**제품 스펙 문서 및 숫자 정보처리를 위한 표 질의 응답 시스템**

**목차**

1. 과제의 배경 및 목표

1.1 과제 배경

1.2 과제 목적

2. 요구 조건 분석

3. 현실적 제약 사항 및 대책

4. 설계 문서

4.1 개발환경

4.2 사용기술

4.3 프로세스

5. 개발 일정 및 역할 분담

5.1 개발 일정

5.2 역할 분담

6. 참고자료

**1 과제의 배경 및 목표**

**1.1 과제 배경**

대부분의 정보는 웹이나 데이터베이스 및 문서들에서 찾을 수 있는 표 형식으로 저장된다. 여기에는 소비자 제품의 기술 사양에서 재무 및 국가 개발 통계, 스포츠 결과 등이 포함된다. 궁금한 점에 대해 답변을 찾으려면 표들을 직접 보거나 특정 질문에 대한 답변을 제공하는 서비스를 사용해야 한다. 이러한 답변을 자연어를 통해 질의할 수 있는 경우 훨씬 더 유용하며 접근성이 증대된다.

최근, 이러한 자연어를 전통적인 의미론적 구문 분석(semantic parsing)이 아닌 BERT 아키텍처를 확장하여 테이블 형식의 데이터 구조와 질문을 함께 인코딩하여 답변을 직접 가리킬 수 있는 모델인 TAPAS(TAble PArSer)오픈소스가 등장했다. TAPAS는 기존의 방식에 비하여 광범위한 도메인의 테이블에 적용할 수 있는 모델로써, 특정 테이블에 대한 질문이 아닌 임의의 질문으로 확장하기 유용하다.

한편, 위키백과 한 페이지 전체에서 표와 리스트가 포함된 지문을 대상으로 질의응답을 수행하는 KorQuAD라는 웹문서 기계독해를 위한 한국어 질의응답 데이터셋 역시 2019년에 발표되었다. 이렇듯 실사용 가능한 자연어 처리 모델 개발은 최근 많은 관심을 받고 있는 분야이다. 그 중에서도 표에 관한 자연어 처리 모델 개발이 현재 전세계적으로 연구가 되고 있지만 제품 스펙 표에 대한 연구는 아직까지 진행이 되지 않았다. 이번 과제에서는 다양한 레이아웃의 표에 대하여 자연어 처리를 해서 사용자의 질의에 응답하는 모델을 개발하는 것에 집중한다.

**1.2 과제 목적**

이번 과제에서는 앞서 언급했던 TAPAS와 KorQuAD와는 차별화되는, 제품 스펙을 제공하기 위한 표에 집중한 질의응답 모델을 만들고자 한다. 기존의 질의응답 모델에서는 똑같은 속성들을 가진 요소들에 대해서 임의의 질문에 대한 답을 얻을 수는 있지만, 실제 같은 카테고리의 제품이라도 제조사와 사용목적 등에 따라 표의 구성요소뿐만 아니라 표 자체의 레이아웃이 크게 다르기 때문에, 다양한 표의 레이아웃을 모두 학습할 수 있는 모델은 더 많은 실제 제품 스펙 데이터를 학습할 수 있다는 점에서 그 사용성이 높다. 이번 과제에서는 다양한 형태의 표를 학습하여 TAPAS처럼 질의응답이 가능한 높은 수준의 성능을 가진 모델 생성을 목적으로 한다.

기존의 TAPAS는 일반적인 형태의 표에 대해서는 높은 성능을 보여주지만 그렇지 않은 표에 대해서는 저조한 성능을 보인다. 실제 환경에서는 일반적이지 않은 Weak Supervision한 표들이 많으므로 이를 전처리 하거나 모델이 이해할 수 있도록 하는것이 목적이다. 이를 위해 우리는 먼저 모델에 학습시킬 인터넷상에서 제품 스펙시트를 수집할 것이다. 그후에 모델이 이해할 수 있도록 수집한 데이터를 전처리 하거나 모델을 튜닝할 것이다.



**그림 1 갤럭시 버즈의 스펙표**

그림 1과 같은 형태의 표는 기본적인 틀에서 벗어난 형태이기 때문에 이러한 데이터들을 처리하는 시스템을 제작한다. 먼저 다양한 표를 카테고리별로 모아서 이용가능한 형태로 데이터화 한다. 이것을 이용하여 질의, 그리고 응답 쌍을 각 구성원이 만들어서 이것들을 데이터에 적용시켜 본다. 이 중에서 가장 보편적으로 적용될 수 있는 것이 좋은 질의응답 쌍이므로 이것을 머신러닝 자연어 처리 툴(구글 Tapas)를 이용하여 학습시킨다.

**2. 요구 조건 분석**

1. 제품 스펙과 관련하여 자연어를 사용한 질문 응답이 가능해야 한다.

- 다음과 같은 자연어 질문들 ,”평균적인 모니터 크기가 얼마나 되니?”,

“용량이 500L 이상인 냉장고 찾아줘 ” , 에 대해 올바른 답을 돌려주어야 한다.

1. 다양한 형태의 표를 언어모델이 이해하도록 해야한다.

- 다양한 형태의 표를 전처리 한다.

- 언어 모델이 다양한 형태의 표를 처리하도록 튜닝한다.

**3. 현실적 제약 사항 및 대책**

1. 현실적 제약 학습을 위해 수집할 표 데이터의 다양한 성격(html코드, 이미지, 텍스트 등)

-> 기본적으로 html태그를 위주로 수집, 추가적인 데이터가 필요하다고 판단되면 지원되는 라이브러리들을 활용하여 text-based PDF과 이미지도 수집할 예정

2. 스펙 표 데이터를 모을 수 있는 카테고리가 너무 많음

-> 데이터를 구하기 쉬운 하나의 카테고리로 먼저 모델링을 한 뒤, 점차 카테고리를 추가할 예정

3. 높은 효율의 질의-응답 쌍을 학습 하도록 하는 방법

-> 조 구성원들이 모두 각각의 학습 데이터 쌍을 만들어서 가장 높은 효율이 나오는 질의-응답 쌍을 채택할 예정

**4. 설계 문서**

**4.1 개발환경**

개발언어: Python(자연어처리, )

개발도구: Pytorch(자연어 처리, 질의-응답 모델 학습), pyscripter, pycharm, VS CODE

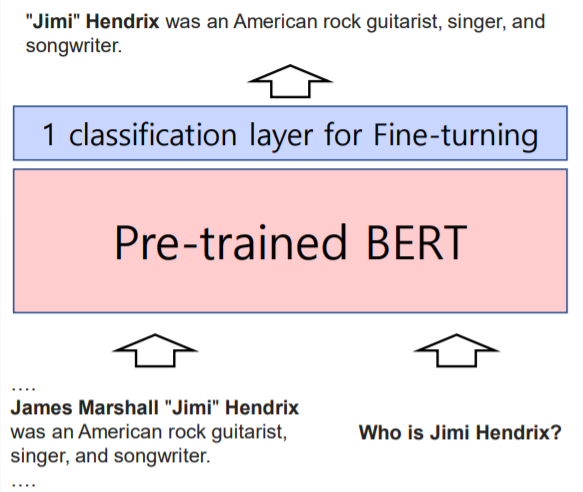
실행환경: AI 연구실 서버, window 환경

**4.2 사용기술**

***BERT(Bidirectional Encoder Representations from Transformers)***

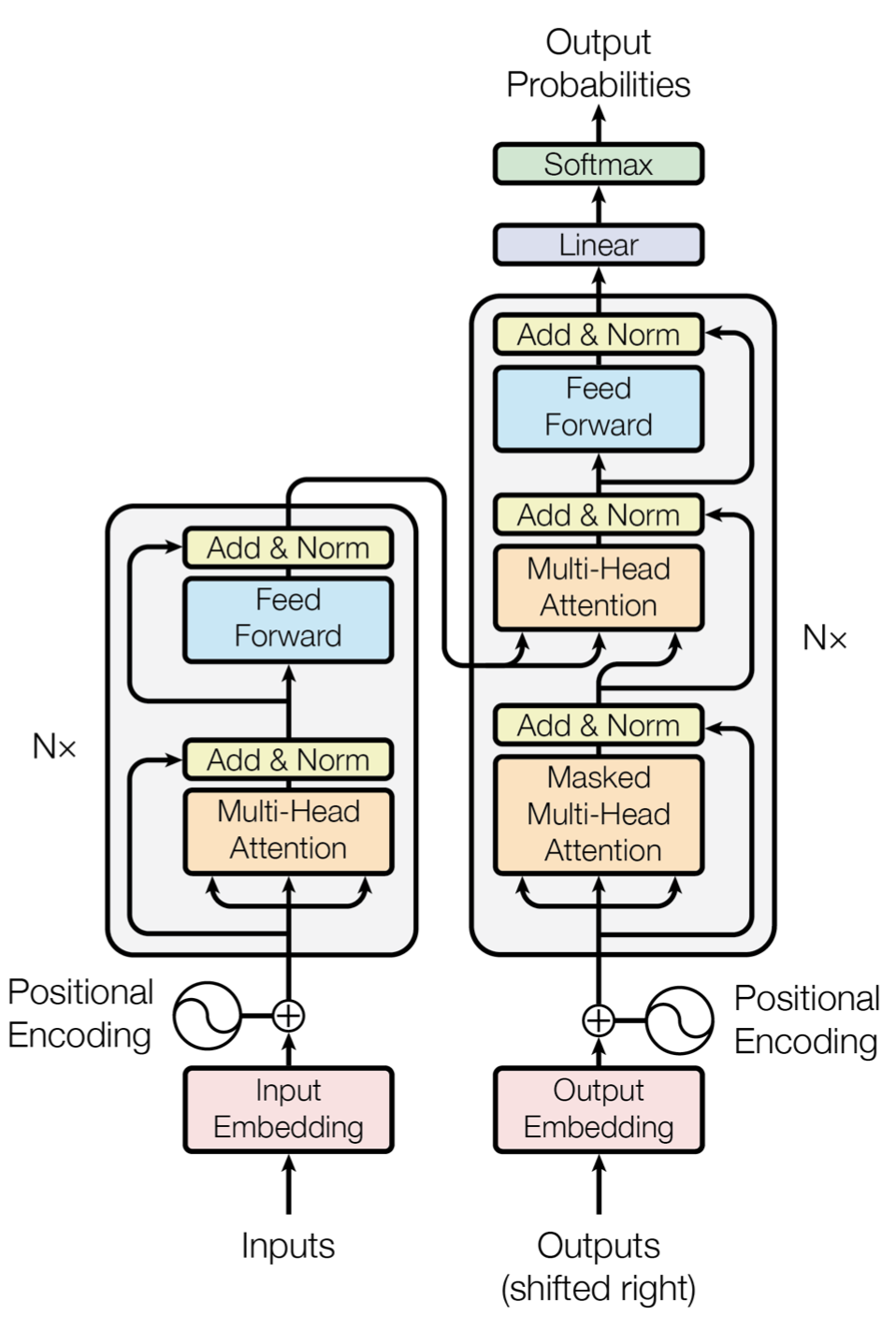
기존에 존재하던 NLP모델은 pre-training이 어려웠기 때문에 특정 task가 있으면 처음부터 학습시켜야 하는 단점이 있었다. 이에 대한 연구가 활발히 진행되었고, 만들어진 모델 중 하나가 BERT이다. BERT는 wikipedia와 bookcorpus단어를 상당수 학습한 모델이기 때문에 각광받는 모델중 하나이다. BERT는 주로 아래와 같은 경우에 사용된다.

1. Question and Answering  
   - 주어진 질문에 적합하게 대답해야 하는 매우 대표적인 문제이다.  
   - KorQuAD, Visual QA etc.
2. Machine Translation  
   - 구글 번역기, 네이버 파파고 등이 있다. .
3. 문장 주제 찾기 또는 분류하기  
   - 역시나 기존 NLP에서도 해결할 수 있는 문제는 해결이 가능하다. .
4. 사람처럼 대화하기  
   - 이와 같은 주제에선 매우 강력함을 보여준다.
5. 이외에도 직접 정의한 다양한 문제에도 적용 가능하다. 물론 꼭 NLP task일 필요는 없다.



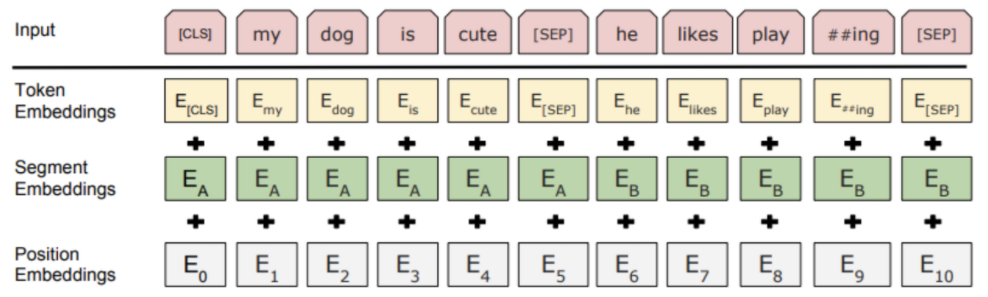
**그림 2 BERT모델을 이용한 결론도출**

위의 그림처럼 미리 학습된 BERT모델에 질의응답쌍을 넣어서 질의에 대한 응답을 도출한다. BERT는 transformer 구조를 사용하면서도 encoder 부분만 사용(아래 그림에서 왼쪽 부분)하여 학습을 진행한다. 기존 모델은 대부분 encoder-decoder으로 이루어져 있으며, GPT 또한, decoder 부분을 사용하여 text generation 문제를 해결하는 모델입니다. Transformer 구조 역시, input에서 text의 표현을 학습하고, decoder에서 우리가 원하는 task의 결과물을 만드는 방식으로 학습이 진행된다.



**그림 3 Attention mechanism - attention Paper**

BERT는 decoder를 사용하지 않고, 두 가지 대표적인 학습 방법으로 encoder를 학습시킨 후에 특정 task의 fine-tuning을 활용하여 결과물을 얻는 방법으로 사용된다. BERT는 학습을 위해 기존의 transformer의 input구조를 사용하면서도 추가로 변형하여 사용한다.



**그림 4 세 가지의 임베딩**

위의 그림처럼 세가지의 임베딩을 사용하여 문장을 표현하는데, 최종적으로는 이 세가지 임베딩을 더한 임베딩을 input으로 사용하게 된다. BERT가 문장표현을 학습하기 위해 사용하는 방법에는 두가지 unsupervised방법이 있다.

1. Masked Language Model
2. Next Sentence Model

Masked Language Model은 문장에서 단어의 일부를 mask토큰으로 바꾸고, 가려진 단어를 예측하도록 학습하는 것을 말하는데, 이 과정에서 BERT는 문맥을 파악하는 능력을 기르게 된다.

ex) 나는 하늘이 예쁘다고 생각한다 -> 나는 하늘이 [Mask] 생각한다.

ex) 나는 하늘이 예쁘다고 생각한다 -> 나는 하늘이 흐리다고 생각한다.

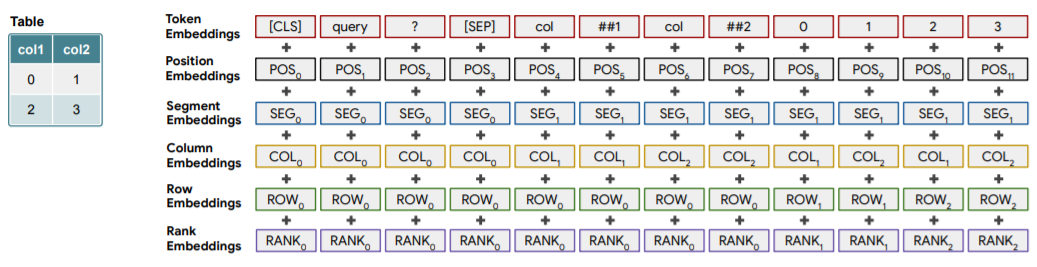
ex) 나는 하늘이 예쁘다고 생각한다 -> 나는 하늘이 예쁘다고 생각한다.

추가적으로 더욱 다양한 표현을 학습할 수 있도록 80%는 [Mask] 토큰으로 바꾸어 학습하지만, 나머지 10%는 token을 random word로 바꾸고, 마지막 10%는 원본 word 그대로를 사용하게 된다.

Next Sentence Prediction은 다음 문장이 올바른 문장인지 맞추는 방법이다. 이 방법을 통해 두 문장 사이의 관계를 학습하게 된다. 문장 A와 B를 이어 붙이는데 B는 50%확률로 관련있는 문장 혹은 관련 없는 문장을 이용한다. 이 방법은 question answering이나 NLI task의 성능 향상에 영향을 끼친다.

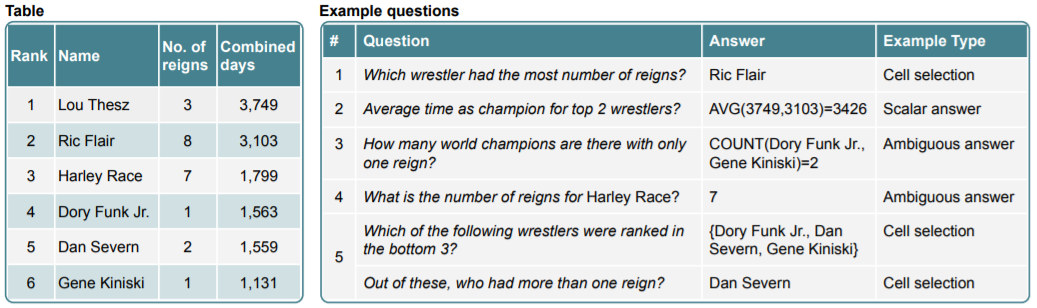
***TAPAS(TAble PArSing)***

TAPAS는 BERT 아키텍처를 확장하여 테이블 형식의 데이터 구조와 함께 질문을 함께 인코딩하여 답변을 직접 가리킬 수 있는, 광범위한 도메인의 테이블에 적용할 수 있는 모델이다. TAPAS에서는 자연어 질의를 처리하기 위해 특수한 임베딩으로 확장된 BERT 모델을 사용하며 질문과 행 내용을 행별로 인코딩한다. 다음 이미지는 입력에서 이들을 모두 추가하여 트랜스포머 레이어에 공급하는 방법을 보여준다.



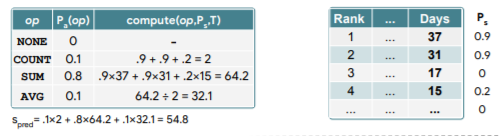
**그림 5 TAPAS embedding 기법**

위 그림은 왼쪽에 표시된 작은 표와 쿼리가 Tapas 모델에 어떻게 인코딩 되는지 보여준다. 각 셀 토큰에는 열 내에서 행, 열 및 숫자 순위를 나타내는 원소들이 있다. 모델에 Query와 Table을 이해시키기 위해 표현하는데 Token, Position, Segment Embedding은 BERT에서 사용했던 것이고 Tapas에서는 Table을 이해시키기 위해 추가적으로 Column, Row, Rank Embedding들을 사용한다. 아래는 학습에 사용한 테이블과 학습한 모델에 대해 질의했을 때 대답을 찾는 과정을 보여준다.



**그림 6 데이터와 질의응답쌍**

모델이 학습하는 과정에서, 해당 쿼리가 Cell selection 문제 인지, Scalar answer 문제인지, Ambiguous answer, Aggregation Operator 인지 판단하여 단순한 cell selection 거나 Scalar answer인 경우 쉽게 처리하고, Aggregation(집계문제)나 Ambigious인 경우 softmax를 적용하여 알맞은 operation을 찾아낸다. 아래는 간단한 operation을 찾는 모델 prediction의 예시이다.



**그림 7 aggregation operator와 확률**

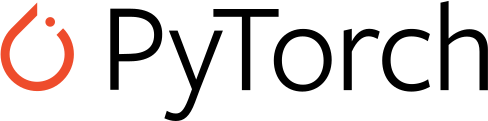
**사용하는 라이브러리 목록**

* ***pandas***



pandas는 데이터 조작 및 분석을 위해 Python 프로그래밍 언어로 작성된 소프트웨어 라이브러리이다. 특히 숫자 테이블과 시계열을 조작하기위한 데이터 구조와 연산을 제공한다. 이번 과제에서는 csv, JSON등의 표 데이터 파일을 pandas Dataframe으로 변환하여 pandas가 제공하는 연산자를 사용할 것이다.

* ***PyTorch***



Pytorch는 Python을 위한 오픈소스 머신 러닝 라이브러리으로 자연어 처리와 같은 어플리케이션을 위해 사용된다. GPU사용이 가능하기 때문에 속도가 상당히 빠르며 Tensorflow에 비해 직관적인 구조와 쉬운 난이도와 활성화된 사용자 커뮤니티로 사용자가 늘어나 있는 추세이다. Pytorch는 강력한 GPU 가속화를 통한 NumPy와 같은 Tensor 계산과 테이프 기반 자동 삭제 시스템을 기반으로 구축된 심층 신경망을 파이썬 패키지 형태로 제공한다.

**4.3 프로세스**

이번 과제를 구현하기 위해서 구상한 방법은 두가지이다. 첫번째는 기존의 TAPAS 모델이 인식할 수 있도록 다양한 레이아웃의 표를 전처리 하는 방법이고, 두번째는 레이아웃이 다양한 표들이 유의미하게 많은 경우, 기존의 TAPAS 모델을 tuning하여 이들을 인식하게 하는 방법이다.

모델 학습 과정의 플로우 차트는 아래와 같다.



**5. 개발 일정 및 역할 분담**

**5.1 개발 일정**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5월 | | | | 6월 | | | | 7월 | | | | 8월 | | | | 9월 | | | |
| 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 |
| tapas 및 관련기술 공부 | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |
| 착수보고서 준비 | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | 표 수집 및 전처리 | | | | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 중간 보고서 준비 | | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | 데이터 학습 및 튜닝 | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | 테스트 및 디버깅 | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 최종발표, 보고서 준비 | | | |

**5.2 역할분담**

|  |  |
| --- | --- |
| 이름 | 역할분담 |
| 민경언 | - 학습용 표 데이터 수집  - 시스템 테스트  - 모델 성능 평가  - 착수 발표 및 시연 준비 |
| 이상진 | - 시스템 테스트  - 보고서 작성  - 학습용 표 데이터 전처리 |
| 권선근 | - 모델 성능 평가  - 착수 발표 및 시연 준비  - 학습용 표 데이터 수집 |
| 공통 | - 전반적인 지식 이해  - 질의-응답쌍 생성 |

**6. 참고 자료**

1. [TaPas: Weakly Supervised Table Parsing via Pre-training](https://www.aclweb.org/anthology/2020.acl-main.398.pdf)

2. T아카데미 자연어 언어모델 ‘BERT’ 강의자료