BERT 임베딩과 선택적 OOV 복사 방법을 사용한 문서요약

Automatic Text Summarization Based on Selective OOV Copy Mechanism with BERT Embedding

저자
(Authors)

이태석, 강승식
Tae-Seok Lee, Seung-Shik Kang

출처
(Source)

정보과학회논문지 47(1), 2020.1, 36-44(9 pages)
Journal of KIISE 47(1), 2020.1, 36-44(9 pages)

발행처
(Publisher)

한국정보과학회
The Korean Institute of Information Scientists and Engineers

URL

http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09289736

APA Style

이태석, 강승식 (2020). BERT 임베딩과 선택적 OOV 복사 방법을 사용한 문서요약. 정보과학회논문지, 47(1), 36-44

ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 47, No. 1, pp. 36-44, 2020. 1
https://doi.org/10.5626/JOK.2020.47.1.36

BERT 임베딩과 선택적 OOV 복사 방법을 사용한 문서요약

(Automatic Text Summarization Based on Selective OOV Copy Mechanism with BERT Embedding)

이 태 석 †　　강 승 식 ††
(Tae-Seok Lee)　　(Seung-Shik Kang)

요 약 문서 자동 요약은 주어진 문서로부터 주요 내용을 추출하거나 생성하는 방식으로 짧게 줄이는 작업이다. 생성 요약은 미리 생성된 워드 임베딩 정보를 사용한다. 하지만, 전문 용어와 같이 저빈도 핵심 어휘는 임베딩 사전에서 누락되는 문제가 발생한다. 문서 자동 요약에서 미등록 어휘의 출현은 요약 성능을 저하시킨다. 본 논문은 Selectively Pointing OOV(Out of Vocabulary) 모델에 BERT(Bidirectional Encoder Representations from Transformers) 형태소 임베딩, Masked OOV, 형태소-to-문장 변환기를 적용하여 미등록 어휘에 대한 선택적 복사 및 요약 성능을 높였다. 기존 연구와 달리 정확한 포인팅 정보와 선택적 복사 지시 정보를 명시적으로 제공하는 선택적 OOV 포인팅 복사 방법과 함께 BERT 임베딩과 OOV 랜덤 마스킹, 형태소-문장 변환기를 추가하였다. 제안한 OOV 모델을 통해서 자동 생성 요약을 수행한 결과 단어 재현 기반의 ROUGE-1이 54.97 나타났으며, 또한 어순 기반의 ROUGE-L이 39.23으로 향상되었다.

키워드: BERT, OOV 랜덤 마스킹, 형태소-문장 변환기, 문서요약, 미등록 단어 인식, 딥러닝, 생성요약

Abstract Automatic text summarization is a process of shortening a text document via extraction or abstraction. Abstractive text summarization involves using pre-generated word embedding information. Low-frequency but salient words such as terminologies are seldom included in dictionaries, that are so called, out-of-vocabulary (OOV) problems. OOV deteriorates the performance of the encoder-decoder model in the neural network. To address OOV words in abstractive text summarization, we propose a copy mechanism to facilitate copying new words in the target document and generating summary sentences. Different from previous studies, the proposed approach combines accurately pointing information, selective copy mechanism, embedded by BERT, randomly masking OOV, and converting sentences from morpheme. Additionally, the neural network gate model to estimate the generation probability and the loss function to optimize the entire abstraction model was applied. Experimental results demonstrate that ROUGE-1 (based on word recall) and ROUGE-L (longest used common subsequence) of the proposed encoding-decoding model have been improved at 54.97 and 39.23, respectively.

Keywords: BERT, random masked OOV, morpheme-to-sentence converter, text summarization, recognition of unknown word, deep-learning, generative summarization.

· 이 논문은 2017년 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017R1D1A1B03036409). 또한, 이 논문은 2017년 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017M3C4A7068186)
· 이 논문은 한국과학기술정보연구 주요사업 과제의 지원을 받아 수행된 연구임(K-20-L01-C07-S01)

† 정 회 원 : KISTI 융합서비스센터 책임연구원
tsyi@kisti.re.kr
†† 종신회원 : 국민대학교 소프트웨어학부 교수(Kookmin Univ.)
sskang@kookmin.ac.kr
(Corresponding author임)

논문접수 : 2019년 4월 15일
(Received 15 April 2019)
논문수정 : 2019년 9월 27일
(Revised 27 September 2019)
심사완료 : 2019년 11월 14일
(Accepted 14 November 2019)

정보과학회논문지 제47권 제1호(2020. 1)

1. 서 론

스마트 기기 이용이 증가하면서 소셜 미디어, 인터넷 뉴스/댓글, 개인 블로그를 통해 텍스트 자료가 대량으로 생산되고 있어 텍스트를 분석하고 활용하는 방법에 대한 연구가 늘고 있다[1]. 그 중 인공지능 챗봇은 일상 대화뿐만 아니라 문서를 읽고 문서내용에 대한 질문에 답변을 하는 전문가 컨설팅 수준으로 발전하고 있다. 따라서 문서로 되어 있는 내용을 읽고 요약하여 정보를 제공하기 위한 문서 자동 요약(Automatic Text Summarization)에 대한 수요가 증가하고 있다.

생성 요약은 입력 문서 안에 문서 전체를 대표하는 문장이 없을 경우 매우 유용하다. 이 경우 입력 문서의 내용을 압축하여 원문에 없는 새로운 문장을 생성하여 요약을 한다. 입력 문서에 대응한 요약 문서를 학습하기 위해 신경망 모델을 활용한다[2,3]. 신경망 모델은 대량의 사례를 학습하여 유사한 내용이나 구조의 문서의 숨겨진 패턴에 따라 요약문을 생성한다. 이때 문장의 생성 규칙인 문법적인 정보와 의미적인 정보가 함께 학습된다[4]. 언어의 순차정보를 함축하는 벡터로 인코딩하고 그 벡터를 디코딩하는 모델을 사용한다. 이러한 인코딩-디코딩 모델은 sequence-to-sequence(seq2seq) 모델이라고 하며 기계번역, 자동띄어쓰기, 문서분류, 오타교정 등 다양하게 활용된다[5,6].

기계독해는 문서를 읽고 문서 내용 중 일부분을 포인팅하여 답변하는 분야이다. 요약은 문서만 입력하지만, 기계독해는 질문이 추가로 입력되는 것이 다른 점이다. 그러나, 어떤 질문에 대해 최적의 정보를 추출하는 측면에서는 요약과 다르지 않다. 그러므로 기계독해의 최신 기술을 요약에 활용하는 것이 가능하다.

최근 스탠퍼드 대학의 기계독해 대회인 SQuAD(The Stanford Question Answering Dataset) 기록에서 상위 10위 모델은 모두 BERT(Bidirectional Encoder Representations from Transformers)를 쓰는 모델이다. SQuAD 1.1은 10만개의 질문과 답(위치정보)에 대해 기계학습을 통해서 이미 인간보다 우수한 성능을 보이고 이 대회는 종료되었다[7]. SQuAD 2.0은 1.1에 답이 없어 답변하기 어려운 질문을 포함하여 5만개의 새로운 질문이 추가되었다. 인간 독해 수준까지 기계 독해 성능이 빠르게 발전하고 있다.

BERT는 구글이 발표한 언어 모델이다. 이 언어 모델은 워드 임베딩의 성능을 향상시킨 것으로 문장의 각 단어를 벡터로 변환해주는 기능을 한다. 가장 큰 특징은 맥락에 따른 임베딩이 가능하다. 즉, 문장의 위치정보와 결합하여 동일한 단어에 대해서도 그 쓰임에 따라 다른 벡터로 표현한다는 것이다. 위키 백과사전과 같은 대형 코퍼스를 비지도 방식으로 학습하여 문서분류, 요약, 기계독해 등 언어처리관련 여러 분야에서 활용 가능한 모델이다[8].

신경망을 이용하는 생성 요약은 학습 데이터에 출현한 단어를 중심으로 미리 임베딩된 사전을 이용하지만, 전문용어와 같이 저빈도 핵심어는 미등록 단어가 된다. 요약 대상 문서에 미등록 단어(Out-of-Vocabulary, OOV)가 있을 경우, 사람은 주변문장의 의미와 개인적인 경험을 기반으로 유추하거나 관련정보를 확인하여 모르는 단어를 이해하고 그 중요성을 파악하여 문서를 요약한다. 경우에 따라서 미등록 단어의 중요도가 낮을 수도 있다. 이를 해결하기 위한 기존 연구는 copy mechanism을 사용하는 것이다[9]. 미등록 단어를 포인팅된 입력문서에서 가져오는 방식이다. copy mechanism의 포인팅 성능을 올리고 생성과 포인팅을 결정하는 게이트를 추가하는 방식으로 발전하였다. 이 게이트를 통해 입력문서와 요약문에 출현하는 핵심 단어를 선정하고 포인팅 위치정보와 단어 생성을 선택적으로 할 수 있다.

본 논문의 제안 방법은 첫째 사전 학습된 언어모델인 BERT를 이용하여 한국어 형태소 단위로 임베딩하고, 둘째 학습 데이터의 OOV 마스크를 랜덤으로 실시하여 미등록 단어의 출현을 인식시키며, 셋째 한국어 형태소 단위로 생성된 요약문의 띄어쓰기 문제를 해결한 선택적 OOV 복사 신경망 모델을 적용한 문서요약을 특징으로 하고 있다.

2. 관련 연구

OOV문제는 문서요약뿐만 아니라, 기계 자동 번역에서도 발생하며 이를 해결하기 위한 다양한 방법이 사용되었다. 주요 방법으로는 어휘 패턴 검색 및 분석으로 후보 단어 목록 생성 방법[10], 중요도 기반 샘플링을 통해 대용량 어휘를 유지하는 방안[11], 단어를 서브워드 조각으로 나누어 어휘를 처리하고 조합하는 방식[12], 번역 문장의 단어를 포인팅하고 복사하여 대체하는 방식[13]을 사용하였다.

어휘 패턴 검색으로 후보 단어를 선별하는 방식은 신경망 모델에서 쓰지 않고 있으며, 대용량 어휘를 쓰는 것은 신조어 및 전문용어가 많은 분야에 적용하기 어렵다. 또, 부분 단어로 조합하는 방식은 BERT에서도 사용하는 방식이나 한글의 의미와 기능과 맞지 않게 분해되고 조합될 수 있으므로 OOV 문제를 완전히 해결하지 못한다. 따라서 본 논문에서는 포인팅하고 복사하는 방식이 요약에서도 유용할 것으로 보고 이와 관련된 모델을 활용하였다.

2.1 Copy mechanism seq2seq 모델

요약 생성에서 필요한 어휘를 입력문서에서 찾아 복

사(copy)하는 방법을 copy mechanism seq2seq 모델이라고 한다[9,14]. 이 모델의 문서요약 생성방식은 디코딩 과정에서 copy mechanism을 사용하여 OOV를 선택한다.

요약문 모든 단어 생성에 copy attention 점수를 합산하여 단어를 결정하는 것은 copy attention 점수가 높은 것이 입력문서의 특정부분의 영향도가 큰 것일 뿐이기 때문에 정상적으로 생성해야 할지 입력문서에서 복사할지 결정하는 기준이 될 수 없다. 따라서 copy mechanism의 경우 요약문의 단어를 생성해야 함에도 입력문서에서 어휘를 복사하는 경우가 발생한다.

copy mechanism 개선 방법으로 입력문서 어휘에서 복사를 할지 또는 생성할지 판단할 수 있는 기준(g)과 OOV 위치(p)를 제공하여 위치정보에 대한 오차를 모델에 반영하였다[15,16]. 필요한 어휘만을 선택적으로 copy mechanism을 실행함으로써 정확도를 높이고 어휘 사전에 있는 단어의 경우 copy attention 점수를 고려하지 않는다.

2.2 BERT 임베딩

BERT언어 모델은 Transformer의 인코더 부분만 가지고 단어를 임베딩한다. BERT 입력 임베딩은 Positional Encoding을 쓰지 않고 대신 Position Embedding을 쓴다. 단어의 위치에 따라 one-hot 인덱스 임베딩을 한다. 그리고 문장의 임베딩과 단어 토큰 임베딩 순으로 더하여 입력을 만든다[8].

구글에서 공개한 BERT의 인코더 블록의 개수는 Base 모델이 12개, Large 모델이 24개로 되어있다. 이는 입력 시퀀스 전체를 여러 블록으로 적층하여 계산하는 것을 의미한다. 인코더 블록의 수가 많을수록 단어 사이에 보다 복잡한 관계를 더 잘 표현할 수 있다.

3. BERT 임베딩 및 OOV 선택을 이용한 자동요약

문서 자동요약 BERT OOV모델은 그림 1과 같이 구성하였다. OOV 모델은 선택적으로 생성과 포인팅을 통해 단어를 복사하도록 하였다. OOV모델 입력은 형태소 단위로 토큰화한 BERT 임베딩 결과를 Selectively Pointing OOV 모델로 입력하였다.

OOV 모델의 출력결과는 형태소 단위로 구분된 토큰을 출력한다. 성능테스트를 위해서는 문장으로 생성하여야 하기 때문에 형태소 출력을 문장으로 조합하는 변환기를 사용하였다.

3.1 BERT 언어 모델 사전 학습

BERT 언어 모델은 사전 학습을 통해 만들었다. 그러기 위해서 Masked Language Model, Next Sentence Prediction 학습을 위한 데이터를 생성하였다. 사용할 어휘사전파일과 문장에서 [MASK] 태그로 치환하는 비율 등의 옵션을 지정하여 생성한 결과 46,615개의 BERT 학습 데이터를 생성하였다.

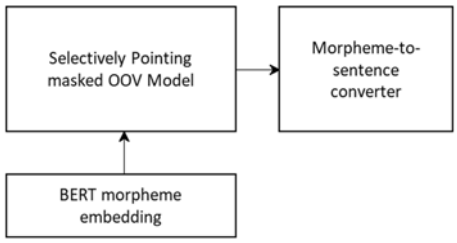

그림 1 BERT OOV 모델 전체 구성
Fig. 1 Overall configuration of the BERT OOV model

BERT 언어 모델을 만들기 위해 과정은 학습횟수, 배치크기, 학습율 등과 같은 관련 옵션을 주고 학습을 수행하였다. 논문의 평균적인 문장길이를 고려하여 128 어절을 최대 순차 길이로 정했다. Masked Language Model을 위한 값으로 0.15를 사용하여 전체단어의 15% 정도만 마스킹 하도록 하였다.

BERT 언어모델 사전학습을 위해서 필요한 bert_config.json 파일은 실험에 사용할 장비의 성능을 고려하여 히든 크기, Attention 헤드 수, 히든 계층 수를 각각 128, 2, 2로 설정하였다. BERT 임베딩에서 사용하는 어휘는 MECAB 한국어 형태소 분석기로 분해한 어절을 사용하였다. 300만번의 사전학습 결과 그림 2와 같이 Masked Language Model의 정확도는 0.89이고, Next Sentence Prediction 1.0으로 학습되었다.

```
global_step = 3,000,000
loss = 0.43
masked_lm_accuracy = 0.89
masked_lm_loss = 0.43
next_sentence_accuracy = 1.0
next_sentence_loss = 1.45e-05
```

그림 2 Pre-training 수행 결과
Fig. 2 Results of pre-training

3.2 Masked OOV 학습

OOV 학습을 위하여 저자의 키워드를 <unk> 태그로 치환하고 학습하는 방법을 사용하였다. 그러나 모든 OOV 대상 키워드가 모두 <unk> 태그로 치환되면 모델 내부에 <unk> 출현 여부에 대한 패턴을 학습하여 학습 자체가 무의미해질 수 있다. <unk> 태그의 전후에 나타나는 문장과 단어의 문맥이 학습되어야 한다. BERT에서 입력값을 의도적으로 왜곡하는 Masked Language Model 기법에 착안하여 그림 3과 같은 Masked OOV 학습 방법을 고안하였다. P1 확률로 <unk> 치환을 랜덤하게 수행하고, P2 확률로 일부는 원래 단어를 그대로 보여주고, P3 확률로 랜덤 단어로 치환하는 방식이다.

- P1 : substituting with 〈unk〉
  ex) NAND형 메모리 압축
  → <unk>형 메모리 압축
- P2 : keeping the original word
  ex) NAND형 메모리 압축
  → NAND형 메모리 압축
- P3 : replacing with a random word
  ex) NAND형 메모리 압축
  → 계산형 메모리 압축

그림 3 Masked OOV 학습 방법
Fig. 3 Training method of the masked OOV

표 1 MOOV적용과 문서 요약 성능
Table 1 Performances of text summarization with the masked OOV

Model	ROUGE		
	1	L	SU4
LSTM g_t+p_t +MOOV(8:2:0)	48.89	26.58	19.74
LSTM g_t+p_t +MOOV(8:1:1)	47.51	26.28	20.24
LSTM g_t+p_t +MOOV(9:1:0)	51.27	32.44	22.00

P1:P2:P3 치환 비율에 따라 학습효과가 어떻게 변화되는지 실험한 결과 표 1과 같다. 학습 데이터에서 <unk> 태그 처리를 80% 수행하고 20%는 정상 어휘로 입력하였을 때와 10%는 정상 어휘로 10%는 랜덤 어휘로 했을 때는 성능향상이 미미했으나, <unk> 태그 처리를 90% 수행하고, 10%는 정상 어휘로 입력하고, 랜덤 어휘를 입력하지 않을 경우 ROUGE-1 51.27, ROUGE-L 32.44로 크게 향상되었다. 모델의 변화는 없이 <unk> 태그의 일부를 정상 어휘로 보여주는 것이 모델 학습에 긍정적인 영향을 주는 것을 확인하였다. 하지만 10% 정도의 랜덤 노이즈 어휘를 보여주는 것은 성능향상에 도움을 주지 못했다.

실험결과 9:1:0 치환 비율을 적용하였을 때 가장 높은 성능을 보였다. BERT Masked Language Model과 같이 다른 단어로 치환하는 것이 Masked OOV 학습에 큰 효과가 있었다. OOV 학습에서 노이즈 효과는 원래의 단어를 치환하는 것만으로 충분하였다. 이것은 BERT가 단어의 고차원 벡터를 계산하는 것이고, OOV 학습은 문맥에 의존한 정확한 위치정보와 생성/복사에 대한 선택을 지시하는 정보를 계산하는 것이기 때문이다.

3.3 선택적 OOV 단어 복사 모델

실험에 사용한 자동 요약 신경망은 그림 4와 같은 LSTM셀을 쓰는 seq2seq 모델이다. 인코딩 과정은 입력 문서에 있는 각 문장의 단어를 순차적으로 입력 받아

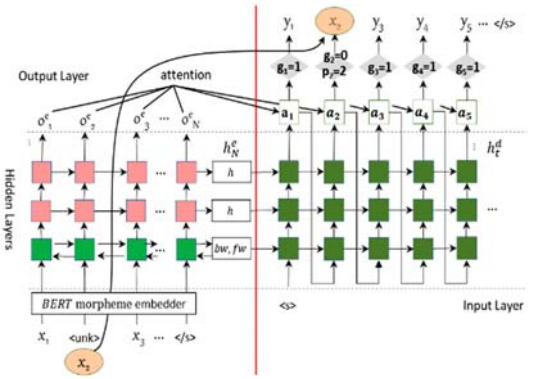

그림 4 Selectively pointing OOV 모델
Fig. 4 Selectively pointing of the OOV model

입력 문서를 벡터로 표현하는 역할을 한다. 디코딩 과정은 인코딩 단계의 최종 히든 상태를 초기 값으로 시작하여 요약문의 각 단어를 순차적으로 풀어내는 역할을 한다. 또한 디코더에서 지시기(p) 정답, 선택기(g) 정답, 요약 단어 생성 오차를 구하고 손실함수를 통해 모델이 최적화되도록 하였다.

입력층에 형태소 분리가 된 단어를 BERT 임베딩을 거쳐 히든층으로 입력한다. 인코딩의 첫 번째 계층은 양방향 RNN(Bidirectional Recurrent Neural Network)을 사용하였고 각 타임 스텝마다 양방향 정보를 처리하였다. 인코딩 은닉층은 양방향 계층을 포함한 3개의 LSTM 층을 두고 대응하는 3계층 디코딩 모델로 설계하였다.

y_2생성에서 g_2가 0으로 지시함으로써 포인팅을 통한 copy mechanism을 통해 OOV 단어를 요약 문장의 단어로 생성하도록 설계하였다. N은 전체 인코딩 타임 스텝수이다. x는 입력단어, y는 출력단어, p_t는 입력단어 선택 번호, h^e와 h^d는 신경망 은닉상태 행렬이다.

인코더, 디코더, OOV 포인팅 attention 신경망을 모두 학습하기 위한 추가 정보로서 g_t와 함께 핵심 단어에 대해 위치값 p_t를 학습 모델에 반영하였다. g_t의 값에 따라 0이면 포인팅을 하고, 1이면 요약 단어 생성을 하도록 하였다.

전체 모델을 식으로 표현하면 식 (1)과 같다. $P(Y/X)$는 선택적 OOV 단어 복사 모델의 입력 단어 목록 X에 대한 요약 단어 목록 Y를 생성하는 조건부 확률이다. 디코딩 진행이 t-1 번째까지 진행된 상태에서 식 (2) $P(y_t|Y_{t-1}, X)$는 요약 단어를 생성하는 확률 함수이고, 식 (3) $P(p_t|Y_{t-1}, X)$는 copy mechanism을 통한 입력단어 포인팅에 대한 확률함수이다. c는 인코딩 결과에 대한 attention weighted context vector이다.

$$P(Y|X) = \sum_t \begin{pmatrix} g_t P(y_t|Y_{t-1}, X) \\ +(1-g_t)P(p_t|Y_{t-1}, X) \end{pmatrix} \quad (1)$$

$$P(y_t|Y_{t-1}, X) = softmax\left(\tanh\begin{pmatrix} Wh_t^d + Wc_{t-1} \\ +Wy_{t-1} + b \end{pmatrix}\right) \quad (2)$$

$$P(p_t|Y_{t-1}, X) = softmax\left(\tanh\left(W\{o^e\} + Wh_t^d + b\right)\right) \quad (3)$$

$$g_t = tanh\left(Wh_t^d + Wc_t + Wy_{t-1} + b\right) \quad (4)$$

NLL(negative log likelihood)은 정답과 예측결과의 차이를 최소화하기 위한 신경망 손실함수이다. 손실함수의 계산을 위해 요약문 각 단어를 생성할 때 손실 함수(*generation_NLL*)와 입력 문서의 단어를 지시하기 위한 손실함수(*pointing_NLL*)를 유도하여 표현하면 다음과 같다.

$$\text{loss} = \sum_t \begin{pmatrix} g_t \cdot generatiing_{NLL} + \\ (1-g_t) \cdot pointing\_NLL \end{pmatrix} \quad (5)$$

식 (5)에서 *generation_NLL* 은 디코더 출력이 실제 요약문의 단어와 일치하도록 학습하는 데 사용되는 손실함수이다. *pointing_NLL* 은 요약으로 생성되어야 하는 단어가 어휘 사전에 존재하지 않는 것을 가정하여 요약 대상 문서의 단어 위치를 지시하는 값이 정확하게 되도록 하는 손실함수이다. 이 두 손실함수는 신경망 출력을 정규화하고 예측된 결과와 정답의 오차에 대한 교차 엔트로피 함수(cross entropy function)로 계산하였다.

3.4 형태소 분해 및 문장 변환기

BERT는 어휘사전 단어와 최장 길이 우선 매칭(greedy longest-match-first) 토큰화 알고리즘을 사용하고 있어 한국어 형태소 단위의 임베딩 및 생성이 되지 않는다. 따라서 한국어 형태소로 분리하여 BERT 임베딩을 수행하고 선택적 OOV 단어 복사 모델의 생성결과가 형태소 단위이므로 문장으로 복원하는 과정이 필요하다.

한국어 형태소 분해는 MECAB-KO[1]를 가지고 어근과 조사/어미를 분해한 것을 이용하였다. 조사 또는 어미가 연속적으로 불규칙하게 나오는 것을 생성단계에서 조합할 수 있도록 어근과 조사/어미를 식별할 수 있는 정보가 추가된 어휘사전을 사용하여 문장으로 변환하였다.

형식형태소에 해당하는 관계언, 선어말어미, 어말어미, 접두사, 접미사에 대한 품사에 해당하는 형태소에는 "·-o·"(종결형) 또는 "·--·"(연결형) 부호를 추가하였다. 형태소 단위로 생성된 문장을 복원할 때는 어근에 해당하는 형태소와 조사/어미 형태소를 연결하여 변환하고 다른 어근이 나타나면 띄어쓰기를 하였다. 즉, 어근과 조사/어미를 구분해 줌으로써 어근 앞에는 공백을 추가하고 연속된 조사/어미는 연결하는 방법을 사용하였다.

형태소 분해 및 문장변환 과정을 상세하게 살펴보면

(1) NAND 형 플래시메모리를 위한 플래시 압축 계층의 설계 및 성능평가
(2) 'NAND', '·형·--·', '플래시메모리', '·를·-o·', '위한', '플래시', '압축', '계층', '·의·-o·', '설계', '및', '성능평가'
(3) 'NAND', '·형·--·', ' ', '플래시메모리', '·를·-o·', ' ', '위한', ' ', '플래시', ' ', '압축', ' ', '계층', '·의·-o·', ' ', '설계', ' ', '및', ' ', '성능평가'
(4) NAND·형·--· 플래시메모리·를·-o· 위한 플래시 압축 계층·의·-o· 설계 및 성능평가
(5) NAND 형 플래시메모리를 위한 플래시 압축 계층의 설계 및 성능평가

그림 5 형태소 분해 및 문장 변환과정
Fig. 5 The morphological decomposition and sentence generation process

그림 5와 같다. (1) 입력된 문장을 (2) MECAB-KO로 형태소 분리를 하고, 형식형태소에 대한 부호를 추가한다. (3) 형태소로 분리된 리스트에서 어근에 해당하는 요소의 앞에 공백을 추가한다. (4) 리스트의 요소를 모두 이어 붙인다. (5) 이어진 문장에서 부호("·-o·", "·--·")를 제거한다. 결과적으로 (1)에서 입력된 문장과 동일한 문장을 얻을 수 있다. 이러한 방법으로 "형태소 to 문장 변환기" 모듈을 Selectively Pointing OOV 모델 생성결과에 적용하였다.

4. 실험 데이터 수집 및 가공

데이터 수집은 한국과학기술정보연구원이 운영하는 NDSL 웹[2]에서 정보과학회 논문을 수집하였다. 실험을 위해서 정제한 후, 표 2와 같이 총 3,382개의 논문을 5:2:3 비율로 학습, 검증, 평가 데이터를 무작위로 나누어 사용하였다.

4.1 핵심 단어가 포함된 학습 데이터

생성요약을 위해 필요한 학습 데이터는 입력문서에 대응하는 요약문의 집합이다. 입력문서는 뉴스 기사와 같이 여러 문단으로 나누어진 문장과 단어로 구성된다. 요약문은 입력문서의 일부 문장과 단어로 구성된 짧은 문장의 문서이다. 미등록 단어를 인식하는 요약 방법에

표 2 실험데이터 통계 정보
Table 2 Statistical results of the data

구 분	학습	검증	평가
문서수	1,723	648	1,011
문서당 평균 문장수	6.93	6.97	7.06
문서당 평균 어절수	119.29	117.47	120.21
요약문 평균 어절수	8.70	8.64	8.73

1) 은전한닢 프로젝트, http://eunjeon.blogspot.com/

2) http://www.ndsl.kr

서는 추가적으로 입력문서와 함께 핵심단어 목록이 필요하다. 핵심단어는 요약문에 쓰일 수도 있고 쓰이지 않을 수도 있다. 요약 패턴을 학습하기 위해 OOV 단어로 쓰일 핵심단어가 필요하다.

논문은 문서의 형태나 표현이 정해져 있고 가장 중요한 저자의 키워드가 있어서 문서 요약 실험 데이터로 적합하다. 따라서 본 논문에서는 논문의 제목, 초록, 저자키워드를 이용하여 요약문, 입력문서, 핵심키워드로 보고 학습 데이터를 만들었다. 논문의 제목이 논문의 내용을 가장 잘 요약하고 있다고 가정하였고 저자의 키워드가 논문에서 가장 중요한 핵심 키워드로 보았다. OOV 단어로 선정하기 위해서 저자의 키워드를 IDF를 계산하여 가장 높은 점수를 가지는 3개를 선택하였다.

4.2 학습 데이터 가공

입력문서는 논문의 제목, 초록, 키워드를 한 행에 한 개의 데이터로 변환하였다. 각 필드는 publisher(데이터 출처 코드), abstract(논문 제목), article(논문 초록), oov_keywords(저자 키워드)로 구분하였다.

입력 문서 정보로부터 데이터 변환을 통해 학습에 필요한 정보를 표 3과 같이 추가로 생산하였다. (1)에서 MECAB-KO 형태소 분석기를 사용하여 한글 텍스트를 토큰화하였다. (2)는 입력문서와 정답 요약문에 나타난 OOV 단어를 <unk> 태그로 치환하는 과정이다. OOV 단어 치환은 90%비율로 랜덤하게 수행하였다. 10%에 대해서 원래의 단어를 유지하였다. 이렇게 함으로써 OOV 출현에 대한 신경망 모델을 학습이 이루어지며, 요약문 생성시에 <unk>이면 입력문서에서 OOV 단어를 복사하는 것이 선택적으로 이루어지도록 설계하였다. (3)은 인덱싱을 위한 어휘 사전을 생성하였다. (4)는 <unk>를 실제 단어로 복원하기 위한 OOV 사전을 생성하였다. (5)는 어휘 사전을 가지고 인덱싱을 하였다. (6)은 요약문 단어 생성과 복사를 결정하기 위한 생성기 정답을 만들었다. 마지막으로 (7)은 OOV를 정확하게 지시하기 위한 지시기 정답을 만들었다.

표 3 학습 데이터 가공 과정(i, 입력; o, 출력)
Table 3 The process of making training data (i, input; o, output)

Step	Results of data conversion
(1) Tokenization by morpheme	(i) 최근 휴대용 정보기기의 사용이 급증함에 따라 NAND 형 플래시 메모리를 시스템의 보조 기억 장치로 사용하는 사례가 급증하고 있다. (o) NAND형 플래시 메모리를 위한 플래시 압축 계층의 설계 및 성능 평가.
(2) Masking randomized OOV token	(i) 최근 휴대용 정보기기의 사용이 급증함에 따라 <unk> 형 <unk> 메모리를 시스템의 보조기억장치로 사용하는 사례가 급증하고 있다. (o) <unk> 형 <unk> 메모리를 위한 <unk> 압축 계층의 설계 및 성능 평가.
(3) Create Dictionary	{“최근”:31, “휴대”:36, “용”:24, “정보”:30, “기기”:7, “의”:26, “형”:35, “메모리”:14, “를”:13, “위한”:25, “압축”:22, “.”:1, “<unk>”:2, ...}
(4) Make OOV position	(i) {“NAND”:13, “플래시”:15} (o) {“NAND”:1, “플래시”:3}
(5) Indexing	(i) {31, 26, 24, 30, 7, 26, ... ,1} (o) {2, 35, 2, 14, 13, 25, 2, 22, 4, 26, ..., 1}
(6) Pointer g_t	(i) {0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1}
(7) Selector p_t	(o) {13, 0, 15, 0, 0, 0, 15, 0, 0, 0, 0, 0, 0, 0}

5. 실험 및 성능 평가

OOV 생성요약을 위한 실험 과정은 그림 6과 같다. 수집된 데이터는 OOV 모델 학습을 위한 데이터와 BERT 언어 모델을 학습하기 위한 데이터로 변환을 하였다. 다음으로 사전학습(Pre-training)을 통해 BERT 모델을 생성하였다. 생성된 BERT 모델을 초기상태로 시작하여 OOV 모델 학습을 통해 미세조정(Fine-tuning)하였다. 마지막으로 ROUGE 테스트로 성능을 평가하였다.

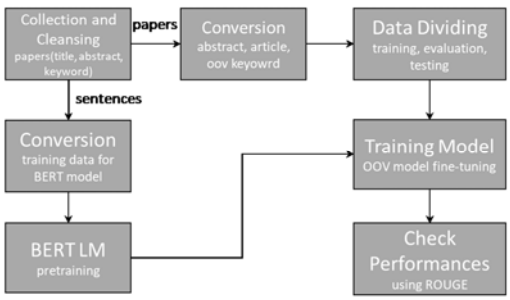

그림 6 BERT OOV 모델 실험 과정
Fig. 6 The experimental process of the BERT OOV model

5.1 실험 설정

요약문의 성능 평가는 생성된 요약문이 정답과 비교하여 정보누락 여부가 중요하다. 문서 이해 및 요약 관련 영어권 학술 컨퍼런스인 DUC(Document Understanding Conference)에서 사용하고 있는 ROUGE (Recall-Oriented Understudy for Gisting Evaluation) 방식을 사용하였다[17]. ROUGE는 정답 단어의 출현, 문장의 어순 일치 길이, 어순 일치 방식 등에 따라 다양한 변형이 존재한다.

실험은 표 4와 같은 제원의 PC 서버 장비에서 수행했다. 전체 학습 시간은 1일 19시간이 소요되었다.

표 4 실험 장비 제원
Table 4 Experimental equipment specification

Equipment	Specification
CPU	Intel i7-4790K 4Ghz 8 core
Memory	128GB
OS	Ubuntu 18.04 LTS
Python	Python 3.6.8
Tensorflow	tensorflow-gpu 1.13.1
GPU	GeForce GTX TITAN Xp
GPU Memory	12GB
GPU Library	CUDA 10.0

표 5 하이퍼 파라미터
Table 5 The hyper parameter

Variables	Values
BERT layers	2
Hidden node size for BERT	128
Mini batch size	32
LSTM layers	3
Max encoding timesteps	600
Max decoding timesteps	50
Min input length	2
Hidden node size for LSTM cell	200
Embedding dimension	100
Dropout	0.8
Using layer normalization	yes
Using dropout	yes
Using batch-shuffle	yes
Optimization function	adam

표 5는 학습수행에 쓰인 하이퍼 파라미터이다. 모델의 성능을 높이기 위해 드롭아웃과 계층정규화를 사용하였으며, 인코딩과 디코딩 LSTM 셀을 3계층으로 실행하였다. BERT임베딩은 2계층을 사용하고 사전학습이 끝난 모델을 사용하였다.

실험은 12만 스텝(1,600 epoch)을 수행하면서 30분마다 검증 데이터에 대한 loss를 측정하였다. 그 결과 학습 수행 5,400 스텝(730 epoch)에서 검증 데이터의 loss가 0.083으로 가장 낮게 나오고 그 이후는 오버피팅 발생으로 더 이상 성능향상이 없었다.

5.2 성능 평가

본 논문에서는 N-gram 기반의 ROUGE-N, LCS (Longest Common Subsequence)기반의 ROUGE-L, Skip-Bigram Co-Occurrence and unigram 기반의 ROUGE-SU를 이용하여 표 6과 같이 제안 모델의 성능 향상을 확인하였다.

평가 문서로 요약을 실행한 ROUGE 측정값을 비교해 보았다. LSTM search $g_t + p_t$ 방법이 GRU search + input feeding + Copy 방법[14]에서 보여준 성능보다 높았고, LSTM search $g_t + p_t$ 모델에 BERT 임베딩과 Masked OOV(MOOV)를 실시하였을 때 더 성능이 향상되었다.

표 6 문서 요약 성능
Table 6 Performances of the text summarization

Model	ROUGE		
	1	L	SU4
LSTM	40.46	22.92	16.49
LSTM p_t feeding	43.60	25.37	16.48
LSTM $g_t + p_t$ feeding	47.01	29.55	17.80
BERT+LSTM $g_t + p_t$+MOOV (9:1:0)	**54.97**	**39.23**	**23.81**

형태소 단위의 BERT임베딩과 함께 Masked OOV 90%만 실시하고 나머지 10%에 대해 정상 어휘를 적용할 때, ROUGE-1 54.97, ROUGE-L 39.23으로 가장 좋은 성능이 나왔다. 이것은 BERT 사전학습 및 Fine-tuning 과정에서 문서의 언어모델에 대한 학습효과가 반영되어 문맥에 따른 형태소 구분 어절의 벡터 표현이 잘된 효과로 해석할 수 있다. ROUGE-1은 요약문 어절의 출현 순서 중심의 성능 지표이고, ROUGE-L은 어절의 연속적 단어 출현에 대한 성능 지표이다. 제안 모델이 포인팅 기반의 LSTM search $g_t + p_t$ 방법보다 더 높은 결과가 나왔다. 성능 향상 정도는 ROUGE-1이 47.01에서 16.93% 증가하였고, ROUGE-L이 29.55에서 32.76% 증가하였다. 이것은 Masked OOV 입력과 BERT 임베딩, 형태소 처리기를 기존 모델에 추가함으로써 요약 문장의 단어를 재현하고 어순의 연속성도 같이 향상되는 것을 의미한다.

5.3 오류 분석

미등록 OOV 단어가 선택적으로 자동 요약되는 사례를 표 7과 같이 확인하였다. “마코프”, “악성코드”와 같은 OOV 단어와 함께 요약 문장이 생성되었지만, 일부 문서의 경우 핵심어를 중심으로 요약이 되지 않는 문제가 있었다. 또한 문장 구성이 문법적으로 어색한 사례가 발견되었다. 따라서 언어적인 의미와 구문 패턴을 동시에 학습할 수 있도록 개선되어야 한다.

요약 성능 개선을 위해 의미를 기반으로 질의-응답 문제를 해결하는 KorQuAD(The Korean Question Answering Dataset)[18]와 SQuAD(The Stanford Question Answering Dataset)[7]의 사례로부터 필요한 문서의 양을 예상할 수 있다. 스탠포드 대학의 NLP 그룹에서 크라우드 소싱을 통해 만든 SQuAD 1.1은 536개의 위키피디아 기사를 2만개 이상의 문단으로 분리하여 107,702개의 질의-응답 쌍을 만들었으며, 2.0은 응답 불가능한 5만개 이상의 악의적 질의-응답 쌍이 추가되

표 7 자동요약 생성 사례
(a, 정답; b, 제안모델 요약; c, selectively pointing OOV LSTM 모델 요약; d, OOV 단어)
Table 7 Examples of automatic-summary text
(a, correct answers; b, summary cases by masked OOV copy model with BERT; c, summary cases by selectively pointing OOV LSTM; d, OOV keyword)

No.	Examples
1	(a) **마코프 논리** 기반의 시맨틱 문서 검색. (b) **마코프 논리** 기반의 순위 검색 알고리즘. (c) **마코프** 웹 검색 기반의 학습. (d) **논리**; **마코프**; **감독**
	(생략) 본 논문에서는 이러한 문제를 극복하기 위해 데이터 기반의 **감독** 학습(supervised learning) 기법과 관련 온톨로지 정보를 **마코프 논리**(Markov logic)에 기반하여 결합한다. (생략)
2	(a) 서비스 기반 모바일 어플리케이션의 **MVC 아키텍처** 및 적용 사례연구. (b) 서비스 기반 모바일 기반의 **MVC 아키텍처** 서비스 개발. (c) 스마트폰 기반 동적 소프트웨어의 동적 **아키텍처** 테스트 프레임워크 개발. (d) **아키텍처**; **디바이스**; **MVC**
	(생략) 모바일 **디바이스**의 자원 제약성으로 인해 복잡도가 높은 어플리케이션에는 한계를 가지고 있다. Mode l View Control (**MVC**) **아키텍처**는 다양한 어플리케이션 설계에 널리 사용되고 있지만, 서비스 기반의 모바일 어플리케이션의 특징을 모두 반영하지 못한다. (생략)
3	(a) 동적으로 갱신가능한 XML 데이터에서 **레이블 재작성**하지 않는 **원형** 레이블링 방법. (b) 동적 효율적인 XML 검색에 대한 **레이블**적 변환을 사용한 안전한 **레이블** 질의처리 모델. (c) 효율적인 XML 질의 처리를 위한 적응형 경로 인덱스. (d) **레이블**; **원형**; **재작성**
	(생략)그러나 **레이블 재작성** 비용, **레이블** 저장을 위한 큰 저장공간 할당 등의 문제점이 있다. 이러한 문제점은 새로운 데이타가 지속적으로 삽입될 경우 더욱 심화된다. (생략) 또한 실험을 통해 제안하는 **원형** 레이블링 방법의 우수성을 보인다. (생략)
4	(a) 사무실 이벤트 검색을 위한 베이지안 네트워크 기반 사용자 **선호도 모델링**. (b) 순수 접근 정보 검색을 위한 예제 기반의 사용자 **선호도** 검색. (c) 실내 위치 서비스 기반 웹 프로파일을 위한 베이지안 네트워크 검색 기법. (d) **모델링**; **동영상**; **선호도**;
	(생략) 최근 웹사이트에서 제공하는 사용자 **모델링** 서비스는 텍스트 기반 페이지 구성이나 추천 검색 등에만 국한되어 있는 단점이 있다. 본 논문에서는 사용자 **모델링** 기법을 **동영상** 검색에 적용하기 위해 사용자의 **선호도**를 베이지안 네트워크로 **모델링**하고, 추론된 확률 값을 검색에 반영하는 방법을 제안한다. (생략)
5	(a) **안드로이드 악성코드** 분석 및 기계학습 기법을 이용한 탐지 방법. (b) **스마트폰 악성코드** 자질 학습에 기반한 자동 문서 영상의 구현. (c) **악성코드** 데이타 **악성코드**를 사용한 공격적인 자동 분석 기법. (d) **스마트폰**; **안드로이드**; **악성코드**
	(생략) 국내의 다양한 **스마트폰** 운영 체제 중, 특히 **안드로이드**의 경우 오픈소스 정책 및 다양한 기기의 보급을 통해 사용자가 증가하고 있다. 이에 따라 **스마트폰** 사용자를 노리는 **악성코드** 또한 증가하는 추세이다. 현재 대부분의 **안드로이드** 용 **악성코드** 탐지 프로그램이 사용하는 방법의 경우, 위변조 혹은 새로운 **악성코드**에 대응이 어렵다는 문제가 존재한다. (생략)

었다. LG CNS가 구축한 KorQuAD 2.0[3]의 전체 데이터는 47,957개의 위키피디아 기사로부터 102,960개의 질의-응답 쌍을 만들었다. 따라서 품질이 좋은 10만개 이상의 요약 사례와 사전에 학습된 언어모델이 필요하다.

6. 실험 및 성능 평가

본 연구는 Selectively Pointing OOV 모델에 BERT 형태소 임베딩, Masked OOV, 형태소-to-문장 변환기를 적용하여 미등록 어휘에 대한 선택적 복사 및 요약 성능을 높였다.

저자 키워드를 이용하여 주요 단어의 위치에 대한 학습은 입력문서의 단어를 포인팅하여 요약문을 생성하는 방법을 사용하였다. 포인팅을 통한 단어 복사와 요약 문장의 생성성능을 함께 향상시키기 위해 포인팅 위치와 생성 여부를 결정하는 신경망 게이트를 학습하였다. 이와 함께 사전 학습된 언어모델인 BERT 임베딩을 사용하고 OOV 랜덤 마스킹 방식을 사용함으로써 <unk> 태그에 대한 생성 규칙에 대한 학습이 이루어졌다. 형태소 단위로 생성된 결과를 문장으로 구성하는 모듈을 추가하여 문장을 생성함으로써 ROUGE 요약 평가성능을 향상시켰다.

실험결과 포인팅과 성성 선택정보의 학습과 함께 BERT 형태소 임베딩과 OOV 랜덤 마스킹, 형태소 문장 생성기를 추가하여 기존 선행 연구의 성능과 비교했을 때 ROUGE-1이 47.01에서 54.97로 향상되고 ROUGE-L과 SU4 모두 좋은 성능이 나왔다.

ROUGE-1은 재현율 중심의 지표이고 ROUGE-L은 단어의 순서에 대한 지표로서 요약문 생성에서 생성성능과 포인팅성능이 향상됨을 보였다. 향후 연구로는 포인팅 단어의 품사 정보 적합성을 고려하여 구문 구조가 더 자연스럽게 생성되도록 하여 문장의 완성도를 높일 계획이다.

References

[1] Hongyeon Yu, Seungwoo Lee, and Youngjoong Ko, "Incremental Clustering and Multi-Document Sum-

3) KorQuAD 2.0 https://korquad.github.io

marization for Issue Analysis Based on Real-Time News," *Journal of KIISE*, Vol. 46, No. 4, pp. 355-362, Apr. 2019. (in Korean)
[2] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to Sequence Learning With Neural Networks," *arXiv:1409.3215v3*, 2014.
[3] Jaehyun Ryu, Yunseok Noh, Su Jeong Choi, and et. al., "Solving for Redundant Repetition Problem of Generating Summarization Using Decoding History," *Journal of KIISE*, Vol. 46, No. 6, pp. 535-543, Jun. 2019. (in Korean)
[4] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton, "Grammar as a Foreign Language," *arXiv:1412.7449v3*, 2015.
[5] Seok-won Jeong, Jintae Kim, and Harksoo Kim, "Document Summarization Using TextRank Based on Sentence Embedding," *Journal of KIISE*, Vol. 46, No. 3, pp. 285-289, Mar. 2019. (in Korean)
[6] Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, and Chandan K. Reddy, "Neural Abstractive Text Summarization With Sequence-to-Sequence Model," *arXiv:1812.02303v2*, 2018.
[7] Stanford NLP Group, "SQuAD," [Online]. Available: https://rajpurkar.github.io/SQuAD-explorer/ [Accessed: 19-Sep-2019]
[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805v2*, 2019.
[9] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li, "Incorporating Copying Mechanism in Sequence-to-Sequence Learning," *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1631-1640, Aug. 2016.
[10] C. Huang, H. Yen, P. Yang, S. Huang, and J. Chang, "Using Sublexical Translations to Handle the OOV Problem in Machine Translation," *ACM Transactions on Asian Language Information Proc. (TALIP)*, Vol. 10, No. 3, Article 16, 2011.
[11] S´ebastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio, "On Using Very Large Target Vocabulary for Neural Machine Translation," *Proc. The annual meeting of the Association for Computational Linguistics (ACL)*, 2015.
[12] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words With Subword Units," *arXiv:1508.07909*, 2015.
[13] M. Luong, I. Sutskever, Q. Le, O. Vinyals, and W. Zaremba, "Addressing the Rare Word Problem in Neural Machine Translation," *arXiv:1410.8206*, 2014.
[14] Kyoungho Choi and Changki Lee, "End-To-End Document Summarization Using Copy Mechanism and Input Feeding," *The 28th Annual Conference on Human & Cognitive Language Technology*, pp. 56-61, Oct. 2016. (in Korean)
[15] Tae-Seok Lee, Choong-Nyoung Seon, Youngim Jung, and Seung-Shik Kang, "Automatic Text Summarization Based on Selective Copy Mechanism Against for Addressing OOV," *Smart Media Journal*, Vol. 8, No. 2, Jun. 2019. (in Korean)
[16] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos Santos, Caglar Gulcehre, and Bing Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond," *arXiv:1602.06023v5*, 2016.
[17] Kavita Ganesan, "ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks," *arXiv:1803.01937v1*, 2018.
[18] Seungyoung Lim, Myungji Kim, and Jooyoul Lee, "KorQuAD: Korean QA Dataset for Machine Comprehension," *Proc. Korea Software Congress (KSC)*, 2018. (in Korean)

이 태 석
1995년 경원대학교 전자계산학과(학사) 2005년 고려대학교 컴퓨터학과(석사). 2016년 국민대학교 컴퓨터공학부(박사 수료) 1997년~현재 한국과학기술정보 연구원 융합서비스센터 책임연구원. 관심분야는 정보검색, 정보 추출, 기계학습

강 승 식
1986년 서울대학교 정보컴퓨터공학부(학사) 1988년 서울대학교 컴퓨터공학과(석사) 1993년 서울대학교 컴퓨터공학과(박사) 1994년~2001년 한성대학교 정보전산학부 부교수. 2001년~현재 국민대학교 소프트웨어학부 교수. 관심분야는 한국어정보처리, 자연어처리, 정보검색, 기계학습