

Enhancing Search Functionality for Website Posts and Product Reviews: Improving BM25 Ranking Algorithm Performance Using the ResNet-Transformer Model

Hong-Ju Yang*, In-Yeop Choi**

*Graduates, Dept. of Computer Science, Kangnam University, Gyeonggi-do, Korea

**Professor, Dept. of Computer Science, Kangnam University, Gyeonggi-do, Korea

[Abstract]

This paper proposes a method to improve the search functionality for website posts and product reviews by using a ResNet-Transformer model in conjunction with the BM25 ranking algorithm. BM25 is a widely used algorithm in text-based search that ranks documents by evaluating their relevance to user queries. However, it has limitations in capturing local features of words and understanding the context of a sentences. To address these issues, this study applies a classification approach that combines the ResNet model, which excels at extracting local features, with the Transformer model, known for its strong contextual understanding, as weights for BM25. Experimental results demonstrate that the proposed method improves the nDCG metric by 9.38% and the aP@5 metric by 11.82% compared to BM25 alone. This suggests that implementing this method in search engines across various websites can provide more accurate results for post and review searches.

▶ **Key words:** ResNet, Transformer, ResNet-Transformer, Website search, BM25, Ranking algorithm

[요 약]

이 논문은 BM25 랭킹 알고리즘에 ResNet-Transformer 모델을 사용하여, 웹사이트의 게시글과 상품평 리뷰에 대한 검색 기능을 개선하는 방법을 제안한다. BM25는 사용자 질의와 문서 간의 관련성을 평가하여 순위화(ranking)하는 알고리즘으로 텍스트 기반 검색에서 광범위하게 사용되고 있다. 하지만 단어의 국소적인 특징 추출과 문장의 맥락을 파악하지 못하는 단점이 있다. 이에 본 논문에서는 국소적인 특징을 잘 추출하는 ResNet 모델과 문맥을 잘 파악하는 트랜스포머 모델을 결합한 분류 방법을 BM25의 가중치로 적용하여, 검색 기능을 향상시켰다. 테스트 결과 본 논문에서 제시하는 방법이 BM25 대비 nDCG 평가지표는 9.38%, aP@5 평가지표는 11.82% 향상됨을 확인하였다. 이를 통해 논문에서 제시한 방법을 여러 웹사이트의 검색창에 적용하면, 게시글과 상품평 리뷰 검색시에 정확한 결과를 제공해 줄 것으로 기대된다.

▶ **주제어:** 레즈넷, 트랜스포머, 레즈넷-트랜스포머, 웹사이트 검색, 랭킹 알고리즘, BM25

- First Author: Hong-Ju Yang, Corresponding Author: In-Yeop Choi
- *Hong-Ju Yang (inpink@kakao.com), Dept. of Computer Science, Kangnam University
- **In-Yeop Choi (billychoi@kangnam.ac.kr), Dept. of Computer Science, Kangnam University
- Received: 2024. 10. 10, Revised: 2024. 11. 08, Accepted: 2024. 11. 11.

I. Introduction

쇼핑몰, 가격 비교 사이트, 도서 검색 사이트, 대학교 홈페이지 등등의 웹사이트에서 사용자들은 웹사이트에 등록된 게시글과 상품평을 보기를 원한다.

게시글은 특정 주제나 경험에 대한 정보를 손쉽게 공유할 수 있고, 댓글이나 답글을 통해 특정 주제에 대한 독자들의 생각과 의견을 교환할 수 있다. 이를 통해 특정 주제에 대한 유용한 정보를 사용자들에게 제공할 수 있다.

상품평은 실제 사용자들의 경험을 바탕으로 작성되었기 때문에 제품의 장단점을 객관적으로 파악할 수 있다. 판매자가 제공하는 정보와 달리 사용자의 상품평은 제품의 실제적인 품질을 파악할 수 있다. 제품에 대한 긍정적인 평가가 많으면 신뢰할 수 있는 상품으로 간주 될 수 있고, 부정적인 평가가 많으면 신뢰할 수 없는 상품으로 간주 될 수 있다. 이것은 제품 품질에 대한 현실적인 정보를 제공한다. 또한 다양한 사용자들의 경험을 비교할 수 있다. 제품에 대한 긍정적인 평가가 많다고 해서 나에게 적합한 상품이라고 말할 수는 없다. 동일한 제품에 대해 다양한 환경에서 사용한 사용자들의 의견은 제품이 특정 환경에서 어떻게 작동하고 어떤 문제점이 있는지를 파악할 수 있다. 이처럼 상품평은 사용자가 구매를 결정하는데 있어서 매우 중요한 정보를 제공한다.

그런데 문제는 웹사이트에 존재하는 수많은 게시글과 상품평 문서 중에서 어떻게 사용자의 질의에 가장 유사한 문서를 찾을 것인가이다. 만약 해당 사이트에서 검색한 문서들을 가장 유사한 문서대로 순위를 부여해 주지 않으면 사용자들이 검색 후 나열된 문서를 일일이 찾아야 한다. 이 경우 검색된 결과에서 운 좋게 자기가 원하는 글을 빨리 찾을 수도 있다. 하지만 대부분 기대하는 검색 결과를 빨리 찾을 수 없다. 최악의 경우는 확인한 글 중에 페이지 끝까지 원하는 정보를 못 얻을 수도 있다. 이러한 과정은 사용자들에게 불편함과 많은 시간을 낭비하게 한다.

이러한 문제점을 해결하기 위해 사용자의 검색어에 가장 유사한 문서를 순위대로 부여하는 랭킹 알고리즘을 사용한다. 랭킹 알고리즘들은 사용자의 질의와 문서의 관련성을 평가하여 가장 유사한 문서들을 순위대로 정렬하여 화면에 보여준다.

현재 다양한 랭킹 알고리즘이 있다. 여러 종류의 랭킹 알고리즘 중에 어떤 랭킹 알고리즘이 게시글과 상품평 리뷰에 적합한지를 보기 위해 각 랭킹 알고리즘의 특징을 살펴보자.

첫 번째는 TF-IDF와 BM25(Best Matching 25)와 같이

단어의 빈도를 기반으로 문서의 관련성을 평가하는 알고리즘이다. TF-IDF는 단어 빈도(Term Frequency)와 단어의 희귀성을 나타내는 역문서 빈도(Inverse Document Frequency)를 결합하여 문서의 관련성을 평가한다. 문서 내 특정 단어의 빈도가 높을수록 중요하다고 보고, 반면 전체 문서에서 많이 등장하는 단어는 덜 중요한 것으로 간주한다. BM25는 TF-IDF를 개선한 알고리즘으로 문서 길이를 고려하여 TF 가중치를 적절히 조정하였다. TF-IDF 알고리즘에 문서 길이 조정 요소를 포함하여 긴 문서와 짧은 문서 간의 균형을 맞추었다.

두 번째는 PageRank와 HITS 같이 링크 구조를 활용한 랭킹 알고리즘이다. PageRank는 웹페이지의 중요도를 다른 페이지로부터의 링크 구조에 따라 평가한다. 더 많은 중요한 페이지로부터 링크된 페이지는 더 중요한 것으로 간주 된다. HITS(Hyperlink-Induced Topic Search)는 페이지를 "허브"와 "권위"로 나누어 평가한다. "허브"는 권위 있는 페이지로 연결된 페이지이고, "권위"는 많은 허브로부터 링크된 페이지이다. 키워드를 기반으로 관련성을 찾고 링크 분석을 통해 페이지의 중요도를 판단한다.

세 번째는 LSI와 Word2vec 같이 의미적(Semantic) 유사성을 기반으로 하는 알고리즘이다. LSI(Latent Semantic Indexing)은 키워드 간의 잠재적 의미를 분석하여 문서와 질의 간의 관계를 평가한다. 텍스트에서 숨겨진 의미 구조를 추출하여 키워드가 직접 등장하지 않더라도 관련성 있는 문서를 찾을 수 있다. Word2vec는 단어 간의 의미적 유사성을 학습하여, 질의에 대해 관련성 있는 문서를 찾는다. 단어를 고차원 벡터로 변환하고, 의미적으로 유사한 단어들이 가까운 벡터로 표현되므로 키워드에 대한 유사성을 잘 반영할 수 있다.

네 번째는 코사인 유사도를 이용한 알고리즘이다. Vector Space Model(VSM)은 문서와 질의를 벡터로 변환한 뒤 코사인 유사도를 이용해 두 벡터 간의 각도로 관련성을 평가한다. 코사인 각도가 작을수록(유사도가 높을수록) 높은 순위로 문서가 위치하게 된다.

다섯 번째는 언어 모델을 기반한 알고리즘으로 LMIR(Language Models for Information Retrieval)은 질의와 문서 간의 확률 분포를 기반으로 관련성을 평가한다. 각 문서가 주어진 질의를 생성할 확률을 모델링하고, 각 문서에서 특정 키워드가 등장할 확률을 계산하여 순위를 매긴다.

여섯 번째는 확률이론, 기계학습과 딥러닝을 기반한 알고리즘이다. PRM(Probabilistic Relevance Model)은 확률 이론에 기반하여 문서가 사용자의 쿼리와 관련된 확률

을 계산하여 평가한다. LTR(Learning to Rank)는 기계 학습을 이용해 다양한 특징(feature)을 학습해 질의와 문서 간의 관련성을 평가한다. Neural Ranking Models은 딥러닝을 활용하여 문서와 질의 간의 비선형 관계를 학습해 관련성을 평가한다.

이러한 다양한 랭킹 알고리즘 중에 게시물과 상품 리뷰에 가장 일반적으로 사용되는 랭킹 알고리즘은 BM25이다. BM25는 사용자 질의와 문서 간의 관련성을 평가하기 위해 개발된 알고리즘으로 문서 기반 검색 시스템의 정보 검색에서 널리 사용되는 점수 계산 모델이다. 점수 계산은 사용자의 질의에 포함된 각 단어의 문서 내 빈도(TF), 역 문서 빈도(IDF), 문서의 길이를 고려하여 계산한다. 이는 게시물과 상품평 같은 문서 기반 페이지의 내용을 검색하는 데 여러 가지 장점이 있다.

첫째는 단어의 중요도를 반영한다. BM25는 단어의 중요도를 반영하여 점수를 계산한다. 자주 등장하는 단어는 점수를 줄이고, 드물게 등장하는 단어는 더 높은 점수를 부여하여 정보의 특성을 잘 반영한다. 두 번째는 단어의 빈도를 정규화하여 처리한다. 정규화된 빈도 처리로 길이가 다른 문서들간에 비교를 공정하게 처리 할수 있다. 세 번째는 부분 일치도 고려한다. 사용자가 입력한 키워드가 문서에 부분적으로만 포함되어도 점수를 부여한다. 이는 사용자가 입력한 단어가 문서에 어떻게 나타나는지를 반영하여 검색 결과의 다양성을 높인다. BM25는 단어의 부분 일치, 단어의 빈도, 단어의 중요도와 문서의 길이에 따라 점수를 조정하여 더 정확한 검색 결과를 제공한다.

하지만 BM25 랭킹 알고리즘에서 검색 결과를 높이기 위해 고려하지 않은 사항이 있다. 그것은 문장들의 문맥을 파악하는 것이다. 문서 전체를 정확하게 파악하려면 문장들의 문맥을 파악하는 것이 매우 중요하다. 또한 문서들의 유사도 검사에서 키워드의 국소적인(local) 특징을 파악하는 것은 매우 유용하다. 이는 특히 텍스트에서 특정 단어들 이 어떤 문맥에서 나타나는지를 고려할 때보다 정확한 유사도 평가에 도움을 준다. 하지만 BM25 랭킹 알고리즘은 문맥을 파악하는 방법과 키워드들의 국소적인 특징을 파악하는 방법이 없다. 그러므로 본 논문은 BM25 랭킹 알고리즘에 문맥 이해와 단어들의 국소적인 특징을 파악하는 방법을 결합한 방법을 가중치로 적용하여 게시글과 상품평의 정확도를 높이고자 한다.

본 논문에서 BM25 알고리즘의 정확도를 높이기 위해 가중치로 적용한 방법은 ResNet-Transformer 모델로 문서를 분류 후 BM25의 가중치로 적용하는 것이다. 2017년 트랜스포머가 발표된 이후로 트랜스포머는 문자 분류에

가장 널리 사용되는 모델이 되었다. 트랜스포머는 시계열 데이터를 처리할 때 RNN 계열의 LSTM처럼 자가 회귀 메커니즘을 사용하지 않고 어텐션(attention) 메커니즘을 사용한다. 트랜스포머의 Self-Attention 메커니즘은 입력 시퀀스의 모든 토큰 간의 관계를 고려할 수 있어 문맥을 잘 이해할 수 있다. 그리고 어텐션을 통해 시계열 데이터나 순차 데이터를 병렬 처리할 수 있다. 트랜스포머는 병렬 처리함으로 LSTM에 비해 학습 속도 및 추론 속도가 빠르고, 장기 의존성을 더욱 효과적으로 처리할 수 있다. 이러한 특징을 가진 트랜스포머를 랭킹 알고리즘에 적용하고자 한다.

그러나 검색어의 경우 전체 문장에서 키워드가 떨어져 있을 수 있고, 키워드들이 여러 개인 경우가 많이 있다. 예를 들어 “인간과 컴퓨터 간의 상호작용 연구”, “컨텐츠 관리 시스템” 등등 많은 복합어 들이 있다. 이 경우 단어를 개별적으로 취급 시에는 유사하지 않은 많은 결과를 보여 줄 수 있다. 이처럼 두 개 이상의 단어가 합쳐져서 키워드가 되는 국소적인 특징이 중요한 경우가 있다.

그러나 트랜스포머는 전역적 의존성을 학습할 수 있는 특징 때문에 강력한 귀납적 편향(inductive bias)을 가지고 있지 않다. 이것은 트랜스포머는 입력 데이터의 국소적인 패턴에 대해 특별히 민감하지 않다는 말이기도 하다. 이러한 이유로 해서 트랜스포머의 인코더(Encoder)에 국소적인 특징을 학습하는데 뛰어난 CNN을 결합한 CNN-Transformer 모델을 사용하여 문자 분류의 성능을 높일 수 있다. CNN은 국소적 패턴을 학습하는데 최적화된 구조를 가지고 있다. 하지만 CNN 모델은 입력된 텍스트를 여러 영역으로 나누어 특정 지역의 특징(feature)만을 추출하므로, 문장 안에서 단어들 간의 의존성을 파악하지 못하는 한계가 있다. 깊은 네트워크에서는 기울기 소실(Gradient Vanishing) 또는 기울기 폭발(Gradient Exploding) 문제가 발생할 수 있다. 이러한 문제를 해결하기 위해 ResNet을 도입하였다. ResNet 모델의 스킵 연결(skip connection)은 CNN 모델의 기울기 소실 문제를 해결한다. 그리고 더 깊은 신경망 학습이 가능함으로 CNN 모델보다 더 좋은 결과를 얻을 수 있다.

본 논문은 웹사이트의 검색창에서 게시글과 상품평의 검색 결과를 향상시키기 위해서 문맥을 잘 파악하는 트랜스포머와 국소적인 특징을 잘 파악하는 ResNet을 결합한 ResNet-Transformer 모델을 BM25 알고리즘의 가중치로 적용하였다. 이를 통해 웹사이트에서 사용자 질의에 가장 유사한 게시글과 상품평을 검색하도록 하였다.

II. Preliminaries

1. Related works

웹사이트에서 제공하는 검색 기능의 효율성을 증가시키기 위해서는 단어가 아니라 문맥을 파악하기 위한 문장 수준 분류 작업이 필요하다. 문장 수준의 분류 작업을 위해서 국소적인 특징을 이해하고 문장의 문맥을 파악하는 것이 핵심 요소이다.

문장 수준 분류 작업을 위해 Kim[1]은 사전 훈련된 단어 벡터 위에 훈련된 CNN을 사용하였다. 영화 리뷰사이트에 게시된 댓글과 평점 정보를 CNN을 이용하여 리뷰가 긍정인지 부정인지 분류하는 모델을 만들었다. n 개의 단어로 이뤄진 리뷰 문장을 각 단어별로 k 차원의 행벡터로 임베딩한 후에 CNN을 이용하여 분류한 것이다. 여러 벤치마크에서 우수한 결과를 달성한다는 것을 보여주었다.

Conneau[2]는 매우 깊은 CNN을 문자 분류에 적용하였다. 이 논문은 문자 분류 작업이 모델의 깊이에 따라 개선되는 것을 보여주었다. CNN의 성능 저하를 극복하기 위해 ResNet 모델을 사용하여, 네트워크에서 기울기(Gradient)가 더 쉽게 흐를 수 있도록 컨볼루션 블록 간의 바로 가기 연결(skip connection)을 도입했다.

Han[3]은 감정 분류 같은 고수준 문서 표현을 위하여 모델의 깊이를 깊게 할 수 있는 ResNet과 문서의 중요한 구성 요소를 포착하기 위한 어텐션 메커니즘을 사용하는 모델을 제시하였다. 2개의 ResNet 블록과 어텐션 메커니즘을 사용하면 3개의 공개 감정 분류 데이터 세트에서 최고의 성능을 달성할 수 있었다고 한다. 그러나 ResNet을 이용한 문자 분류에 대한 논문은 2018년 이후로 연구되지 않는 경향을 보인다. ResNet은 문자 분류 보다는 이미지 처리에 최적화[4] 되어 CNN의 특성을 더욱 강화하는데 주로 사용되기 때문이다.

Wang[5]은 단어 임베딩 후에 CNN을 이용하여 지역적인 특징을 추출한 다음에 LSTM을 사용하여 추출한 특징을 연속적으로 통합하였다. CNN 모델과 LSTM 모델을 결합해 문장 내의 지역 정보를 순차적으로 통합해서, 긴 문장에서 단어 간의 의존성을 파악할 수 있게 하였다. Park[6]은 CNN과 LSTM의 조합모델을 이용하여 문자를 분류하고 이를 통해 감성분석의 분류 정확도를 개선하였다. CNN은 지역적 특징을 추출하는데 강점이 있고, LSTM은 장기 의존성을 기억하는데 강점이 있는 점을 이용한 것이다. 그러나 LSTM은 병렬 처리를 하지 못하고 여전히 긴 문장에서는 장기 의존성 문제가 발생하는 한계가 있다.

Vaswani[7]는 기계 번역 작업을 하는데 재귀와 합성곱

을 전혀 사용하지 않고, 어텐션 메커니즘에만 사용하는 네트워크 아키텍처인 트랜스포머를 제안한다. 트랜스포머 모델은 RNN 계열과 달리 병렬 처리할 수 있고, 상태를 기억하지 않고 문맥 파악 능력이 뛰어나서 문자 분류에 널리 사용된다. 하지만 트랜스포머는 입력 데이터의 국소적인 패턴에 대해 특별히 민감하지 않은 단점이 있다.

이에 Benarab[8]은 트랜스포머의 인코더 모델의 장점과 지역적인 패턴을 잘 파악하는 CNN 모델을 결합한 CNN-TransEnc 모델을 제시하였다. CNN-TransEnc 모델은 AG 뉴스 데이터 세트[9]와 DBPedia-14에서 높은 성능을 달성하였다. 그러나 CNN-TransEnc 모델은 CNN의 장거리 의존성 한계를 트랜스포머가 커버할 수 있지만, 다양한 크기의 텍스트 패턴을 한 번에 학습하는 데는 한계가 있다.

이에 본 논문은 트랜스포머의 인코더 모델에 매우 깊은 네트워크를 효과적으로 학습하여 문자 분류에 있어서 매우 복잡한 패턴도 학습할 수 있는 ResNet-Transformer 모델을 제시한다. 이 모델을 단어의 빈도를 기반으로 하는 BM25 랭킹 알고리즘의 가중치로 적용하여 정확도를 더 높일 수 있다.

III. The Proposed Scheme

웹사이트에서 사용자의 질의에 가장 유사한 게시글과 상품평 리뷰 결과가 검색되도록 본 논문에서 제시하는 방법은 BM25 랭킹 알고리즘에 ResNet-Transformer 모델 분류 가중치를 부여하는 방법이다. BM25 랭킹 알고리즘의 문맥을 파악하지 못하고, 국소적인 특징을 추출하지 못하는 단점을 보완한 것이다. 문맥 파악에 장점이 있는 트랜스포머와 국소적 특징 추출에 장점이 있는 ResNet 모델을 결합한 분류 방법을 BM25의 가중치로 적용하여 사용자 질의에 대한 검색의 정확성을 높였다.

이를 증명하기 위해 두 가지 단계를 수행하였다. 첫째는 ResNet-Transformer 모델과 여러 유형의 신경망을 적용한 모델을 비교하였다. ResNet-Transformer 모델이 Transformer 모델, CNN-Transformer 모델, CNN-LSTM 모델보다 더 좋은 성능을 보여주는지 테스트를 진행하였다. 둘째는 테스트 웹사이트를 구축한 후 기존의 검색 방법과 BM25에 ResNet-Transformer 모델의 가중치를 적용한 검색 방법을 비교하였다. 비교 방법은 테스트 웹사이트의 검색창에서 질의를 입력하고 검색하면, 질의에 대한 검색 결과 중 가장 유사도가 높은 문서 순위부터 순

위별로 5개를 보여준다. 이처럼 랭킹된 문서 5개와 사용자의 검색 키워드가 얼마나 유사한지를 10명의 평가자가 평가를 수행했다. 그 평가 결과를 통해 본 논문에서 제시하는 방법의 검색 기능이 얼마나 향상됐는지를 보였다.

1. Data collection and labeling

본 논문에 사용되는 데이터는 강남대학교 홈페이지[10]의 웹페이지들을 웹크롤링 하여 수집하였다. 수집한 데이터를 Table 1.처럼 총 4개의 class로 분류하였다.

Table 1. Class classification number

Class	Classification Number
Bachelor's Information	0
Scholarship	1
Study Counseling	2
Employment Startup	3

한 class당 1,560개씩으로 데이터를 수집한 후에 라벨링하여, 총 데이터 6,240개를 만들었다. 라벨링은 Table 2와 같이 수행했다. 각 웹페이지의 게시글들을 한국 형태소 분석기 노리(Nori) [11] 이용하여 명사를 추출하였다. 형용사인 경우에 형용사에 있는 명사 부분만 추출하였다. 예를 들어 “사랑스럽게”라는 형용사 같은 경우에 “사랑”이라는 명사만을 추출하였다. 부사에서도 명사 단어가 포함된 경우에도 명사만을 추출하였다. 한 문장에서 중복단어가 있는 경우 중복을 허용하여 "nouns"에 남겨두었다. 명사만 추출한 이유는 검색할 때 명사 단어를 넣기 때문이다. 아래 테이블은 한 페이지의 글에서 위와 같은 방법으로 명사를 추출한 후에 라벨링한 방법을 보여준다.

Table 2. Labeling Method

label	1
nouns	'학생', '신청', '국가', '근로', '신청', '학기', '국가', '근로', '장학금', '일반', '유형', '추가', '학생', '신청', '안내'

페이지의 게시글에는 제목과 본문이 있는데, 제목과 본문에 대해서 가중치를 주지 않고, 하나의 데이터로 사용하였다. 검색 결과는 제목과 본문에 있는 단어들을 찾기 때문이다. 글자수 길이가 한 개인 경우는 제외하였다. 이후 Word2vec 모델[12]을 이용해 nouns 데이터를 고차원 벡터로 변환한다. 이를 통해 단어는 뜻을 더욱 효과적으로 지니며, 단어들간의 유사도 또한 파악할 수 있다. 만약 nouns의 단어가 Word2vec로 변환할 수 없다면 np.zeros를 넣는다. nouns의 최소 길이는 0, 최대 길이 1792, 평균 길이 84.8152이다. 문장에서 중복되는 중복단어를 제거하지 않았다. 그 이유는 중복되는 것과 중복 위치, 단어가 등장하는 순서 모두 문맥을 형성하는데 중요한 힌트가 될 수 있기 때문이다.

2. Performance comparison with various neural networks

CNN-LSTM 모델, Transformer 모델, CNN-Transformer 모델과 본 논문에서 제시한 ResNet-Transformer 모델을 수집한 데이터를 가지고 성능을 비교하였다. 데이터는 4개의 모델별로 각각 1,560개씩 이고, 이 데이터를 훈련, validation, 테스트용으로 각각 8:1:1로 구성하였다. 라벨링한 데이터를 가지고 4개의 모델을 학습한 후에 측정된 훈련과 평가의 정확도는 Fig 1.과 같다.

Table 3. Training and evaluation accuracy of four models

Model	Train Accuracy	Validation Accuracy
CNN-LSTM	80.73%	81.57%
Transformer	91.75%	90.38%
CNN-Transformer	87.62%	87.98%
ResNet-Transformer	98.08%	93.59%

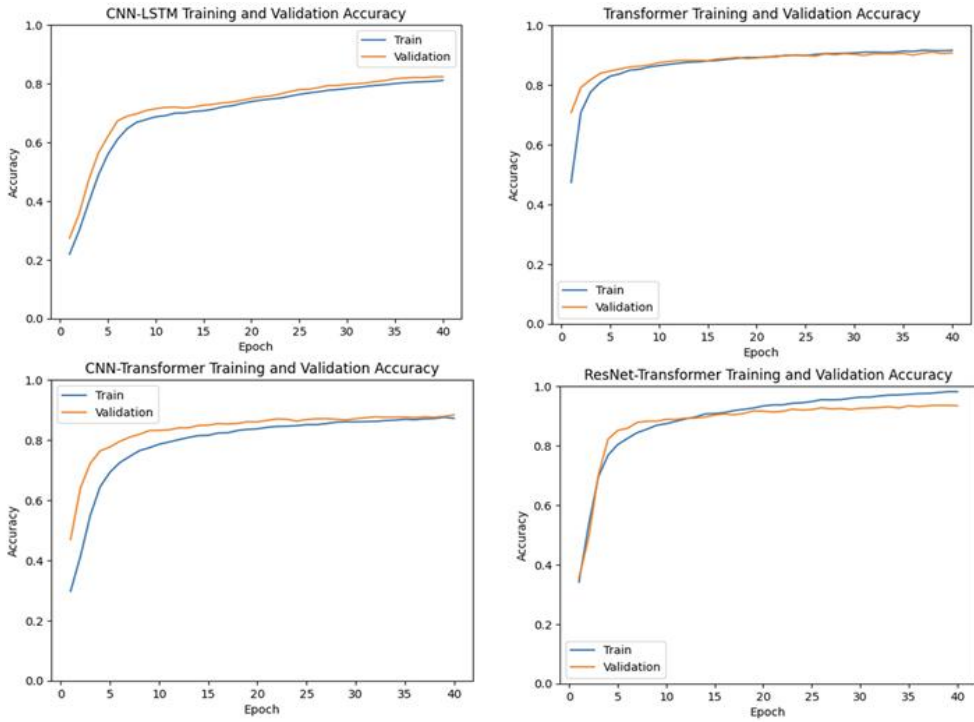


Fig. 1. Training and evaluation accuracy of models

4개 모델에 대한 테스트 정확도를 보자. 테스트 3번을 수행한 후에 평균을 구하였다.

평균 정확도는 다음과 같다.

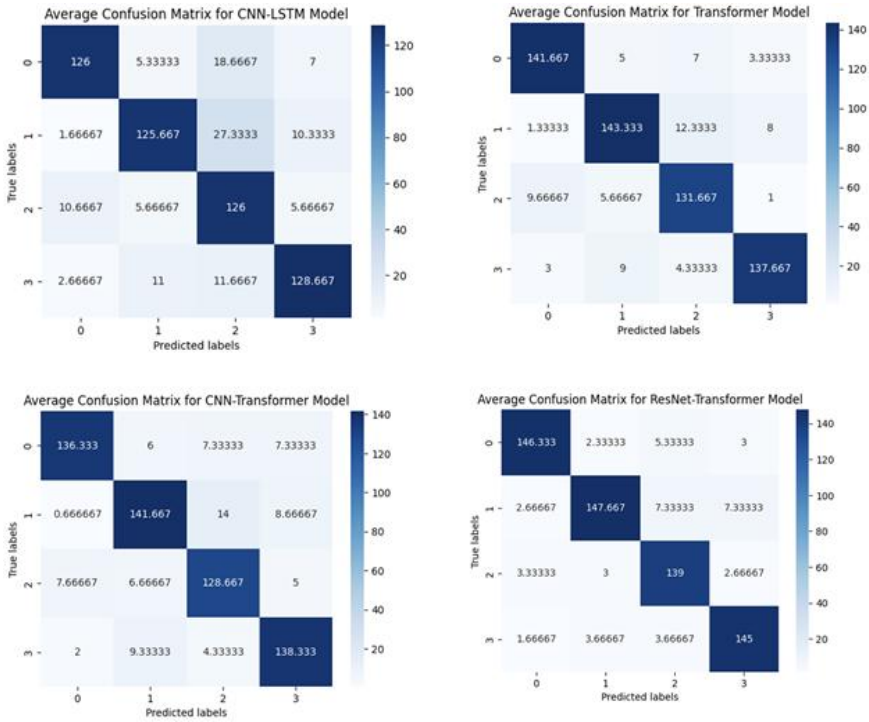


Fig. 2. Test accuracy of models

Table 4. Test accuracy of four models

Model	Test Accuracy
CNN-LSTM	82.25%
Transformer	89.00%
CNN-Transformer	87.50%
ResNet-Transformer	92.75%

ResNet-Transformer 모델의 평균 테스트 정확도는 92.75% 이다. 위의 실험을 통해서 ResNet-Transformer 모델의 성능이 가장 좋음을 확인하였다. 이제 Resnet-Transformer 모델을 BM25 알고리즘의 가중치로 적용하여 BM25의 검색 결과값과 비교하였다.

3. Applying the Resnet-Transformer model as a weight for the ranking algorithm

ResNet-Transformer 모델이 다른 3가지 모델에 비해서 성능이 좋음을 확인하였다. 이후 BM25 랭킹 알고리즘에 Resnet-Transformer 모델의 가중치를 적용한 방법이 검색 기능의 정확도를 얼마나 향상시키는지 테스트 하였다. 이후로 이 알고리즘을 “RNTrans BM25”로 명명하겠다.

3.1 Collect test data

보안상 강남대학교의 웹서버 및 데이터베이스를 이용할 수 없었다. 따라서 강남대학교의 허가를 받고 메인 홈페이지의 게시글들을 Jsoup[13]를 사용하여 크롤링했다. 백엔드 환경인 Spring Boot[14]에서 정적인 게시글을 크롤링하기 위해 Jsoup 라이브러리를 사용했다.

3.2 Frontend implementation

프론트엔드 부분은 Thymeleaf[15]를 사용하여 동적인 HTML로 만들었다. 작성된 HTML은 HTTP 프로토콜을 통해 클라이언트의 웹브라우저에게 전달되어 웹페이지의 내용을 표시한다. Fig. 3은 작성된 웹사이트[16]이다.



Fig. 3. Website UI

3.3 Backend implementation

RNTrans BM25의 백엔드의 웹 어플리케이션은 자바로 개발을 하였고, 자바 기반의 웹어플리케이션을 실행하기 위해 Spring Boot 프레임워크를 사용하였다. ResNet-Transformer 모델은 파이썬으로 작성되었기 때문에 이를 실행하기 위해 파이썬으로 작성된 마이크로 웹 프레임워크인 Flask[17]를 사용하였다. Spring Boot의 웹 어플리케이션과 Flask의 웹 어플리케이션은 RESTful API[18][19]를 통해 데이터를 주고받는다. DB는 MySQL을 사용하였고, 게시글들은 MySQL에 테이블 형태로 저장된다. 강남대학교 메인 홈페이지에서 크롤링한 게시글들은 MySQL에 테이블 형태로 저장된다. 크롤링한 게시글에는 분류 값들이 함께 들어가 있다.

3.4 Build a test website

Fig. 4는 본 논문을 위해 강남대학교 웹사이트의 데이터를 가지고 구현한 테스트 웹사이트 구성도이다. ①~⑦까지의 숫자는 RNTrans BM25가 수행하는 순서를 표기하였다. 테스트 웹사이트에서 총 3가지 랭킹 방법을 비교하였다. 첫째는 현재 강남대 홈페이지에서 사용하는 것으로 보이는 검색 방법이다. 이 방법은 검색 키워드가 포함된 페이지들을 모두 검색하여 최신순으로 보여주었다. 이렇게 구현한 이유는 똑같은 키워드로 강남대 홈페이지의 검색창에서 테스트했을 때 나온 결과가 거의 비슷하기 때문이다. 이를 “KNU”로 명명하겠다. 둘째는 BM25이다. KNU와 BM25는 Fig. 4의 ②와 ③은 수행하지 않는다. 셋째는 본 논문에서 제시하는 RNTrans BM25이다.

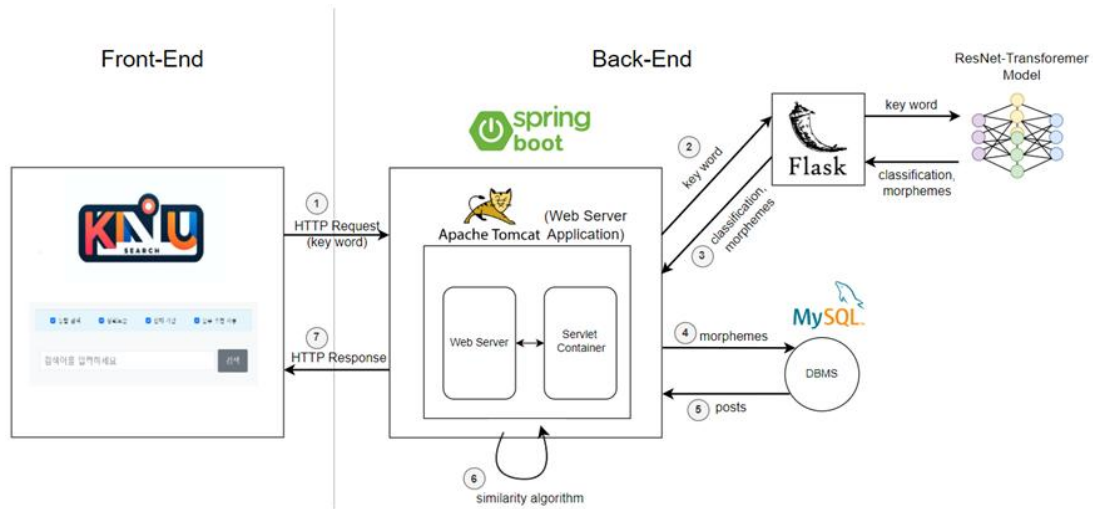


Fig. 4. System Configuration Diagram for the Test Website

3.5 Explanation of the RNTrans BM25 Execution Process

RNTrans BM25의 동작은 Fig. 4의 ①~⑦의 순서대로 수행한다. 사용자가 검색하면 Spring Boot의 웹어플리케이션이 HTTP 요청을 받는다. Spring Boot의 웹어플리케이션은 Restful API을 통해 Flask 서버에 있는 웹어플리케이션에 요청된 사용자의 키워드를 전달한다. Flask의 웹어플리케이션은 키워드를 형태소 단위로 분리하고 ResNet-Transformer 모델을 통해 분류를 수행한다. 그리고 분류 값과 형태소를 Spring Boot의 웹어플리케이션에 전달한다. Spring Boot의 웹어플리케이션은 받은 분류 값과 형태소를 가지고, MySQL에 쿼리를 수행한다. 쿼리가 수행된 후에 나온 결과를 아래의 두 가지 알고리즘을 적용하여 순위화한다.

첫 번째는 BM25 알고리즘으로 도출된 점수에 AI 분류 가중치 값을 적용하는 것이다.

$$Score_{final}(post) = \begin{cases} \text{if } post.classification = \text{주어진 분류} \\ Score_{BM25}(post) \times Weight \\ \text{otherwise} \\ Score_{BM25}(post) \end{cases}$$

여기서 Weight 값은 2이다. 이 Weight 값은 여러 가지 값을 적용하여 테스트한 후에 가장 좋은 성능을 보인 값이다.

결국 테스트를 통해 최적화한 것이다. 이 값을 적용한 후에 마지막으로 시간 가중치를 적용하였다.

$$DaysAgo = \text{현재 날짜} - \text{게시글 날짜}$$

$$Time_{weight} = \frac{1}{\log(1 + DaysAgo) + 1}$$

$$Score_{final} = Score_{final} \times Time_{weight}$$

시간 가중치를 적용한 이유는 점수 값이 같을 경우 최신의 게시글과 상품평의 순위를 상위에 정렬하기 위함이다. 날짜 차이에 따른 값의 변화율을 완만하게 조정하기 위해 log 함수를 사용하였다. 이렇게 2가지 알고리즘을 적용한 후에 테스트 웹상에 상위 5개를 순서대로 보여준다.

IV. Experimental Results

테스트 웹사이트 검색창에서 키워드를 입력한 후에 KNU, BM25와 RNTrans BM25의 검색 결과를 비교하였다.

1. Search results performed on a test website

8명의 학생들에게 학교 홈페이지에서 검색하고 싶은 질의어 20개를 받았다. 질의어는 짧은 문장과 긴 문장이 있으며 비슷한 문장은 제거하여 총 100개를 선택했다. RNTrans BM25의 검색 결과의 성능을 평가하기 위해서 평가 웹사이트[20]를 만들었다.

평가 웹사이트에서 질의어를 입력하고 검색을 수행하면, 아래 그림과 같이 상위 5개의 링크가 보이는 4개의 검색 결과가 나온다. KNU+ResNetTrans는 논문과 상관없이 KNU와 단순히 비교하기 위해 추가하였다.



Fig. 5. Search results for evaluation websites

평가 방법은 입력된 질의어와 Fig. 10과 같이 랭킹된 상위 5개의 내용을 비교해 보면서 질의어와 검색 결과의 유사성을 평가하는 것이다. 평가자 10명이 아래의 Table 5와 같이 평가 점수를 기록하도록 하였다.

Table 5. Example of completing a questionnaire

Rank No	KNU	BM25	RNTrans BM25
1	3	3	5
2	2	1	2
3	1	4	2
4	5	5	1
5	1	3	5

평가자는 가장 유사하다고 느끼는 정도에 따라 (5점, 4점, 3점, 2점, 1점)을 부여하도록 했다. 검색 결과가 없으면 0점을 부여했다. 테스트에 사용한 10개의 질의어는 8명의 학생들로부터 받은 100개의 질의어 중에 무작위로 선택하였다. 질의어를 이후로 “QW(Query Words)”으로 명명하겠다.

Table 6. Query Words

No	Query Words
QW1	미국 교육
QW2	교환학생 휴학
QW3	조기 졸업
QW4	사회복지학과 전공기초 강의명 리스트
QW5	일본 학생 소프트웨어 전공 데이터 사이언스 전공 전공 취직 일본 취업
QW6	장학금 수혜내역 어디서 확인
QW7	안녕하세요 제가 졸업요건을 보려 하는데 19년도 경제세무학과 입학입니다. 과이름이 바뀌면서 군복학후 정경학부 세무학전공으로 변경하였는데 그러면 졸업요건이 19년도 기준인지 21년도 기준인지 궁금합니다.
QW8	일본취업 하는 방법 전공은 소프트웨어
QW9	무료 토익 특강은 없나요
QW10	한국 장학재단 장학금 지원 조건

평가 웹사이트에서 Table 6의 10개의 키워드를 가지고 검색을 수행하였다. 랭킹 알고리즘의 결과를 측정하기 위해 서 nDCG(Normalized Discounted Cumulative Gain)와 Average Precision@K[21]를 사용하였다. nDCG는 검색 시스템과 랭킹 알고리즘의 성과를 평가하기 위해 가장 많이 사용하는 평가지표이다. nDCG는 문서의 순서와 관련성의 중요성을 고려한 지표로 검색 결과가 상위에 위치할수록 높은 값을 갖는다. nDCG는 랭킹된 결과의 품질을 직접적으로 평가할 수 있기 때문에 실용적인 기준을 제공한다.

Precision@K는 랭킹 시스템의 성능을 평가하기 위해 사용하는 지표로 랭킹된 상위 K개의 항목 중 실제로 사용자가 관심을 가진 항목의 비율을 측정한다. 이는 랭킹 리스트의 정확성을 평가하고, 추천한 랭킹이 얼마나 관련성 있는 항목을 추천하는지 평가한다. 본 논문에서는 K의 1~5까지 값을 평균으로 계산한 Average Precision@K(K:5)를 사용하였다. 키워드 10개로 검색을 수행 후에 3개의 랭킹 알고리즘의 nDCG와 aP@5의 값은 아래와 같다.

Table 7. nDCG result

QW	KNU	BM25	RNTrans BM25
QW1	0.036	0.678	0.885
QW2	0.252	0.420	0.566
QW3	0.032	0.782	0.902
QW4	0.044	0.387	0.422
QW5	0.032	0.476	0.532
QW6	0.176	0.604	0.624
QW7	0.034	0.153	0.329
QW8	0.034	0.657	0.772
QW9	0.032	0.880	0.900
QW10	0.193	0.613	0.685

위의 테스트 결과를 질의어 10개의 평균으로 nDCG를 계산해 보면 KNU의 nDCG는 8.65%, BM25는 56.50%, RNTrans BM25는 65.88%이다. BM25가 KNU에 비해 nDCG 평가 지표가 47.85% 높은 것을 볼 수 있다. 본 논문에서 제시한 RNTrans BM25는 BM25 대비하여 nDCG 평가지표가 9.38% 향상됨을 알 수 있다.

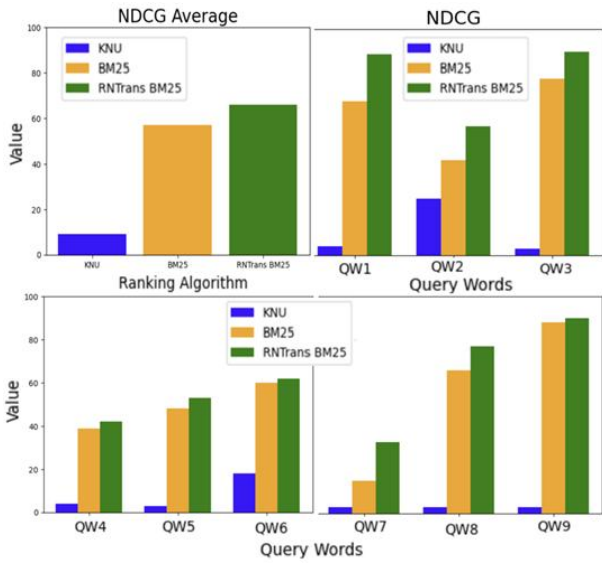


Fig. 6. Graph of nDCG results for each query

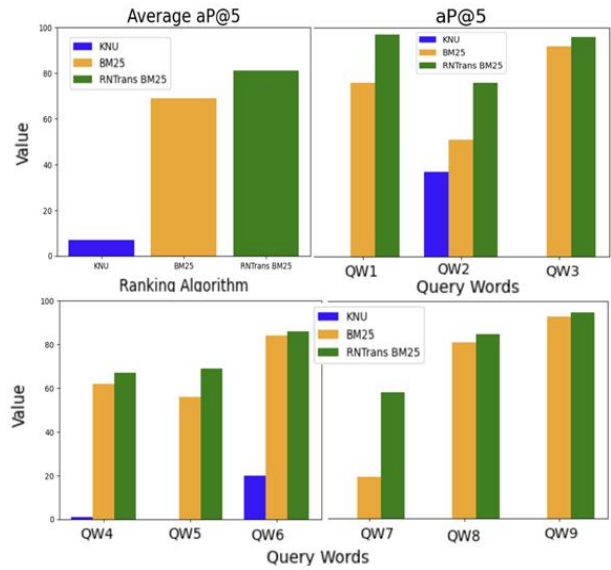


Fig. 7. Graph of aP@5 results for each query

그러면 Precision@1, Precision@2, Precision@3, Precision@4, Precision@5의 평균을 구한 Average Precision@5(aP@5)를 비교해 보자

Table 8. Average Precision@5 result

QW	KNU	BM25	RNTrans BM25
QW1	0	0.758	0.970
QW2	0.365	0.506	0.761
QW3	0	0.915	0.956
QW4	0.004	0.621	0.672
QW5	0	0.561	0.691
QW6	0.195	0.841	0.857
QW7	0	0.192	0.578
QW8	0	0.808	0.849
QW9	0	0.933	0.946
QW10	0.154	0.773	0.811

위의 테스트 결과를 질의어 10개의 평균으로 aP@5을 계산해 보면 KNU의 aP@5는 7.18%, BM25는 69.08%, RNTrans BM25는 80.90%이다. BM25가 KNU에 비해 aP@5 평가지표가 61.90% 높은 것을 볼 수 있다. 본 논문에서 제시한 RNTrans BM25는 BM25 대비하여 aP@5 평가지표가 11.82% 향상됨을 알 수 있다.

이 결과를 통해 BM25 랭킹 알고리즘에 Resnet-Transformer 모델의 분류 가중치를 적용한 RNTrans BM25 모델이 기존의 BM25 모델에 비해 nDCG 평가지표는 9.38%, aP@5 평가지표는 11.82%가 향상된 것을 볼 수 있다.

2. Search Performance Optimization Methods

결과를 통해 본 논문에서 제안한 방법이 BM25 랭킹 알고리즘에 비해 성능이 향상됨을 알 수 있다. 하지만 논문에서 구현된 시스템은 정확도를 높이는 것에 초점이 되어 있다. 그래서 현장에서 바로 적용하기에는 속도 측면에서 실용성이 떨어지는 점이 있다. 그 이유는 검색 요청 시마다 게시글을 DB에서 가져오므로 CPU 및 메모리 사용량이 높다. 특히, 랭킹 작업을 위해 검색어가 포함된 모든 관련 게시글을 가져와야하기 때문에 MySQL의 성능이 자주 병목이 된다. 이러한 문제를 해결하기 위해 Elasticsearch 나 AWS OpenSearch의 도입을 고려할 수 있다. 이러한 검색 엔진은 BM25 알고리즘을 지원한다. 그리고 효율적인 검색을 위해 Skip List를 사용하여 모든 게시글을 읽지 않고도 상위 n개의 게시글을 빠르게 추출할 수 있어서 자원 사용량을 줄일 수 있다. 또한, 이러한 검색 엔진은 게시글 저장과 검색의 역할을 분리할 수 있어 전체 시스템의 효율성을 높인다. 그리고 메모리 캐싱 기능을 통해 최근 검색된 쿼리와 결과를 저장하여 검색 성능을 향상시킬 수 있다. 이러한 점을 보완하여 실제 웹사이트에서 본 논문에서 제안한 방법을 적용한다면 속도의 제약 없이 정확한 검색 결과를 사용자들에게 제공할 것으로 기대된다.

V. Conclusions

여러 웹사이트에서 제공하는 게시글, 상품평에 대한 검색 결과의 정확도를 높이는 것은 사용자에게 중요한 일이다. 사용자 질의에 대한 검색 결과를 향상하기 위해 문맥을 잘 파악하는 트랜스포머 모델과 국소적인 특징을 잘 파악하는 ResNet 모델을 결합한 ResNet-Transformer 모델을 BM25 랭킹 알고리즘의 가중치로 적용하였다. 테스트 결과 nDCG 평가지표에서는 BM25 대비 9.35%가 향상되었고, 평균 Precision@5 평가지표에서는 BM25 대비 11.82%가 향상되었다. 이를 통해 본 논문에서 제안한 방법이 BM25 랭킹 알고리즘의 정확도를 향상시켰음을 보였다. 본 논문에서 제안한 방법을 웹사이트의 검색창에 적용 시에 사용자들의 질의어에 유사한 게시글과 상품평을 더 정확하게 제공할 수 있을 것이다.

REFERENCES

- [1] Y. Kim, "Convolution Neural Networks for Sentence Classification," Computer Science and Computation Language, Sep 2014. DOI: <https://doi.org/10.48550/arXiv.1408.5882>
- [2] A. Conneau, H. Schwenk, Y. L. Cun and L. Barrault, "Very Deep Convolutional Networks for Text Classification," Computer Science and Computation Language, Jan 2017. DOI: <https://doi.org/10.48550/arXiv.1606.01781>
- [3] H. Han, X. Bai and J. Liu, "Attention-based ResNet for Chinese Text Sentiment Classification," Advances in Computer Science Research, Vol. 80, Feb 2018. DOI: 10.2991/csece-18.2018.108
- [4] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," The Computer Vision and Pattern Recognition, Dec 2015. <https://doi.org/10.48550/arXiv.1512.03385>
- [5] J. Wang, L. C. Yu, K. R. Lai and X. Zhang, "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model," 2016 Association for Computational Linguistics, pp. 225-230, Berlin, Germany, Aug 2016.
- [6] H. Y. Park and K. J. Kim, "Sentiment Analysis of Movie Review Using Integrated CNN-LSTM Mode," Journal of Intelligence and information system, Vol. 25, pp. 141-154, Dec 2019.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, Jun 2017. <https://doi.org/10.48550/arXiv.1706.03762>
- [8] C. E. Benarab and S. Gui, "CNN-Trans-Enc: A CNN-Enhanced Transformer-Encoder On Top Of Static BERT representations for Document Classification," Journal of Korea Design Knowledge, Vol. 33, pp. 401-409, March 2015.
- [9] AG News, <https://paperswithcode.com/dataset/ag-news>
- [10] Kangnam University Homepage, <https://web.kangnam.ac.kr>
- [11] Nori Korean morphological analyzer, <https://esbook.kimjmin.net/06-text-analysis/6.7-stemming/6.7.2-nori>
- [12] S. A. Saqqa and A. Awajan, "The Use of Word2vec Model in Sentiment Analysis: A Survey," Association for Computing Machinery, Cairo, Egypt, Dec 2019. DOI:<https://doi.org/10.1145/3388218.3388229>
- [13] Jsoup, <https://jsoup.org/>
- [14] Spring Boot, <https://spring.io/projects/spring-boot>
- [15] Thymeleaf, <https://www.thymeleaf.org/>
- [16] Website on the front-end, <https://www.knusearch.site/search>
- [17] User's Guide of Flask, <https://flask.palletsprojects.com/>
- [18] Representational State Transfer (REST) architecture , https://ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [19] REST API Tutorial, <https://restfulapi.net/>
- [20] A website that evaluates search results, <https://knusearch.site/research>
- [21] K. Jarvelin and J. Kekalainen, "Cumulated Gain-based Evaluation of IR Techniques," ACM Transactions on Information System, Vol. 20, No. 4, pp. 422-446, Oct 2002.

Authors



Hong-Ju Yang received the B.S. degrees in Computer Science from Kangnam University, Korea, in 2024. Hong-Ju Yang conducted research on improving search algorithms by combining the BM25 ranking algorithm with

ResNet-Transformer models. Her interests include AI-driven search optimization, data structures, and web backend development.



In-Yeop Choi received the B.S. degrees in Computer Science from Hanyang University in 2001 and the Integrated Master's-PhD degree in Computer Science from Korea University, Korea, in 2023.

Dr. Choi joined the faculty of the Department of Computer Science at Kangnam University, Gyeonggi-do, Korea, in 2023. He developed embedded systems at the company for 15 years. He is currently a Professor in the Department of Computer Science, Kangnam University. He is interested in On-Device AI, Embedded AI and Multimodal small LLM, and TinyML.