

# Question Generation을 사용한 Question Answering

지도 교수님: 황승원

지도 조교님: 한호재

팀명: Information Retrievers

팀원: 김주찬, 이상현, 이준영

목차:

1. 해결한 문제
2. 기존 연구의 한계점
  - 2.1 기존대비 향상을 위한 방법론
3. 연구 소개
  - 3.1 관련 연구 조사
    - 3.1.1 Question Generation 모델
      - 3.1.1.1 GPT-2
      - 3.1.1.2 GPT-2 Copynet
      - 3.1.1.3 T5
    - 3.1.2 Question Answering 모델: BERT
    - 3.1.3 Relevance Matching 모델
      - 3.1.3.1 Pointwise BERT
      - 3.1.3.2 Pairwise BERT
  - 3.2 데이터셋
    - 3.2.1 SQuAD 데이터셋
    - 3.2.2 AddSent SQuAD 데이터셋
  4. 연구 및 실험 방법
    - 4.1 연구 개요
    - 4.2 Question Generation
      - 4.2.1 데이터분석 및 전처리
      - 4.2.2 모델 구현
    - 4.3 Relevance Matching
      - 4.3.1 데이터분석 및 전처리
      - 4.3.2 모델 구현
    - 4.4 BERT Question Answering 모델
  5. 실험 결과 및 분석
    - 5.1 추후 연구 계획
  6. 팀의 구성 및 팀원의 역할 배분

## 1. 해결한 문제

Information retrieving이란 사용자가 필요로 하는 정보를 수집하여 내용을 분석한 뒤 집합적인 정보로부터 찾고자 하는 내용과 관련이 있는 정보를 추출해 내는 것이다. Google이나 Naver와 같은 거의 모든 검색 엔진들은 사용자가 필요한 정보에 가장 관련성이 높은 정보를 찾아야 하기 때문에 이 information retrieval 기술을 사용한다. 사용자가 정보를 검색한다는 것은 다른 말로 사용자의 query와 가장 관련성이 높은 문서들을 찾는다는 것인데, 이때 찾아진 문서들은 유저 query와의 relevance 점수에 따라 순위가 매겨져 검색 목록에 나열이 되는 것이다.

우리는 information retrieval의 이 relevance ranking 방법에 추가적으로 question generation 기술을 question answering에 적용하여 adversarial 데이터에 대한 question answering 모델의 성능 향상을 목표로 하였다.

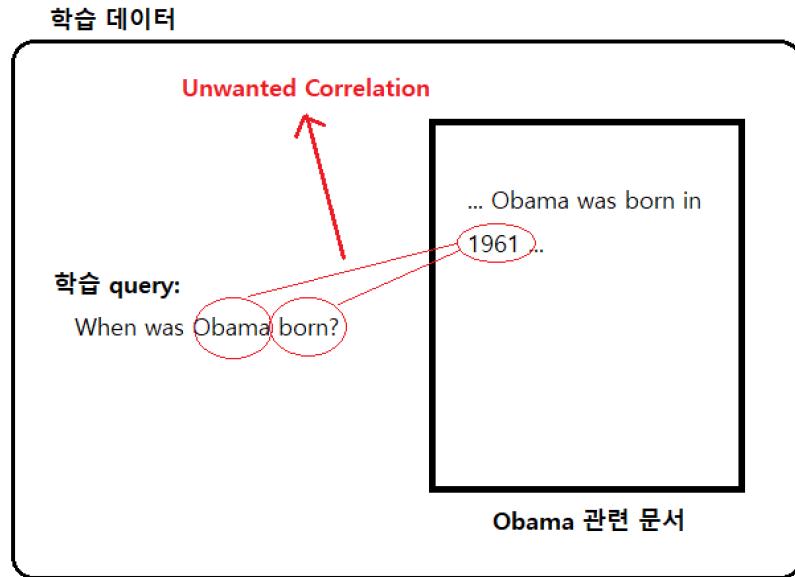
## 2. 기존 연구의 한계점

기존에 있는 연구들을 조사해 본 결과 기존 information retrieval 모델들에 한계점이 있다는 것을 알 수 있었다. 예를 들어 현재 진행되고 있는 연구들과 사용되고 있는 (“Document Expansion by Query Prediction”, Nogueira et al.) 모델들은 다음과 같은 방법으로 문서들의 순위를 매기게 된다:

일단 유저의 query와의 관련성을 측정할 문서, 그리고 그 문서를 사용해 생성한 추가적인 정보를 합하여 extended document를 만든다. 여기서 생성하는 추가적인 정보는 상황에 따라 해당 문서에 대한 질문이 되는 경우도 있으며 해당 문서에서 가장 중요한 부분을 추출한 문장인 경우도 있다. 그리고 나서는 유저의 query와 이렇게 생성된 extended document의 유사성을 relevance 점수를 매겨 순위를 정하는 것이다.

하지만 이러한 방법을 사용한 question answering 모델 학습은 유저의 query에 대한 답이 들어가 있는 문서를 사용하게 되고 이로 인해 필요가 없는 구체적인 정보와 overfitting을 초래할 수 있다는 것이 여기의 한계점이다.

아래 그림을 예로 본다면 이 경우에는 학습 데이터로 Obama의 출생년도를 묻는 query와 Obama 관련 문서가 들어간다. 이때 이대로 학습이 진행된다면 ‘Obama’라는 단어와 ‘1961’이라는 단어, 또는 ‘born’이라는 단어와 ‘1961’이라는 단어 등의 원치 않는 연관성이 학습되어 버릴 수 있다는 것이다. 이는 추후 이 모델을 사용하여 Obama의 임관년도에 대한



정보나 Obama가 아닌 다른 누군가의 출생년도에 대한 정보를 추출해야 할 때 전혀 필요가 없는 연관성이라고 볼 수 있다. 따라서 이러한 의미 없는 연관성이 학습되며 발생하는 overfitting 상황을 방지하기 위해 이 한계점을 해결할 필요가 있다.

추가적으로 기존에 있는 question answering 모델들의 경우, adversarial attack, 즉 의도적으로 모델의 성능을 떨어뜨리는 행위에 매우 취약하여 모델의 F1 스코어가 급격히 떨어지는 모습을 볼 수 있었다. 이는 노이즈와 오류들로 가득한 실제 웹상에서 치명적일 수 있는데 우리는 question generation에 사용되는 방식을 적용하여 이를 보완시키고자 이 실험을 진행하였다.

## 2.1 기존대비 향상을 위한 방법론

앞서 언급된 한계점을 해결하기 위해 우리는 유저의 query와 문서의 관련성을 측정하는 과정에서 차별점을 두고자 한다. 전체적인 과정은 총 4단계로 구성될 것이다:

1. 직접 fine tuning을 한 GPT-2를 사용하여 question generation 모델을 학습시킨다.
2. 이 question generation 모델의 테스트 데이터셋에 있는 문서들을 문장 단위로 나누어 앞서 학습된 question generation 모델로 각 문장마다 질문을 생성한다.
3. 직접 fine tuning을 한 BERT를 사용하여 relevance matching 모델을 학습시킨다.
4. 앞서 학습된 relevance matching 모델을 사용하여 유저의 query와 문서들의 연관성을 측정하는데 이때 문서 자체가 아닌 해당 문서로 생성한 질문들을 비교하여 각 질문마다 (한 문서에 여러개씩) relevance 점수를 측정한다. 이렇게 생성된 relevance 점수들은 최종적으로 document 자체의 순위를 매기는 데에 사용이 된다.

이 방법을 통해 관련성을 하게 된다면 기존 연구들과 달리 구체적인 정보를 포함한 기존 문서들을 relevance matching 과정에서 제외하게 된다. 이렇게 문서의 문장들을 사용해 생성된 query들만을 가지고 관련성을 측정하여 기존에 있었던 overfitting 문제를 해결한다는 데에 있어 기존 연구들과 차별을 둔다.

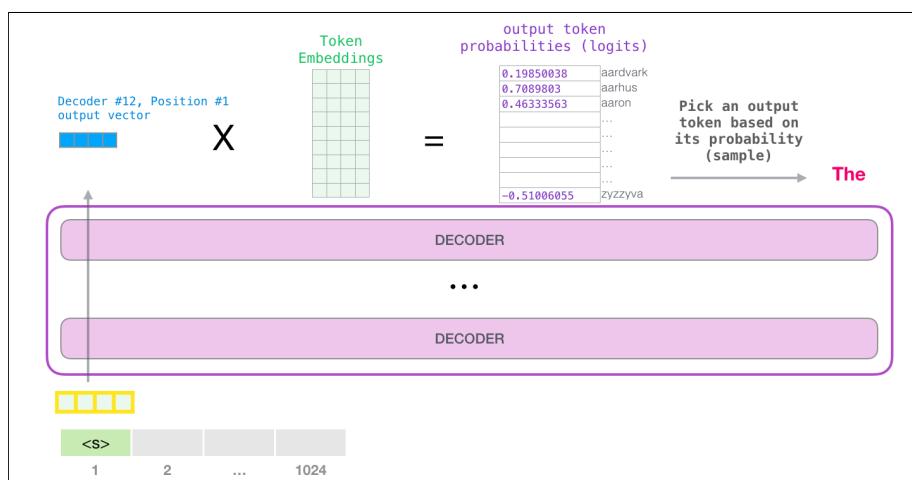
### 3. 연구 소개

#### 3.1 관련 연구 조사

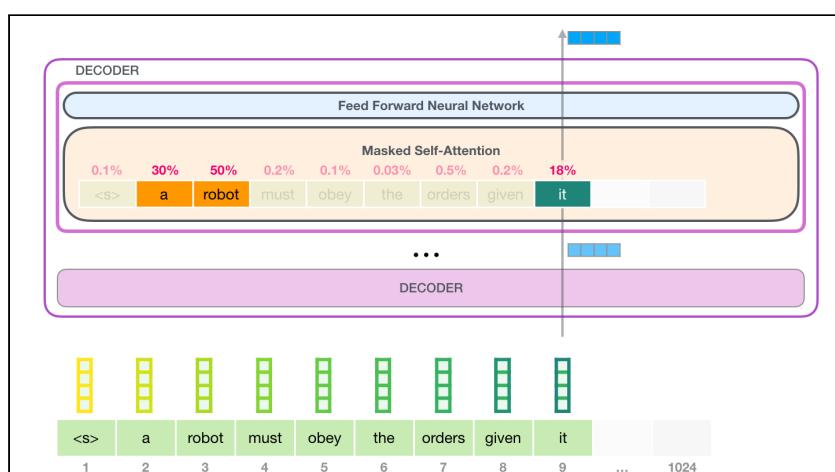
##### 3.1.1 Question Generation 모델

###### 3.1.1.1 GPT-2

GPT-2는 ‘Generative Pre-Training 2’라는 모델로 단방향 언어모델로 decoder로만 층을 쌓은 구조를 가지는 모델이다. GPT-2는 단방향 언어모델이기에 입력값으로 하나의 토큰만을 받게 된다. 이 토큰은 token embedding과 positional encoding을 합한 새로운 vector를 만들어 입력값으로 넣게 된다.



GPT-2의 작동 방식을 위의 그림을 예시로 들면, 먼저, <s>라는 start token이 모델의 입력으로 들어가게 되고, 모델의 여러 decoder layer를 지나면서 position 1에 대한 output vector를 얻게 된다. 이를 token embedding과 곱해서 가지고 있던 단어들에 대한 다음으로 나올 수 있는 확률을 구하고, 이 중에서 가장 확률이 높은 단어를 선택하게 되는 것이다. 이

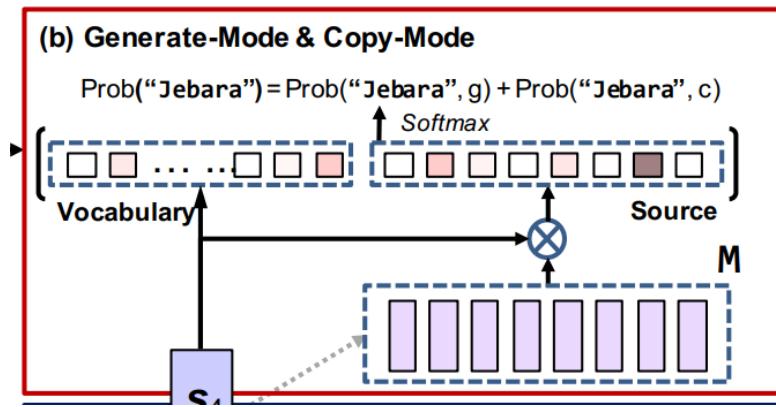


예시에서는 ‘The’가 선택되었음을 볼 수 있다. 그리고 ‘The’는 다음 position의 입력 token으로 들어가게 되어 이를 문장이 끝날 때 까지 반복하게 된다.

이 모델에서 사용된 Decoder는 masked self-attention layer 위에 feed forward neural network layer가 쌓여 있는 구조이다. masked self-attention layer에서 지금 까지 들어왔던 입력들에 대한 attention 가중치를 계산하게 되고, 이를 이용하여 feed forward neural network에서 계산을 하게 되는 것이다.

### 3.1.1.2 GPT-2 Copynet

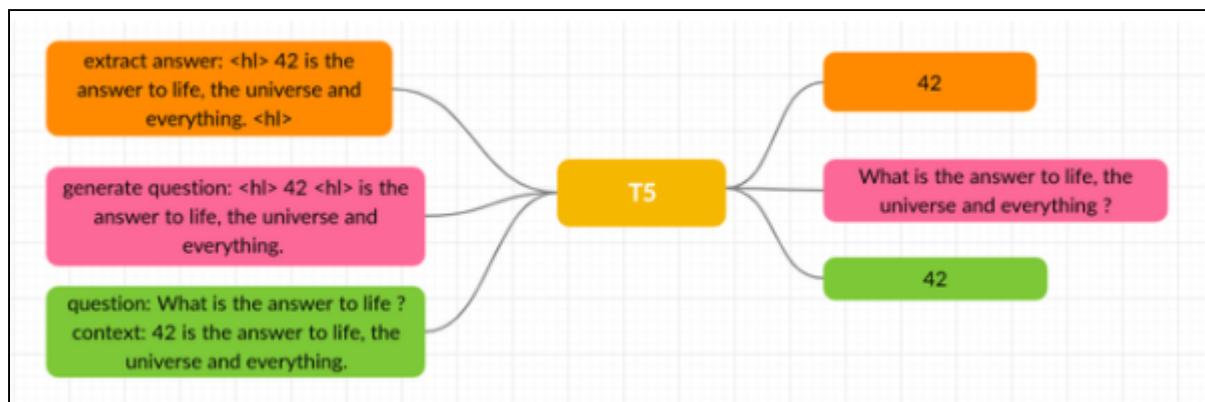
GPT-2 Copynet은 GPT-2 모델에 Copynet technique을 접목시켜서 만든 모델이다.



Copynet은 ‘Incorporating Copying Mechanism in Sequence-to-Sequence Learning’이라는 모델에서 제안된 기술로, 기존 언어 모델들이 문장을 생성할 때, 기존 학습을 통해 만들어둔 단어의 확률에 따라 예측을 하는 것이 아닌, context로 주어진 질문 문장에 있는 단어의 확률에 따라 위의 사진처럼 question을 생성할 때 같이 더해주어 고려하는 방식이다. 이 기술을 이용하면 질문이 context와 좀 더 연관되어 질문의 퀄리티의 향상이 있을 수 있다.

### 3.1.1.3 T5

T5 question generation 모델의 경우에는 GPT-2를 대신하며 더 성능이 좋은 question generation 모델을 사용하기 위해 찾은 모델이다.



여기에서 사용된 T5모델은 multitask QA-QG를 사용하여 학습되었다. T5는 세부적으로 3가지 task를 수행하는 모델로 나뉘어 있다. 첫번째 모델은 context에서 answer like span을 뽑아내는 모델이다. 두번째 모델은 해당 answer like span에 대하여 question generation을 수행하는 모델이다. 마지막 모델은 해당 generate된 question을 이용하여 답변을 생성하는 QA모델이다. 위의 T5모델은 multi-task로 3가지 tsak에 대한 작업을 동시에 수행하며 fine tuned 된다.

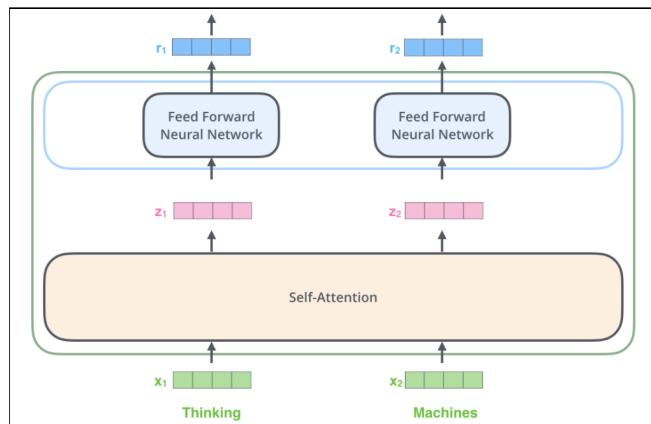
Question generation을 위해 이 모델은 우선 정답과 가장 일치할 것 같은 answer like span을 다수 선택하여 각각에 대한 질문을 생성하는데 이때 어떤 질문을 최종적으로 output할지를 결정하는 방식에 따라 Random, Max BM25, 그리고 Max F1방식으로 나누었다.

Random 방식은 말 그대로 이 여러개의 질문 후보 중 임의의 질문을 뽑아 output으로 정하는 방식이며 Max BM25와 Max F1 방식들은 이 후보 중에서 각각 가장 높은 BM25 또는 F1 스코어를 갖고 있는 질문들을 선택하는 방식들이다. 이때 BM25와 F1스코어를 매기기 위한 비교 대상은 원본 질문이다.

이 연구에서 사용한 T5 모델은 이미 학습된 모델을 huggingface에서 가져온 모델이며 앞서 언급된 3가지 방식들이 모두 사용되었다.

### 3.1.2 Question Answering 모델: BERT

BERT는 ‘Bidirectional Encoder Representations from Transformer’로 이름에서 알 수 있듯이, transformer encoder들이 연결되어 있는 구조의 모델이다.

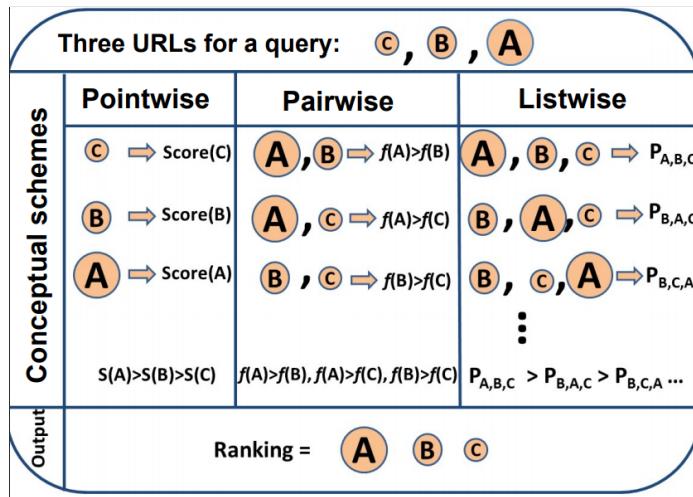


이 encoder들은 단어들의 모음(sequentially)을 인풋으로 받게 된다. 이 단어들을 그냥 넣는 것이 아닌, 512의 크기를 가진 word embedding vector로 변환되어 입력값으로 넣는 방식을 사용한다.

BERT의 작동 방식을 설명하면, input이 들어오면, 먼저 self-attention layer에서 입력 vector에 대해 queries, keys 그리고 values vector를 만들게 된다. 만들어진 queries vector와 keys는 vector의 곱으로 score가 구해진 뒤 8로 나뉘어 지게 된다. (실제로는  $\sqrt{D_K}$ 를 넣지만 기본값은 8이다). 이 과정을 통해 나온 값을 softmax function에 넣어 나오는 값을 value

vector와 곱해준다. 마지막으로 계산되어 나온 value vector를 더해주게 되면 self-attention layer의 출력값이 된다. 나온 vector는 다음 층인 feed forward neural network의 입력으로 들어가게 되며, neural network에서 나온 결과가 다음 encoder층의 입력으로 들어가게 되는 과정을 반복하게 되는 것이다. 이런 방식으로 query가 들어오면 그에 대한 답을 찾는 것이 BERT의 question answering 모델이다.

### 3.1.3 Relevance Matching 모델



#### 3.1.3.1 Pointwise BERT

BERT의 기본적인 Pointwise 모델의 경우 loss function에서 한번에 하나의 문서에 대해서만 고려를 하는 방식이다. 따라서 각 문서에 대한 점수는 다른 문서와는 독립적이게 된다.

#### 3.1.3.2 Pairwise BERT

Pairwise 모델의 경우에는 loss function에서 한번에 문서 한 쌍을 고려하며 그 한 쌍에 대한 상대적인 순위를 계산하는 방식이다. Pointwise의 class label이나 relevance 점수로 예측하는 것보다, 두 문서 사이의 상대적 순위를 비교하여 계산하는 것이 훨씬 더 자연스러운 방식이고, 실제로도 pairwise를 이용한 경우가 pointwise를 이용하는 경우에 비해 성능이 높게 나오게 된다.

## 3.2 데이터셋

### 3.2.1 SQuAD 데이터셋

사용한 데이터는 스쿼드(SQuAD, Stanford Question Answering Dataset) 데이터셋으로 질문과 질문에 대한 답이 존재하는 document(context), 그리고 그에 대한 대답으로 이루어져 있다. 데이터셋의 구성의 아래의 사진과 같다.

index	question	context	text	answer_start	c_id
56be85543aeeee14008c9063	When did Beyonce start becoming popular?	Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ b... in the late 1990s		269.0	0
56be85543aeeee14008c9065	What areas did Beyonce compete in when she was...	Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ b...	singing and dancing	207.0	0
56be85543aeeee14008c9066	When did Beyonce leave Destiny's Child and bec...	Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ b...	2003	526.0	0

각 항목에 대한 설명은 다음과 같다: context는 문서를 뜻하고 context에 대한 질문은 question이며 text는 이 question에 대한 context 내의 답변(정답)이다. answer\_start는 context내의 정답의 위치를 나타내고 c\_id는 context의 번호이다. 하나의 context에는 여러개의 질문이 존재하며 각 질문에 대한 답이 포함되어 있다.

### 3.2.2 AddSent SQuAD 데이터셋

AddSent SQuAD 데이터셋의 경우에는 ‘Adversarial Examples for Evaluating Reading Comprehension Systems’라는 논문에서 제안된 것으로, 위에서 언급한 SQuAD 데이터셋의 context에 adversarial attack sentence가 추가된 데이터셋으로 모델이 adversarial attack에 잘 대응 할 수 있는지를 평가하기 위해 만들어진 데이터셋이다.

**Article:** Super Bowl 50

**Paragraph:** *Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*

**Question:** *“What is the name of the quarterback who was 38 in Super Bowl XXXIII?”*

**Original Prediction:** John Elway

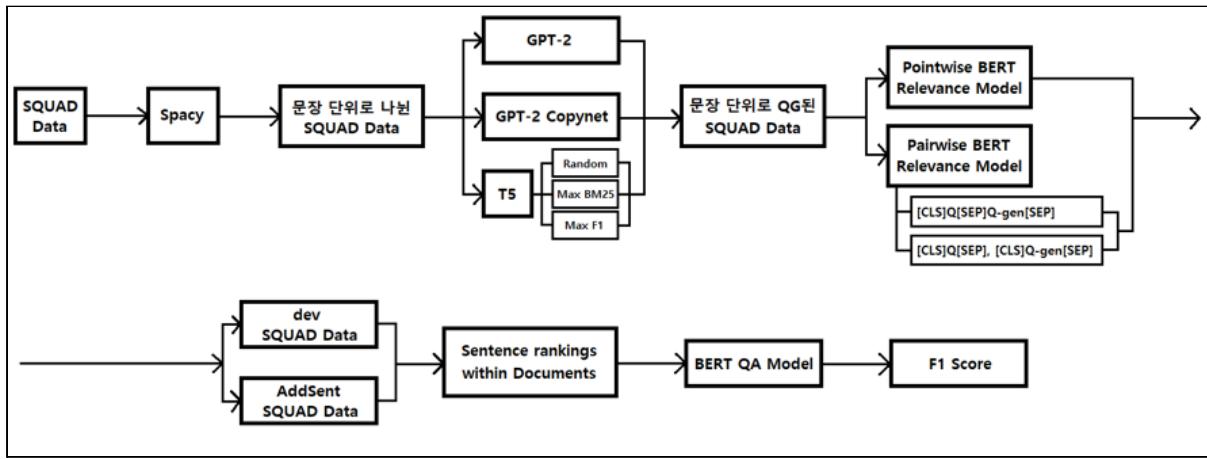
**Prediction under adversary:** Jeff Dean

위의 사진에서 볼 수 있듯이 파란색으로 된 adversarial sentence가 추가되었고, 질문에 대해 제대로된 예측의 경우 John Elway여야 했지만, adversarial sentence에 영향을 받은 경우에는 Jeff Dean이라고 정답을 예측하게 되는 것이다.

## 4. 연구 및 실험 방법

### 4.1 연구 개요

우리가 실험을 진행했던 과정과 실험 방법들의 전체적인 개요는 아래 그림과 같다.



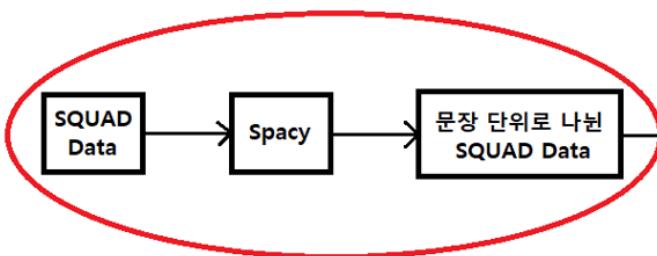
화살표 방향대로 SQuAD Data로부터 시작하여 전처리 과정을 거쳐 문장 단위로 나눈 SQuAD Data를 만든다. 이후 question generation 과정을 거쳐 각 문장들에 대한 질문을 생성한다.

다음으로 QG된 SQuAD 데이터셋을 이용하여 BERT relevance matching 모델을 학습시키는데 이 학습시킨 모델을 이용하여 일반데이터와 adversarial데이터에 각각 적용하여 rank를 매긴 뒤 rank가 가장 높았던 문장을 question answering 모델에 넣어 F1스코어를 측정한다.

## 4.2 Question Generation

### 4.2.1 데이터분석 및 전처리

여기서부터는 실험의 가장 처음 부분인 question generation 데이터 전처리부터 시작하여 실험 과정을 부분별로 설명한 것이다.



가장 먼저 데이터를 문장단위로 나누기 위하여 우리는 Spacy 라이브러리에서 제공하는 **sentence segmenter** 기능을 사용하였다. 이를 SQuAD dataset의 **context**에 적용시켜 문장단위의 데이터를 얻을 수 있었다.

여기서 연구에 사용될 question generation 모델을 학습시키기 위해서는 데이터셋의 **context**에서 질문에 대한 정답이 있는 문장만을 뽑아내는 작업이 필요하다. 데이터셋에서 **answer\_start**는 **context**내의 정답(데이터의 **text column**)의 위치 말해주는데 이를 이용하여

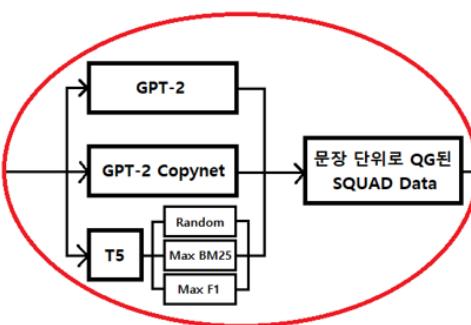
데이터를 question과 context내의 정답이 있는 문장의 쌍으로 만들어 주었다. 이를 적용한 데이터의 구성은 아래와 같다:

question	context	text	answer_start	c_id	short_text
When did Beyonce start becoming popular?	Beyoncé Giselle Knowles-Carter (/biːˈjɒnsə/ b...)	in the late 1990s	269.0	0	Born and raised in Houston, Texas, she perform...
What areas did Beyonce compete in when she was...	Beyoncé Giselle Knowles-Carter (/biːˈjɒnsə/ b...)	singing and dancing	207.0	0	Born and raised in Houston, Texas, she perform...
When did Beyonce leave Destiny's Child and bec...	Beyoncé Giselle Knowles-Carter (/biːˈjɒnsə/ b...)	2003	526.0	0	Their hiatus saw the release of Beyoncé's debu...

여기서 short\_text는 정답이 존재하는 문장만을 Spacy를 이용하여 가져온 것이다. 이후 question generation model은 short\_text를 통하여 question을 예측하도록 학습을 시킬 것이다.

## 4.2.2 모델 구현

Question generation 모델의 경우 앞서 언급하였듯이 3가지의 모델을 사용해 보았다. 이때 사용된 모델들은 GPT-2, Copynet논문을 GPT-2에 적용한 GPT-2 Copynet, 그리고 T5 모델이다. (자세한 설명은 앞의 관련 연구 조사를 참고)



Copynet의 경우에는 입력된 Source 토큰들의 확률 분포를 계산하고 이를 generation할 단어 확률 분포 값에 더해주는 방식을 사용하여 결과값에서 source 토큰들이 더 자주 나타날 수 있게 해준다.

## 4.3 Relevance Matching

### 4.3.1 데이터 분석 및 전처리

다음으로는 BERT를 학습시키기 위한 데이터를 만드는 과정이다. BERT를 위한 데이터를 만들기 위해서는 위에서 언급한 **question generation** 모델로 생성된 질문들이 필요하다. context를 Spacy를 이용하여 문장으로 나눈 뒤 각 문장에 **question generation** 모델을 사용하여 질문들을 생성하였다. 아래가 그 결과이다.

question	context	text	answer_start	c_id	q_id	sent_list	gen_list	sid_list	stlist	edlist	anwloc
Which instruments can Madonna play?	Besides singing Madonna has the ability to pla...	drum and guitar	97.0	9450	0	Madonna later played guitar on her demo record...	What type of instrument did Madonna play on th...	3	319	370	0
Which instruments can Madonna play?	Besides singing Madonna has the ability to pla...	drum and guitar	97.0	9450	0	On the liner notes of Pre-Madonna, Stephen Bra...	What kind of life did Jay Z think needed to pr...	4	371	503	0
Which instruments can Madonna play?	Besides singing Madonna has the ability to pla...	drum and guitar	97.0	9450	0	After her career breakthrough, Madonna focused...	What year did Madonna release her hit song "Li...	5	504	669	0

여기서 **sent\_list**은 spacy로 나눈 context에 대한 문장이며 **gen\_list**은 **snet\_list**를 학습된 **question generation model**에 넣어 생성된 질문들이다. **stlist**과 **edlist**은 전체 document 내에서 **sent\_list**가 위치하는 시작점과 끝점을 의미한다. **anwloc**은 **sent\_list**가 질문에 대한 정답을 가지고 있는 문장이면 1, 그렇지 않은 문장이면 0을 나타낸다.

이때 **pointwise**한 방법으로 학습을 하기 위해서 BERT에 **question**과 **gen\_list**을 입력으로 준 뒤 **anwloc**이 1이면 **relevance score**가 높아지도록 학습시키고, **anwloc**이 0이면 **relevance score**가 0에 가까워 지도록 학습시켰다. 즉 원래 **question**과 정답이 있는 문장에서 생성된 질문들 간의 연관성이 높다고 판단하도록 학습하는 것이다.

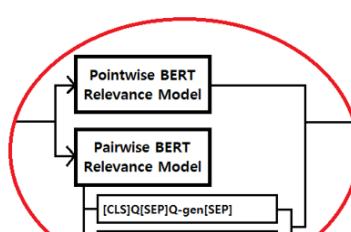
**Pairwise** 학습방법을 사용하려면 정답이 있는 문장에서 생성된 질문과 정답이 없는 문장에서 생성된 질문이 학습때 같이 들어와야 한다. 이를 위하여 데이터를 아래와 같이

question	context	text	answer_start	c_id	q_id	sent_list	gen_list_x	sid_list	stlist	edlist	anwloc	gen_list_y
Which instruments can Madonna play?	Besides singing Madonna has the ability to pla...	drum and guitar	97.0	9450	0	On the liner notes of Pre-Madonna, Stephen Bra...	What kind of life did Jay Z think needed to pr...	4	371	503	0	Where did Victoria start playing drum and guitar?
Which instruments can Madonna play?	Besides singing Madonna has the ability to pla...	drum and guitar	97.0	9450	0	After her career breakthrough, Madonna focused...	What year did Madonna release her hit song "Li..."	5	504	669	0	Where did Victoria start playing drum and guitar?

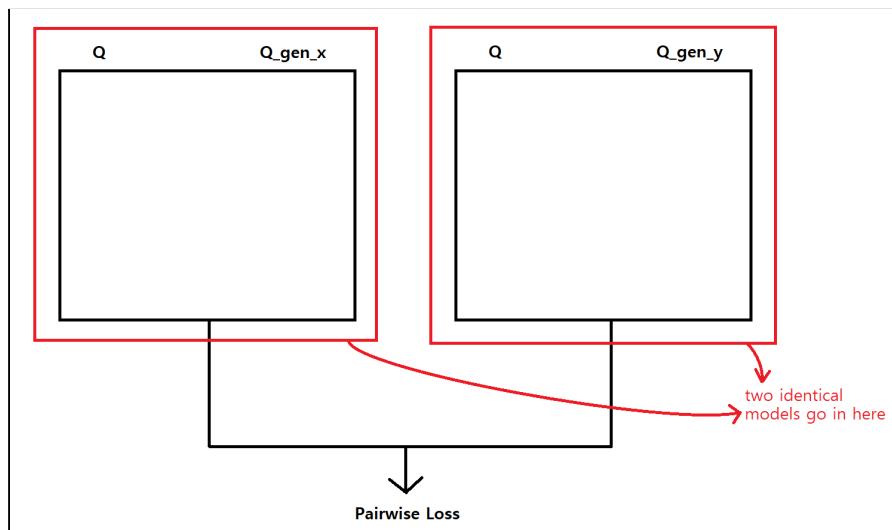
구성하였다.

**gent\_list\_x**는 **question**에 대한 정답이 없는 문장에서 GPT-2를 이용하여 **generate**된 **question**이다. **gen\_list\_y**는 **question**에 대한 정답이 있는 문장에서 GPT-2를 이용하여 생성된 문장이다. 이후 모델에 의하여 둘의 상대적 순위(score)에 차이가 생기게 될 것이다.

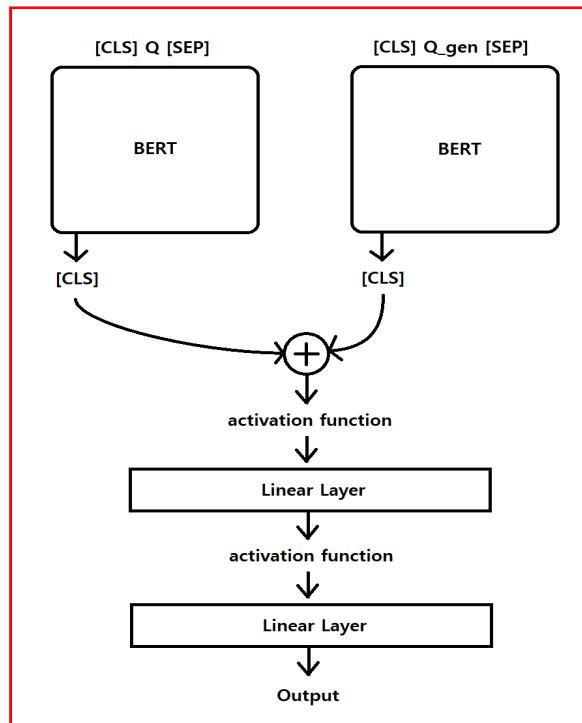
#### 4.3.2 모델구현



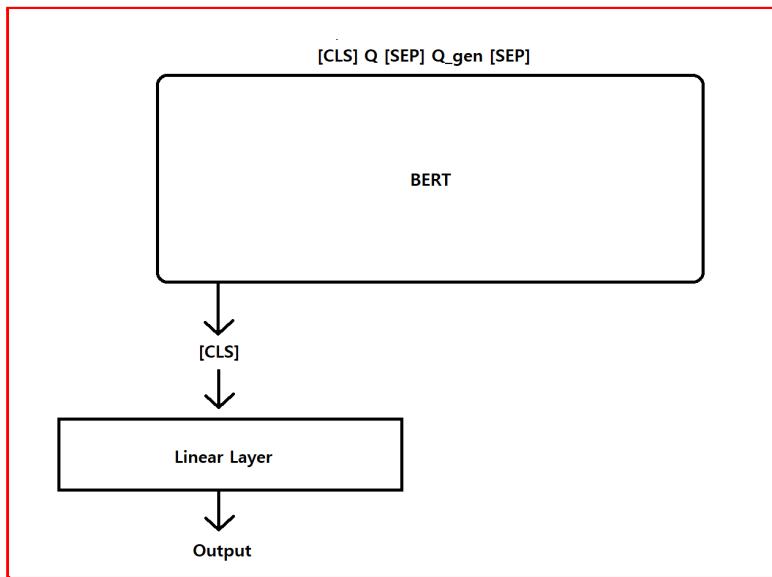
Pairwise 를 적용한 BERT의 경우에는 2가지 모델을 구성하여 실험을 진행하였다.



위 그림이 실험한 모델의 전체적인 틀로 붉은색 상자 안에는 2가지 종류의 모델을 넣어서 실험을 할 예정이다.  $Q_{gen\_x}$ 는 정답이 없는 문장에서 생성된 질문을,  $Q_{gen\_y}$ 는 정답이 있는 문장에서 생성된 질문을 뜻한다. 붉은색 상자 안에서 출력된 값들은 pairwise loss를 사용하여 loss가 계산된다.

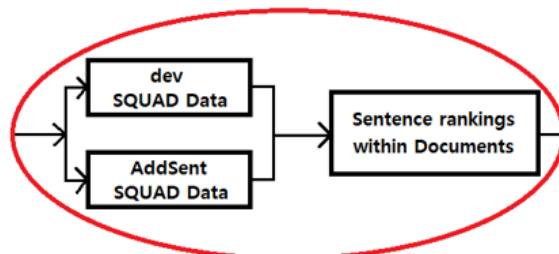


먼저 첫번째로 테스트한 모델은 위의 그림과 같다. BERT에서 각각 나온 CLS 토큰의 vector를 concatenate시킨뒤 linear 와 activation function을 2번 통과시킨후 나온 output을 사용한다.



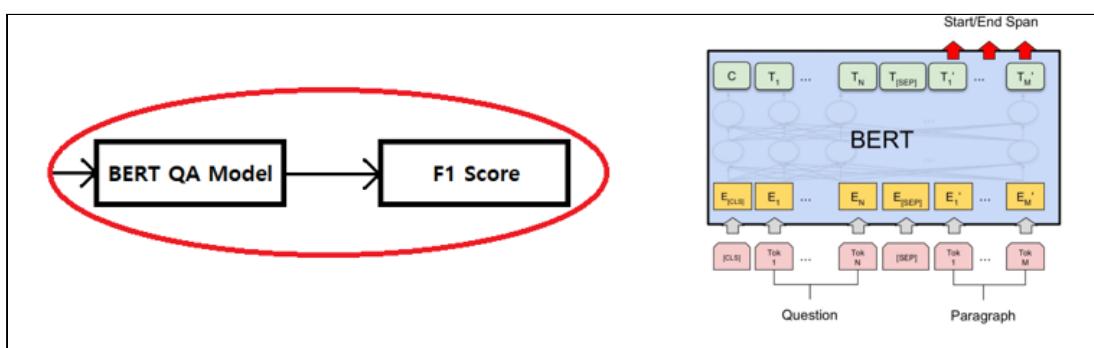
두번째로 테스트한 모델은 위의 그림과 같다. 여기서는 Q와 Q\_gen이 SEP토큰과 같이 들어오며 CLSToken에서 나온 vector을 하나의 linear 레이어를 통과시켜서 바로 output을 내보낸다.

위의 모델을 문장 단위로 question generation이 된 dev SQuAD Data, AddSent SQuAD 데이터셋에 적용시켜 각각의 문장에 대하여 rank를 매겼다.



#### 4.4 BERT Question Answering 모델

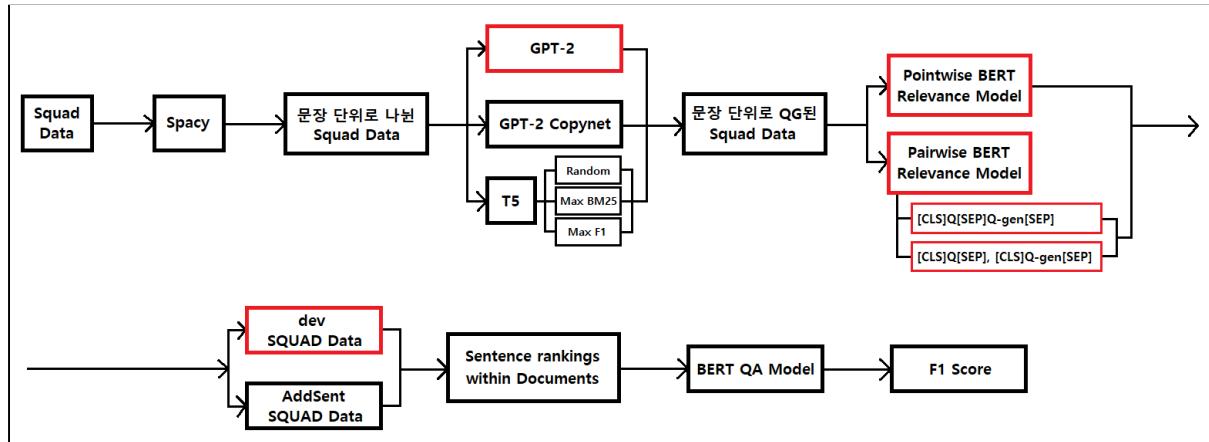
Question Answering 모델의 경우 SQuAD 데이터셋 자체가 QA를 위한 데이터셋이기 때문에 따로 학습하기 위한 데이터 전처리가 필요하지 않았다.



학습된 BERT question answering 모델은 이전에 relevance matching에서 rank가 가장 높았던 문장의 원본 문장에 대하여 question answering를 수행한 뒤 F1 Score를 측정하는데 사용되었다.

## 5. 실험 결과 및 분석

가장 먼저 진행을 했던 실험 과정을 표시한 연구 개요는 다음과 같다.



우선 기본적인 GPT-2를 이용하여 실험을 하였으며 아직 BERT relevance matching 모델들 중 어떤 모델이 가장 좋은 성능을 갖고 있는지 모르는 상태였기 때문에 GPT-2 question generation 모델을 사용하여 생성된 문장들을 pointwise와 2개의 pairwise 방식 모두에 각각 적용해보며 relevance 모델을 학습시켰다. 그 다음으로는 test 데이터에 ranking을 측정한 뒤, 정답이 있는 문장을 찾아낸 비율을 측정하였다.

이에 대한 결과는 다음과 같다.

Rank results using a model trained with <b>normal pointwise BERT training</b> .	<table border="1"> <tbody> <tr><td>rank1</td><td>0.51</td></tr> <tr><td>rank2</td><td>0.708</td></tr> <tr><td>rank3</td><td>0.834</td></tr> <tr><td>rank4</td><td>0.887</td></tr> </tbody> </table>	rank1	0.51	rank2	0.708	rank3	0.834	rank4	0.887
rank1	0.51								
rank2	0.708								
rank3	0.834								
rank4	0.887								
Rank results after training the model in the form <b>[CLS] Q [SEP] Q_gen [SEP]</b> with <b>pairwise training</b> .	<table border="1"> <tbody> <tr><td>rank1</td><td>0.541</td></tr> <tr><td>rank2</td><td>0.739</td></tr> <tr><td>rank3</td><td>0.858</td></tr> <tr><td>rank4</td><td>0.905</td></tr> </tbody> </table> <div style="text-align: right;"> </div>	rank1	0.541	rank2	0.739	rank3	0.858	rank4	0.905
rank1	0.541								
rank2	0.739								
rank3	0.858								
rank4	0.905								
Rank results after training the model in the form <b>[CLS] Q [SEP], [CLS] Q_gen [SEP]</b> with <b>pairwise training</b> .	<table border="1"> <tbody> <tr><td>rank1</td><td>0.221</td></tr> <tr><td>rank2</td><td>0.436</td></tr> <tr><td>rank3</td><td>0.624</td></tr> <tr><td>rank4</td><td>0.851</td></tr> </tbody> </table> <div style="text-align: right;"> </div>	rank1	0.221	rank2	0.436	rank3	0.624	rank4	0.851
rank1	0.221								
rank2	0.436								
rank3	0.624								
rank4	0.851								

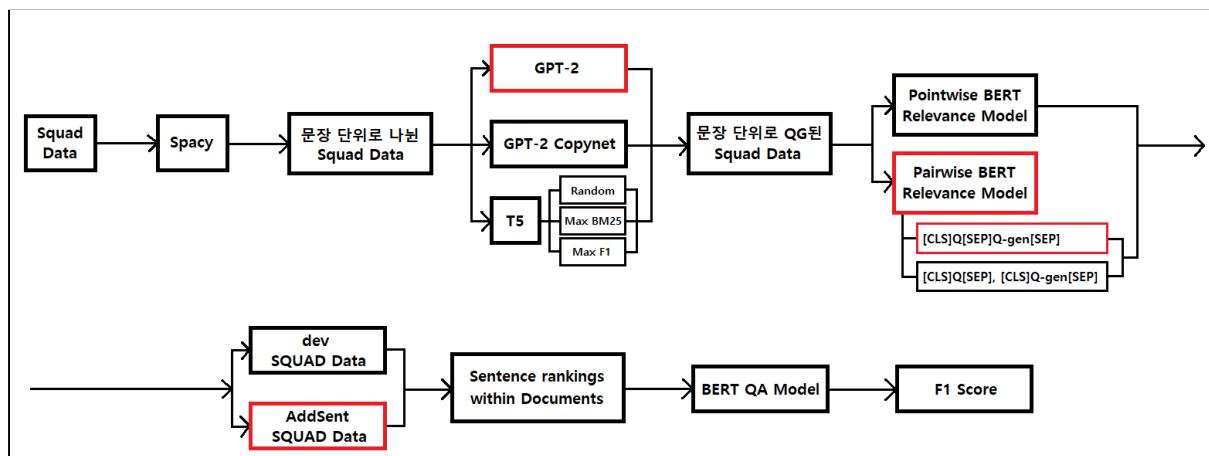
여기서 rank가 의미하는 것은 document 내의 각 문장 별로 score를 매겨 높은 순으로 순위를 매기는데, 이때 가장 높은 순위의 문장에 찾고자 하는 답이 들어 있다면 rank1, 1순위 또는 2순위에 있다면 rank2, 등등 이런 방식으로 측정을 한 결과이다.

먼저 첫 번째 결과는 일반적으로 사용되는 pointwise BERT를 사용한 결과이고, 두 번째 결과는 pairwise 모델 중에서도 Q와 Q\_gen이 함께 들어가는 모델을 사용한 결과이다. Pairwais로 학습을 시킨 경우, pointwise 방식에 비해 성능의 향상이 있음을 알 수 있는데 이는 앞서 언급한 대로 한 번에 query를 두 개씩 고려하여 상대적인 순위를 계산하는 pairwise의 특성이 여기서는 더 유리하기 때문이다. 세 번째 결과는 또 다른 pairwise 모델이었던 Q와 Q\_gen을 따로 받는 모델의 결과인데 앞선 모델들보다 성능이 좋지 않다는 것을 볼수 있었다. 이에 추후 진행된 실험들에서는 성능이 가장 좋았던 Q와 Q\_gen을 함께 받는 pairwise BERT 모델을 사용하게 되었다.

Rank results using the <b>original questions</b> and the <b>model trained with</b> pairwise training using <b>questions generated</b> with GPT-2 from document sentences.	<table border="1"> <tbody> <tr><td>rank1</td><td>0.541</td></tr> <tr><td>rank2</td><td>0.739</td></tr> <tr><td>rank3</td><td>0.858</td></tr> <tr><td>rank4</td><td>0.905</td></tr> </tbody> </table>	rank1	0.541	rank2	0.739	rank3	0.858	rank4	0.905
rank1	0.541								
rank2	0.739								
rank3	0.858								
rank4	0.905								
Rank results using the <b>original questions</b> and the <b>model trained with</b> pairwise training using the <b>original document sentences</b> .	<table border="1"> <tbody> <tr><td>rank1</td><td>0.837</td></tr> <tr><td>rank2</td><td>0.935</td></tr> <tr><td>rank3</td><td>0.965</td></tr> <tr><td>rank4</td><td>0.977</td></tr> </tbody> </table>	rank1	0.837	rank2	0.935	rank3	0.965	rank4	0.977
rank1	0.837								
rank2	0.935								
rank3	0.965								
rank4	0.977								

위 2개의 표들은 해당 pairwise 모델을 사용한 결과들이다. 첫 번째 표는 GPT-2로 생성한 question들을 사용하여 BERT를 학습시킨 뒤, 이 BERT로 ranking을 한 결과이며 그 아래의 표는 원본 sentence들을 사용하여 BERT를 학습시킨 뒤, ranking을 한 결과이다. GPT-2를 사용하지 않은 결과가 더 좋은 것을 알 수 있었다.

다음으로는 기존 연구에서의 한계점 중 하나였던 adversarial attack에서 우리가 구현한



방식의 성능을 측정해보기 위해 위 그림과 같이 실험을 진행하였다. Question generation 모델은 그대로 GPT-2를 사용하였으나, relevance matching 모델의 경우 이전 연구에서 가장

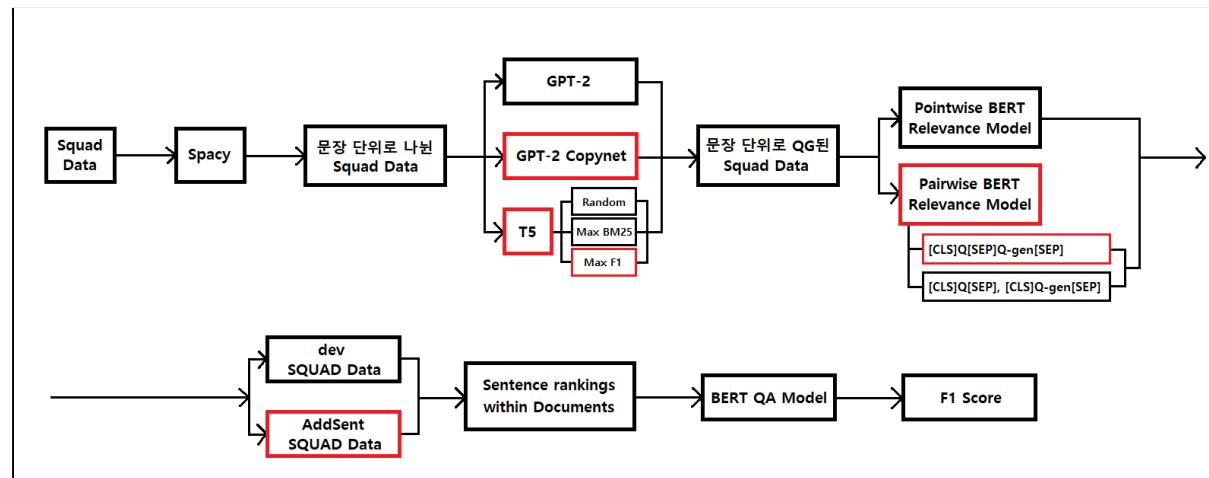
성능이 좋았던 pairwise BERT relevance matching model을 채택하기로 하였고 데이터셋의 경우에는 AddSent adversarial SQuAD 데이터셋을 사용하였다.

이에 대한 결과는 다음과 같다.

Rank results of <b>AddSent dataset</b> using model trained by pairwise BERT on <b>questions generated</b> by GPT-2.	<table border="1"><tr><td>rank1</td><td>0.356</td></tr><tr><td>rank2</td><td>0.67</td></tr><tr><td>rank3</td><td>0.817</td></tr><tr><td>rank4</td><td>0.904</td></tr></table>	rank1	0.356	rank2	0.67	rank3	0.817	rank4	0.904
rank1	0.356								
rank2	0.67								
rank3	0.817								
rank4	0.904								
Rank results of <b>AddSent dataset</b> using model trained by pairwise BERT on <b>original sentences</b> .	<table border="1"><tr><td>rank1</td><td>0.477</td></tr><tr><td>rank2</td><td>0.879</td></tr><tr><td>rank3</td><td>0.936</td></tr><tr><td>rank4</td><td>0.964</td></tr></table>	rank1	0.477	rank2	0.879	rank3	0.936	rank4	0.964
rank1	0.477								
rank2	0.879								
rank3	0.936								
rank4	0.964								

첫 번째 표가 adversarial SQuAD 데이터셋을 GPT-2를 사용하여 생성한 질문들로 학습한 BERT모델에 적용한 결과이며 그 밑은 같은 데이터를 원본 sentence 그대로 넣어 BERT모델에 적용한 결과이다. 원본 sentence를 그대로 사용하는 것이 GPT-2로 question generation을 수행하여 사용하는 것보다 성능이 더 좋은 것을 알 수 있었다.

여러번의 실험 끝에 우리는 question generation 모델의 성능이 전체적인 실험 결과의 병목현상을 일으키는 원인이라고 판단하였다. 이에 따라 다음 실험으로는 이 question generation 모델의 성능을 향상시키는 데에 목표를 두게 되었고 GPT-2를 대신하여 GPT-2의 향상 된 버전인 GPT-2 CopyNet과 또 다른 question generation 모델인 T5모델을 각각 사용해 보며 진행하였다.



다음은 기존의 GPT-2와 CopyNet 모델을 사용하여 실험을 진행한 결과들이다.

Rank results using **GPT-2**

rank1	0.356
rank2	0.67
rank3	0.817
rank4	0.904

Rank results using **Copynet\_0.01**

rank1	0.356
rank2	0.67
rank3	0.813
rank4	0.895

Rank results using **Copynet\_0.1**

rank1	0.369
rank2	0.692
rank3	0.84
rank4	0.918

첫 번째 결과가 일반적인 GPT-2를 사용한 결과이며 GPT-2 CopyNet의 경우 모델을 구현할 때 조절할 수 있는 파라미터의 확률을 높게 잡을 경우 질문이 잘 생성되지 않는 경우가 발생해 이 값을 0.01이나 0.1로 바꾸어 가며 실험을 진행하였다. 두 번째와 세 번째 표가 각각 이 값을 0.01과 0.1로 잡았을 때의 결과이다. CopyNet\_0.1의 경우 기존 GPT-2를 사용한 것 보다 약간의 성능향상이 있는 것을 볼 수 있었다.

Rank results using **T5\_Random**

rank1	0.365
rank2	0.707
rank3	0.827
rank4	0.898

Rank results using **T5\_max BM25**

rank1	0.372
rank2	0.72
rank3	0.834
rank4	0.899

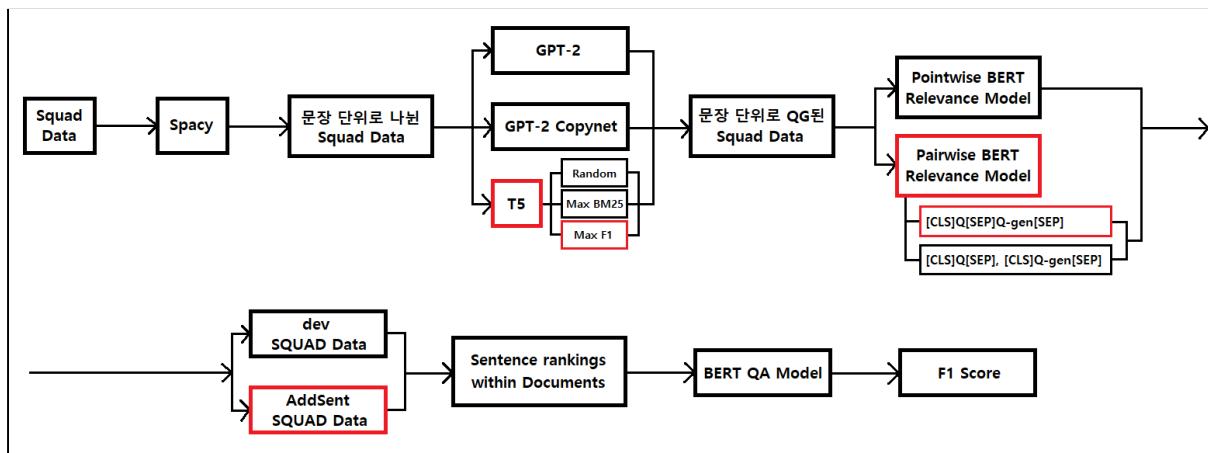
Rank results using **T5\_max F1**

rank1	0.405
rank2	0.737
rank3	0.86
rank4	0.914

Rank results using **T5\_max F1  
(hyperparameter tuning)**

rank1	0.437
rank2	0.72
rank3	0.837
rank4	0.905

위 표들은 T5 question generation 모델을 사용한 결과이다. 앞서 설명한 바와 같이 T5는 Random, Max BM25, Max F1, 이렇게 3가지 방식이 있는데 위의 3개의 표가 이에 대응하는 모델들의 결과이다. 이중에서 T5의 Max F1 방식이 3가지 방식 중 가장 결과가 좋은 것을 알 수 있었으며 마지막 표는 이를 hyperparameter tuning을 통해 조금 더 성능을 끌어 올린 결과이다. CopyNet\_0.1의 결과였던 0.369에 비해 훨씬 향상된 성능을 볼 수 있다.



이렇게 다양한 조건을 가지고 실험을 계속 진행해 본 결과, T5모델의 Max F1 방식을 사용한 question generation과 BERT의 Q와 Q\_gen을 함께 받는 pairwise relevance matching 모델을 이용하여 adversarial SQuAD 데이터셋에 적용하는 것이 가장 큰 성능 향상을 보이는 것으로 나타났다.

마지막으로 다음은 BERT question answering 모델을 사용하여 위의 실험들의 F1 스코어를 측정한 결과이다.

Model	Original	ADDSENT
ReasoNet-E	81.1	39.4
SEDT-E	80.1	35.0
BiDAF-E	80.0	34.2
Mnemonic-E	79.1	46.2
Ruminating	78.8	37.4
jNet	78.6	37.9
Mnemonic-S	78.5	46.6
ReasoNet-S	78.2	39.4
MPCM-S	77.0	40.3
SEDT-S	76.9	33.9
RaSOR	76.2	39.5
BiDAF-S	75.5	34.3
Match-E	75.4	29.4
Match-S	71.4	27.3
DCR	69.3	37.8
Logistic	50.4	23.2
<b>BertQA</b>	<b>73.9</b>	-
<b>OG Sent. Pairwise</b>	-	<b>41.9</b>
<b>QG GPT-2 CopyNet</b>	-	<b>35.4</b>
<b>QG T5-maxF1</b>	-	<b>40.4</b>

굵은 글씨로 쓰여진 결과 값들이 우리의 실험 결과이며 나머지는 기존 연구들의 결과이다. 우리가 학습시킨 BERT question answering 모델의 성능은 F1 스코어가 73.9이다. 원본문장,

GPT-2 CopyNet으로 생성한 question, T5 Max F1으로 생성한 question, 이렇게 3가지의 데이터에 대하여 각각 bert relevance 모델이 document 내에서 1등으로 ranking한 original문장에 대하여 QA모델을 적용하였을 경우의 F1 Score이다.

결론적으로 우리는 question generation 모델의 성능이 향상될수록 문서내에서 ranking을 더 잘하게 된다는 것을 알 수 있었다. 실제로 GPT-2 모델보다 더 좋은 성능의 T5모델을 사용하여 question generation을 한 결과가 더 좋은 것을 볼 수 있었다.

또한 question answering 모델 역시 성능이 향상되면 F1스코어를 올릴 수 있을 것이라 예상된다.

마지막으로 우리의 실험방식은 일반적인 데이터셋보다는 adversarial 데이터셋에서 더 강하다는 것을 알 수 있었다. 이는 dev SQuAD dataset보다 AddSent SQuAD dataset에서 기존 모델에 비해 더 성능이 향상되었다는 점을 통해서 알 수 있었다.

## 5.1 추후 연구 계획

우리가 사용한 question generation 모델은 정답이 무엇인지 모르는 상태에서 질문을 생성하는데 T5와 GPT-2, GPT-2 copynet에 대한 결과의 가장 큰 차이는 여러개의 정답 후보에 대하여 question generation을 하였는지 단 하나의 정답 후보에 대하여 하였는지의 차이였다. 즉 문장내의 각각의 단어들에 대하여 question generation을 수행한다면 실제 정답에 대하여 question generation을 수행할 확률이 올라가게 되고 원본질문과 유사한 질문이 나올 가능성도 높아지게 된다. 추후 이러한 모델을 사용하여 실험을 진행해 볼 예정이다.

다음으로 문장내의 대명사들을 원래의 고유명사로 바꿔줄 수 있는 모델이 있다면, question generation이 원본 question과 더 유사해질 가능성이 있기에 성능향상이 있을것이라고 생각된다. 예를 들어보면 she was born in 1996보다는 Sara was born in 1996가 더 원본 question과 비슷한 질문을 생성할 가능성이 높아진다. 따라서 이렇게 고유명사화를 해주는 모델을 구현하거나 구현된 모델을 가져와 연구에 적용을 해볼 예정이다.

## 6. 팀의 구성 및 팀원의 역할 배분

이 프로젝트를 함께 수행한 연구원들은 김주찬, 이상현, 이준영 이렇게 3명이 있으며 Data Intelligence Lab 소속 황승원 교수님과 한호재 조교님의 지도를 받았다.

다음은 우리 팀의 역할 분담이다.

김주찬: 서베이, 데이터 전처리, 모델 실험, 모델 학습 및 개선, PPT/리포트, 발표

이상현: 서베이, 데이터 전처리, 코드 구현 및 실험, PPT/리포트

이준영: 서베이, 데이터 전처리, 모델 실험, 모델 학습 및 개선, PPT/리포트, 발표

## 참고 문헌

- 1) Burges, Christopher JC. "From ranknet to lambdarank to lambdaMart: An overview." *Learning* 11.23-581 (2010): 81.
- 2) Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
- 3) dos Santos, Cicero, et al. "Beyond [CLS] through Ranking by Generation." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020.
- 4) Gu, Jiatao, et al. "Incorporating copying mechanism in sequence-to-sequence learning." arXiv preprint arXiv:1603.06393 (2016).
- 5) Jia, Robin, and Percy Liang. "Adversarial examples for evaluating reading comprehension systems." arXiv preprint arXiv:1707.07328 (2017).
- 6) Nogueira, Rodrigo, et al. "Document expansion by query prediction." arXiv preprint arXiv:1904.08375 (2019).
- 7) Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI blog* 1.8 (2019): 9.
- 8) Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).
- 9) Rajpurkar, Pranav, Robin Jia, and Percy Liang. "Know what you don't know: Unanswerable questions for SQuAD." arXiv preprint arXiv:1806.03822 (2018).
- 10) Tang, Duyu, et al. "Learning to collaborate for question answering and asking." *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018.