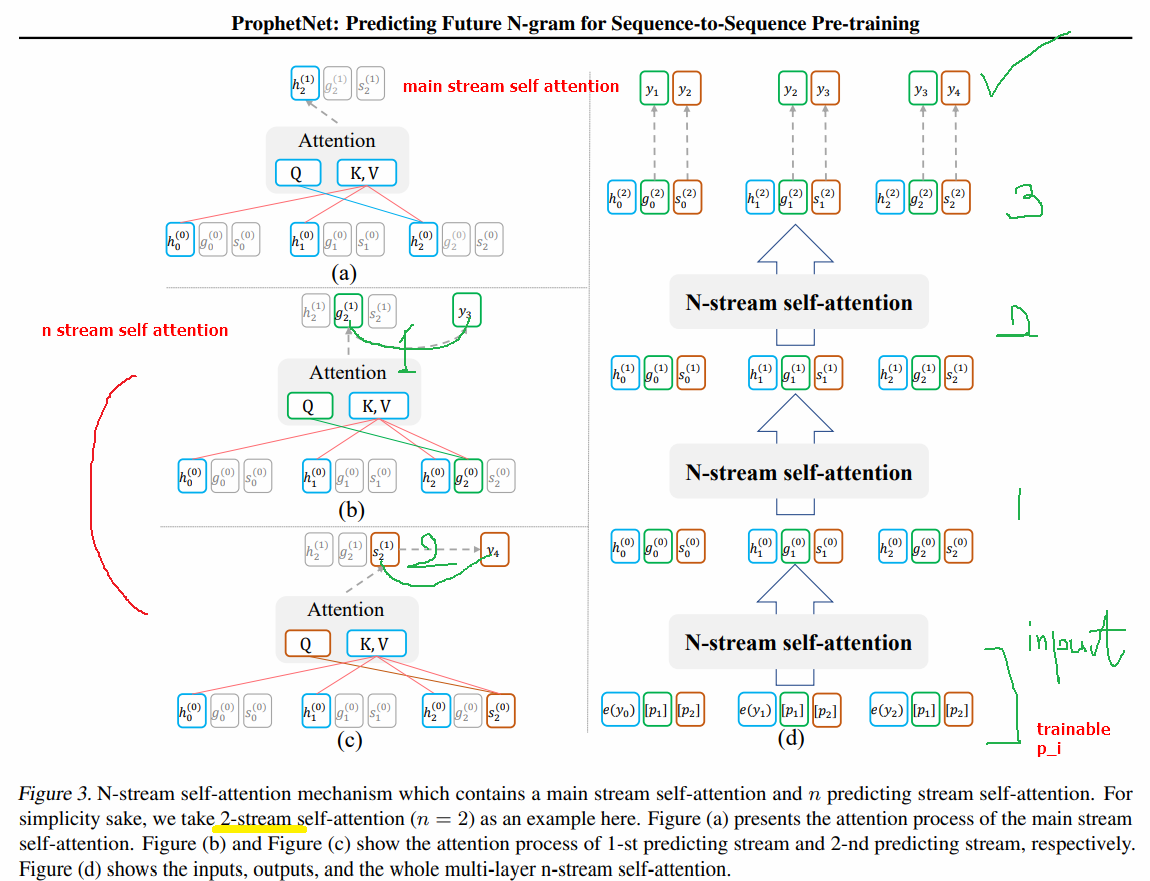


n은 n future token의 것을 의미하고, t는 decoder의 timestep(혹은 seq len)을 의미함. 즉, 다시 정리하자면 N future token을 예측하는 것이 목적이고, small n은 N future token 중 sub을 의미한다. (참 헷갈리게 적었음.) 그래서 이것을 각각의 decoder seq len마다 더해서 식을 적어나간 것이고. 이는 위 식 처럼 두 개로 나눠진다.



**2.3) N stream self attention**

Original Transformer에 (이것이 main stream self attention이고) 여기에 n-stream self attention mechanism이 추가됨.



중요한 것은 predicting stream and main stream이 training 시 parameter를 공유한다는 것. 그래서 inference 혹은 fine tunning시 전통적인 transformer로 귀환이 가능하다.

Code로 생각해보면.

Main\_stream\_self\_attention(Masked self attention decoder)

Model=nn.MultiheadAttention()

첫번째 layer만 생각하게 된다면

Model.forward(h2\_(0),src,src,src\_mask,src\_key\_pad\_mask)

여기에서 src는 위 그림에서 h0\_(0), h1\_(0), h2\_(0)임

Shape로 보게 되면, seq len, batch size, d v

Predicting stream self attention Model.forward(s2\_(0), src\_1, src\_1,src\_1\_mask,src\_1\_key\_pad\_mask)

Src1\_1의 shape를 보게 되면, seq\_len, batch size, 2 \* d\_v

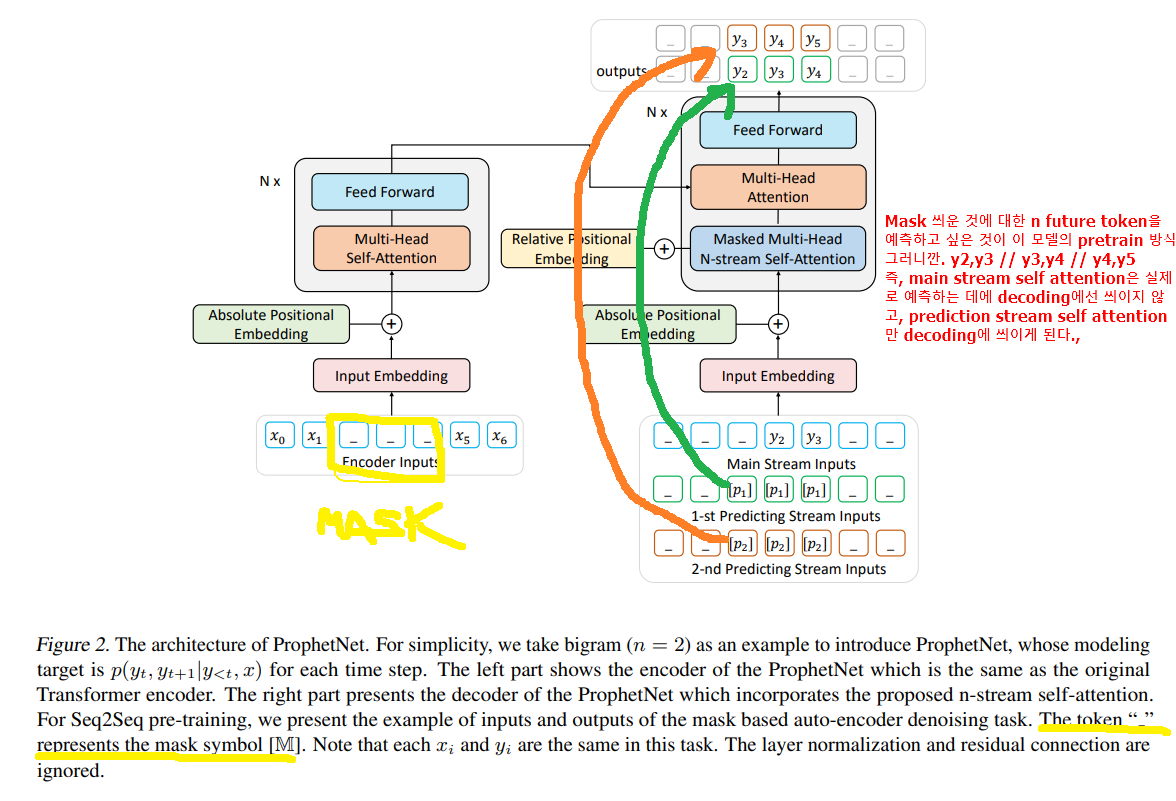
Src\_1은 h0\_(0):s0\_(0), h1\_(0):s1\_(0), h2\_(0):s2\_(0)즉 concat된 것

2.4) Positional Embedding

기존의 absolute positional embedding(BERT서 활용한 것)에 relative positional embedding(T5에서 활용한 것)을 추가로 같이 활용했다.

2.5) Seq2Seq Pre-training on Denoising Task

Paired 된 데이터를 확보하기 어렵기 때문에, 인위적으로 input의 특정 부분에 mask를 씌우고(denoised 되고) 이를 복원하는 식으로 모델을 학습시켰다. 방식으로는 token span masking을 활용함(MASS에서 썼다고 한다.) ProphetNet은 주어진 masked token span에서 the next n future tokens을 예측한다.[Figure 2로 다시 봐보자]



3. Experiments and Results

1) Pretraining

Layer : 12, Embedding dim : 1024, d\_ff : 4096

Batch size : 1024, training step : 500K

Optimizer Adam(lr=3e-4) , n stream self attention 에서 n을 2로만 상정.

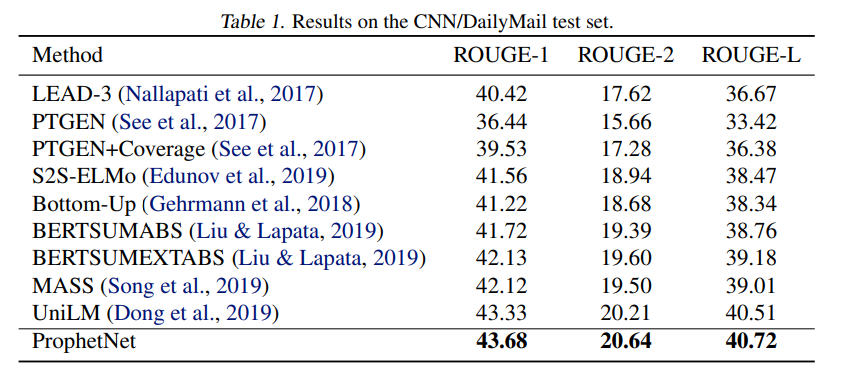
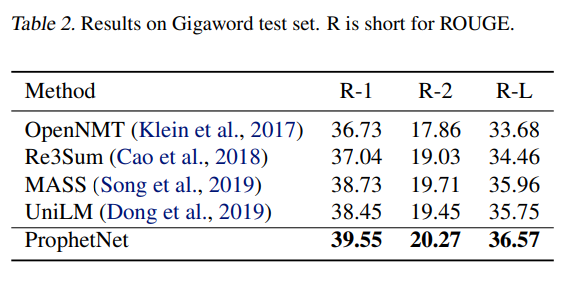
Data는 Book Corpus와 English Wikipedia(16GB)를 활용함.

Input length를 512로 하고, 64개의 token마다 randomly pick a starting position u 그리고 u로부터 연속적인 span에 mask를 씌운다.

“512를 64로 나눠서 8개의 chunk가 나오고, 이러한 8개 중에서 random하게 u(각 chunk의 시작점)을 뽑고, 그 chunck에 mask를 씌운다고 해석하면 될 듯.”

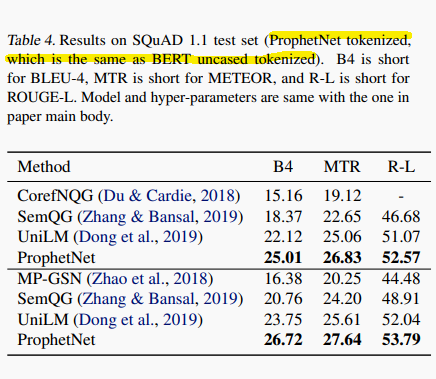
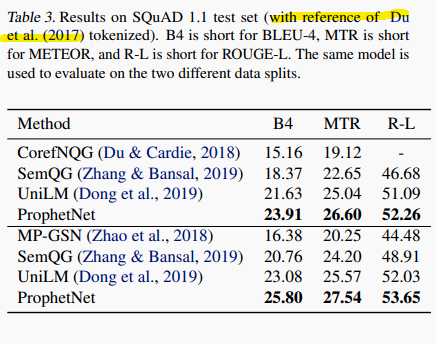
그 다음에 BERT와 동일하게 선택된 span에서 80프로는 [MASK]로 10프로는 Random한 token으로 10프로는 변화시키지 않는다. 그리고 computational cost 때문에 mask를 씌운 경우에만 cost를 계산하게 함(BERT와는 다른 경우지)

3.2) Text Summarization 실험 결과

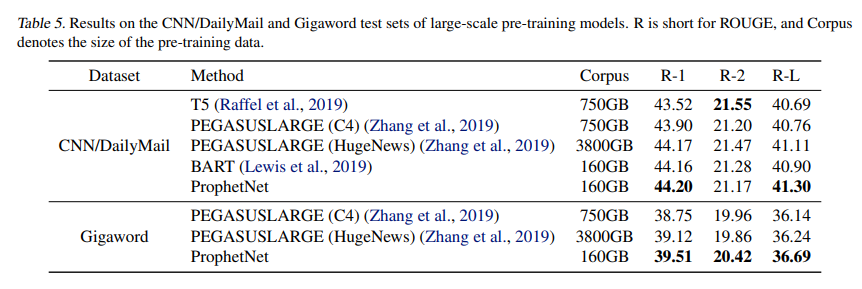


ROUGE에서 싹 다 SOTA를 달성함.

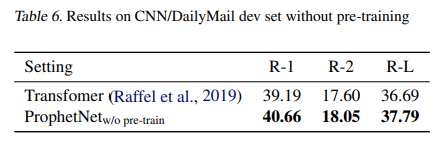
3.3) Question Generation



3.4) Large scale pretraining



3.5) ProphetNet without Pre-training



Pretraining하지 않고, training 할 때엔 future n gram prediction을 해주고, test시에는 original transformer로 예측을 한다 ( parameter를 공유하니 가능한 일로 사료됨)