**TextRank와 BERT를 통한 한국어 기사 문서 요약**

CUAI 3기 Machine Learning Track 2팀

김선종(통계학과), 박하람(사회학과), 이남규(미디어커뮤니케이션학부), 정영주(산업보안학과), 정현희(통계학과)

**[요약]** 프로젝트 목표, 과정, 결과를 요약하여 200자 이내로 기술

문서 요약은 원본의 의미를 가장 잘 요약하는 문장을 선택하는 자연어 처리 기법이다. 본 논문에서는 원본에 있는 내용을 바탕으로 문서를 요약하는 추출요약을 다루며, 이를 위해 고전적으로 사용되어 온 TextRank 기법을 사용하여 한국어 기사문을 요약한다. 추가적으로 최근 자연어처리 분야에서 압도적인 성능을 자랑하는 BERT의 문서요약 성능을 점검하였다.

**1. 서 론**

십수년전부터 인류는 정보의 바다속을 유영하고 있었다. 그러나 정보의 바다는 끝없이 팽창했고, 이제는 인류가 감당하기 힘들 정도의 거대한 정보의 우주가 되었다.

이 광막한 정보의 우주를 탐험하기 위해서는 내 앞에 놓인 거대한 텍스트 뭉치가 나에게 필요한 핵심 정보를 가지고 있는지 빠르게 판단하는 능력이 필요하다.

이를 위해 문서요약 기술이 발전하고 있다. 문서 요약이란 원본의 의미를 가장 잘 요약하고 있는 문장을 선택하여 원본보다 짧은 형태의 문서, 문장을 만드는 자연어처리 태스크다.[1] 문서 요약은 다양한 방법론을 통해 시도되고 있는데, 대표적으로 머신러닝의 방법론인 TextRank, 딥러닝의 방법론인 BERT(Bidirectional Encoder Representations from Transformers)가 존재한다.

해당 프로젝트에서는 두 문서요약 기법인 TextRank와 BERT모델을 이용한 한국어 기사문 요약 모델을 제안한다.

**2. 본 론**

**1) 한국어 기사문 요약 (진하게)**

문서요약은 크게 추출요약과 생성요약으로 나누어진다. 추출요약은 원본 문서내에 있는 문장을 사용하여 원본을 요약하는 반면, 생성요약은 모델이 문서의 내용을 압축할 수 있는 새로운 문장을 만들어 낸다.

추출 요약은 문서 내에서 동떨어진 문장을 이어 붙인 형태이기 때문에 문장간 유기성이 다소 떨어진다. 반면 생성요약은 모델이 새로운 문장을 만든다는 점에서 유기적인 요약문을 만들 수 있지만, 아직 자연어 처리 기술이 고도로 발달하지 않아 자연스러운 문장을 구성하지 못한다. 그러므로 이번 프로젝트에서는 추출요약에 초점을 맞추어 한국어 기사를 요약한다.

모델의 성능은 모델의 요약문과 사람이 만든 요약문의 ROUGE-N에 대한 F1-SCORE를 사용하여 평가한다. 즉 미리 만들어진 사람의 문서 요약문과 모델의 문서 요약문이 얼마나 유사한지를 바탕으로 모델의 성능을 측정한다.

**2) TextRank 알고리즘**

TextRank 알고리즘은 1999년 제안되었으며 초기 구글의 검색엔진 알고리즘으로 유명한 PageRank 알고리즘을 자연어처리 분야에 응용한 알고리즘이다. PageRank 알고리즘의 주요 아이디어는 다음의 두가지이다. 첫째, 다른 페이지들로부터 가장 많은 유입링크를 지니는 페이지가 가장 중요한 페이지이다. 둘째, 중요한 페이지로부터 유입링크를 받는 페이지 역시 중요한 페이지이다.

텍스트, 손목시계이(가) 표시된 사진

자동 생성된 설명

수식 1: PageRank 계산식

위 수식에서 PR(u)는 타겟 노드의 중요성을 의미하며 v는 링크된 노드, 는 각 마디 v의 링크 개수를 의미한다. 또한 페이지에 임의로 유입되는 경우를 고려하여 임의의 값 c로 값을 보정한다.

TextRank는 node를 page가 아닌 word를 사용한다는 점에서 차이가 있다. 만약 여러 개의 문장에서 특정 단어가 반복적으로 사용된다면 우리는 해당 단어를 핵심 키워드라고 판단할 것이다. 물론 형식형태소나 무의미하게 반복되는 단어가 키워드가 되는 것을 방지하기 위해 stopwords는 제거해야 한다. 이 방식을 통해 단어간의 동시출현행렬을 얻은 후 PageRank 수식을 적용하면 주요 키워드로 이루어진 word graph를 얻을 수 있다.

문서의 주요 키워드를 얻었다면 이제 중요 문장을 추출할 차례이다. TextRank에서 중요문장은 다른 문장과의 유사도가 높은 문장을 의미한다. 위에서 이미 문장을 주요 키워드 단위로 바꿔 놨기 때문에 다른 문장과 유사도가 가장 높은 문장이 곧 의미 있는 단어들로 구성된 문장이기 때문이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

수식 2: 문장 유사도 계산식

TextRank의 문장유사도 계산식에서 눈여겨볼 점은 분모에 로그를 취한점이다. 분모 S1, S2는 문장의 길이를 의미하는데, 문장의 길이가 짧을 경우 단어가 조금만 겹쳐도 문장의 유사도가 확 높아지는 경향이 있다. 이를 방지하기 위해 분모에 로그를 취함으로써 길이가 긴 문장이 유사도 측면에서 유리할 수 있도록 조정하였다.[2]

하지만 기사문의 특성상 모든 문장이 동일한 중요도를 가지지 않는다. 주요 키워드를 포함하는 지 여부와는 별개로 기자는 의도적으로 중요한 내용을 기사의 앞부분으로 배치한다. 그리고 중요도에 따라 차례대로 문장을 구성하는 역 피라미드형 글 구조가 기사문의 특징이다. 모델에서도 이를 반영하여 각 문장에 bias를 주었다. 본 프로젝트에서 각 문장의 bias는 임의의 값을 설정하다. 그 값은 첫번째 문장부터 네번째 문장까지 7, 5, 4, 3의 bias를 적용하였고, 이후 문장에는 bias를 적용하지 않았다.

위와 같은 모델을 적용한 TextRank 추출요약 모델의 성능은 다음과 같다.

표 1: TextRank 예측성능

|  |  |  |  |
| --- | --- | --- | --- |
| **모델** | **Rouge-1**  **(Recall)** | **Rouge-2**  **(Precision)** | **Rouge-L**  **(F-1)** |
| TextRank | 0.432 | 0.284 | 0.37 |

**3) KoBERT 알고리즘**

최근 딥러닝의 비약적 성장과 함께 텍스트 추출요약 분야에서도 딥러닝 알고리즘을 적용한 모델이 다수 제안되고 있다. 대표적인 알고리즘이 바로 BERT 알고리즘이다.

앞서 언급한 TextRank의 자연어처리 방식에는 근본적인 문제가 있는데, 문맥을 고려하지 않고 단어 단위로 워드 임베딩이 일어난다는 것이다. 특히 동음이의어의 경우 기존의 모델로는 의미가 다르지만 같은 벡터공간에 배정되어 그 차이를 인식할 수 없다. 이러한 단점을 해결하기 위해 딥러닝 방식을 이용한 임베딩 방식이 고안되었다.

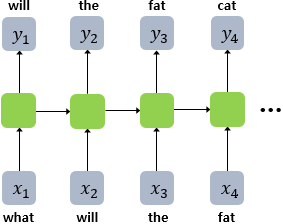


그림 1: RNN 언어모델

그림에서 보듯 딥러닝 모델은 이전 단어를 고려하여 output 벡터를 생성한다. 하지만 RNN 언어모델과 같이 초기 언어모델들은 단방향으로만 문장을 학습해 나가기 때문에 문장이 길어질수록 앞에 있는 단어의 영향력이 줄어들고, 특정 단어의 뒤쪽에 있는 단어가 앞쪽에 있는 단어에 영향을 미치지 못하는 문제가 발생한다.

BERT는 Transformer 구조와 Masked Language Model을 통해 이를 해결한다. Transformer는 주어진 시퀀스(단어들)를 한 번에 입력 받아 처리함으로써 특정 단어를 이해하기 위해 문장 내 다른 단어들과의 맥락을 고려하는 모델이다.[3] 이러한 특성 덕분에 문장의 마지막만 예측할 수 있는 다른 언어모델과 달리 BERT는 문장의 중간에 위치한 단어도 예측할 수 있다.

물론 이러한 성과를 얻기 위해서는 엄청나게 많은 양의 학습이 선행되어야 하는데, 일반적으로 BERT모델을 사용하는 경우 pre-trained 모델을 가지고 전이 학습하는 방식을 차용한다. 본 프로젝트에서는 한국어 위키 문장 5백만개와 한국어 뉴스 문장 2천만개로 pre trained된 KoBERT 모델을 사용하였다.

BERT를 활용하여 추출요약을 하기 위해서는 우선 각 문서에서 몇 번 문장이 가장 중요한 문장인지가 표시된 input 데이터가 필요하다. 이 input 데이터를 통해 도출된 output은 문장에 대한 정보를 담고 있다. 이를 활용하여 본격적인 추출요약을 하기 위해선 추가적인 작업이 필요하다.

우선 BERT모델을 통해 도출된 벡터를 Simple Classifier를 활용하여 중요 요약문장인지 예측하고, 이를 실제 값과 비교하여 로스값을 계산한다. 이어서 RNN모델의 일종인 LSTM을 활용하여 한 번 더 output을 도출하고, 위와 동일한 방식으로 예측 레이블을 정답 레이블과 비교한다. 마지막으로는 Transformer 구조를 한번 더 활용한다. 이미 BERT에서도 Transformer를 활용하지만, 이는 단어(토큰) 단위에 적용하는 것이기 때문에 문장 간의 관계를 반영하지 못한다. 이를 위해 Inter-sentence Transformer를 적용하여 예측 label을 도출하고 이를 정답 label과 비교한다.[4]

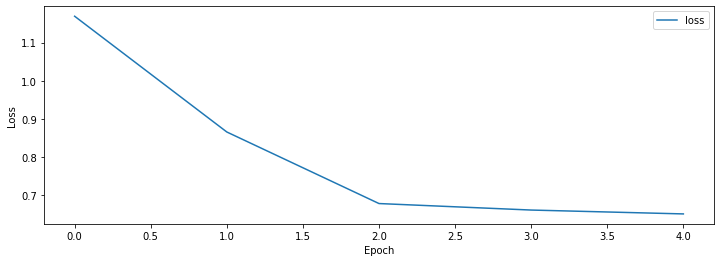


그림2: train 데이터 loss

BERT 모델을 통해 train 데이터를 반복 학습한 결과 Loss 값 0.65를 얻을 수 있었다.

**3. 결 론**

본 프로젝트에서는 TextRank와 BERT 알고리즘을 통해 한국어 기사문을 요약했다. 향후 보다 정교화된 모델이 정보의 우주를 여행하는 히치하이커를 위한 안내서로써 역할을 하길 기대한다.

**참고 문헌**

[1] 전재원, 황현선, 이창기, “BERT를 이용한 2단계 한국어 문서 요약”, 한국정보과학회 학술발표논문집, 2019년 12월

[2] TextRank를 이용한 키워드 추출과 핵심 문장 추출(구현과 실험), <https://lovit.github.io/nlp/2019/04/30/textrank/>

[3] 황상픔, 김도현, “한국어 기술문서 분석을 위한 BERT 기반의 분류모델”, 한국전자거래학회지

[4] Yang Liu, “Fine-tune BERT for Extractive Summarization”, arxiv:1903.10318