

Sentence Recommendation Using Beam Search in a Military Intelligent Image Analysis System

Hyung-Sun Na[†] · Tae-Hyeon Jeon^{††} · Hyung-Seok Kang^{†††} · Jinhyun Ahn^{††††} · Dong-Hyuk Im^{†††††}

ABSTRACT

Existing image analysis systems in use in the military field are carried out by readers analyzing and identifying images themselves, writing and disseminating related content, and in this process, repetitive tasks are frequent, resulting in workload. In this paper, to solve the previous problem, we proposed an algorithm that can operate the Seq2Seq model on a word basis, which operates on a sentence basis, and applied the Attention technique to improve accuracy. In addition, by applying the Beam Search technique, we would like to recommend various current identification sentences based on the past identification contents of a specific area. It was confirmed through experiments that the Beam Search technique recommends sentences more effectively than the existing greedy Search technique, and confirmed that the accuracy of recommendation increases when the size of Beam is large.

Keywords : NLP, Beam Search, Military, Seq2seq, Attention Mechanism

군사용 지능형 영상 판독 시스템에서의 빔서치를 활용한 문장 추천

나 형 선[†] · 전 태 현^{††} · 강 형 석^{†††} · 안 진 현^{††††} · 임 동 혁^{†††††}

요 약

군사 분야에서 사용 중인 기존 영상 판독 시스템은 판독관들이 직접 영상을 분석 및 식별하여 관련 내용을 보고서에 작성하고 전파하는 방식으로 진행되는데 이 과정에서 반복 작업이 빈번하여 업무 과부하가 발생한다. 본 논문에서는 이러한 문제를 해결하고자, 기존의 문장 단위로 동작하는 Seq2Seq 모델을 단어 단위로 동작할 수 있는 알고리즘을 제안하고, Attention 기법을 적용해 정확도를 향상시키고자 한다. 또한 Beam 탐색 기법을 응용하여 특정 지역의 과거 식별내용을 바탕으로 현재 식별 문장을 다양하게 추천하고자 한다. 실험을 통해 Beam 탐색 기법이 기존 Greedy 탐색 기법보다 효과적으로 문장을 추천하는 것을 확인하였고, Beam의 크기가 클 때 추천의 정확도가 높아지는 것을 확인하였다.

키워드 : 자연어 처리, 빔 서치, 군사 분야, 시퀀스-투-시퀀스, 어텐션 기법

1. 서 론

군사 분야에서 사용하는 기존 판독 시스템은 판독관들이 영상을 분석하여 식별된 내용을 정보의 형태로 기록하고 전파하는 방식을 사용한다. 이러한 기존 방식은 단순히 영상에서 개체의 형태, 종류 등의 정적인 정보를 인식하는데 초점을

맞추고 있고, 분석 결과는 판독관이 검증하는 과정을 통하여 판독 보고서의 형태로 저장된다[1]. 군사적인 목적 중 징후 감시를 수행하기 위하여 표적들 간의 관계 및 앞으로 발생할 행위에 대한 예측정보가 중요한데, 이를 추론하는 것은 과거 정보에 대한 판독관들의 경험과 숙련도에 의존적이기 때문에 이는 반복 작업이 빈번하여 판독관들의 업무 과 부하를 일으킬 수 있다[2].

본 논문에서는 그동안 축적된 판독 보고서의 데이터를 기반으로 인공지능 언어 모델을 구축하여 영상 판독관들에게 과거 데이터를 제공함과 동시에 과거 데이터 기반 핵심 문장을 추천 해주는 시스템을 제안한다. 실제 판독보고서는 군사 기밀임으로 Fig. 1은 가 데이터로 작성된 판독보고서임을 밝힌다.

※ 이 논문은 국방과학연구소의 지원을 받아 수행된 연구임(UD190025FD).
† 준 회 원 : 광운대학교 인공지능융합학과 석사과정
†† 비 회 원 : 호서대학교 컴퓨터공학과 학사과정
††† 비 회 원 : 국방과학연구소 연구원
†††† 정 회 원 : 제주대학교 경영정보학과 조교수
††††† 종신회원 : 광운대학교 정부융합학부 부교수
Manuscript Received : October 7, 2021
First Revision : October 20, 2021
Accepted : October 23, 2021
* Corresponding Author : Dong-Hyuk Im(dhim@kw.ac.kr)

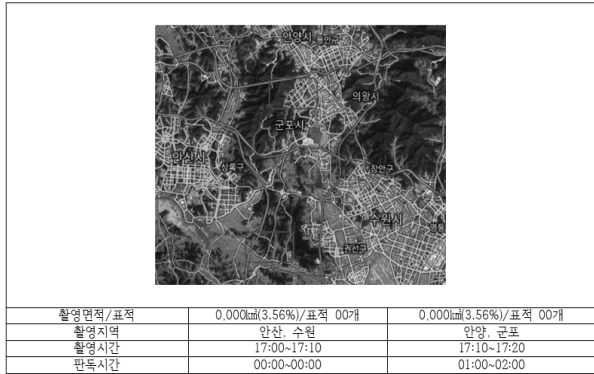


Fig. 1. Image Analysis Report Sample

문장 생성을 위한 기존 언어 모델인 Seq2Seq 모델은 인코더에 문장이 입력된 후 디코더에서 문장을 출력하는 방식을 사용하기 때문에 인코더의 입력 값으로 단어를 하나씩 입력할 수 없다[3-6]. 본 논문에서는 Attention 기법[7]을 적용한 디코더에서 인코더에 입력된 단어들의 다음 단어를 예측하도록 모델을 학습하고, 출력 시에는 기존 입력받은 단어들과 디코더에서 예측한 단어를 다시 인코더에 넣어 다음 단어를 예측하는 방식을 사용하여 위의 문제를 해결하였다.

또한 출력 시에 기존 Greedy 탐색 기법을 확장한 Beam 탐색 기법[8]을 사용하여 기존 Greedy 탐색 기법보다 높은 유사도를 가지는 문장들을 추천해주는 것을 실험을 통해 확인하였다. 또한 사용자에게 더 다양한 문장을 추천해 줌으로써 기존 문장 추천 시스템의 한계를 벗어나 판독 보고서를 작성하는 판독관들의 업무 효율을 높여 업무 과 부하를 줄일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 시스템의 가장 기초가 되는 데이터 전처리 과정을 서술하고, 3장에서는 제안하는 모델의 전체적인 개요와 적용된 기법, 구축 방법 등을 서술한다. 4장에서는 디코더에서 문장 생성을 할 때 동작 방법을 서술하고, 5장에서는 실험방법과 평가방법을 설명한 후 실험 결과를 보여준다. 마지막 6장에서는 결론과 향후계획을 서술하며 마무리한다.

2. 데이터 전처리

[1]은 본 논문의 이전 단계이며 [1]의 전처리를 기반으로 본 논문의 작업이 진행된다. [1]에서의 전처리 방식을 요약하면 다음과 같으며, Fig. 2는 데이터 전처리 과정의 도식화이며 Fig. 1. 과 마찬가지로 실제 데이터는 공개가 불가능한 보호 데이터임으로 도식화에 사용한 데이터는 기존의 판독 보고서 형식과 유사하게 제작한 샘플 데이터임을 밝힌다.

2.1 데이터 형식

본 논문에서 제안하는 시스템에서 사용한 데이터는 영상 판독보고서이며, 특징으로는 주제와 작성자에 따라 보고서의 형식 및 구조가 다르고, 오타자 등의 휴먼 오류가 존재하며, 정보 전달을 위한 간결한 문어체로 작성되어 있다. 보고서가 일정한 형식으로 통일되어 있지 않아 본 연구에서 제안하는 시스템에 적용하기 위해 보고서의 내용을 구조화된 형태로 통일할 필요가 있다.

2.2 데이터 구조화

영상 판독 보고서의 경우 한글파일(.hwp)로 작성되어 있어 데이터 구조화를 위해 txt파일로 변환 할 필요가 있고, 본 논문에서는 pyhwp 오픈소스 라이브러리를 사용하였다. pyhwp란 HWP 문서 버전 5 파일의 내부 스트림을 분석/분리 하는 기능과 .hwp파일을 OpenDocument(.odt)나 일반 텍스트 문서로 변환하는 기능을 제공하는 오픈소스 라이브러리이다[9]. hwp 파일을 txt로 변환하는 과정에서 깨진 파일, 암호화된 파일 등 파일 자체에 문제가 있는 것들은 모두 제거하였다.

2.3 데이터 클리닝

영상 판독 보고서를 작성하는 과정에서 판독관마다 문서 작성 방식이 조금씩 다르고, 오타 혹은 줄 밀림 현상 등이 존재한다. 문서 작성 방식의 경우는 큰 틀은 같았기에 정규 표현식을 활용하여 파싱하였고, 오타는 뒤에 서술할 FastText [10] 기법의 특징인 SubWord 학습 방법으로 완화하였다. 줄

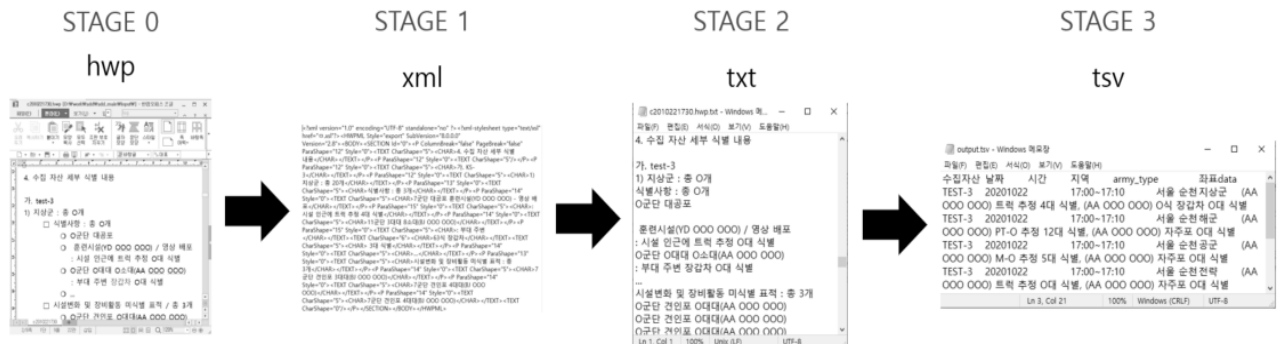


Fig. 2. Data Preprocessing Process

밀립 현상은 보고서의 챗터들의 키워드를 기준으로 매칭 함으로써 해결하였다.

2.4 형태소 분석

한국어는 교착어로 구문적 복잡성을 내재하고 있고, 문법이 복잡하기 때문에 형태소 분석은 필수적으로 필요한 작업이다. 형태소 분석기는 꼬꼬마, KoNLP, MeCab-ko, KOMRAN 등 다양한 오픈소스 라이브러리가 존재한다[11]. 본 논문에서는 다양한 형태소 분석기들 중 연산 속도가 가장 빠르고 분석 품질 또한 우수한 MeCab 형태소 분석기를 채택하여 사용했다[12].

3. 모델 구축

Table 1은 본 논문에서 사용하는 용어 데이터에 대해 정의한다. X 는 인코더에 들어갈 영상 판독 문장을 의미하며 x 은 각 문장의 요소 단어를 의미한다. Y 는 디코더 학습에서 필요한 문장을 의미하며 $\langle \text{sos} \rangle$ 는 시작 토큰, $\langle \text{eos} \rangle$ 는 종료 토큰을 의미한다. V 는 학습에 사용한 데이터들의 단어 집합을 의미하며 W 는 문장을 의미한다. T 는 각 문장의 길이를 의미하고, t 는 각 단어가 입력되는 시점을 의미한다. 마지막으로 c_t 는 인코더의 마지막 hidden State인 Context Vector를 의미한다.

3.1 언어 모델

언어 모델은 문장의 각 단어에 확률을 할당하는 모델이다. 이는 NLP(Natural Language Processing) 분야에서 핵심 기술 중 하나이다. w 는 하나의 단어, 대문자 W 는 단어들의 모음인 문장이라고 가정하며, n 개의 단어가 등장하는 문장의 확률 식은 다음 Equation (1)과 같다.

$$P(W) = P(w_1, w_2, \dots, w_n) \quad (1)$$

언어 모델은 문장의 각 단어에 확률을 할당하기 위해 이전 단어들이 주어졌을 때, 다음 단어를 예측한다. 전체 문장의 확률은 조건부 확률을 이용해 Equation (2)와 같이 표현할 수 있다.

Table 1. Notations

Variable	Definition
X	$\{x_1, x_2, \dots, x_m\}$ (source sentence)
Y	$\{\langle \text{sos} \rangle, y_1, y_2, \dots, y_n, \langle \text{eos} \rangle\}$ (target sentence)
V	Vocab
W	Word()
T	Sentence \leq ngth
t	Time Step
c_t	Context Vector

$$P(W) = \prod_{i=1}^n P(w_n | w_1, w_2, \dots, w_{n-1}) \quad (2)$$

Conditional 언어 모델은 기존 문장 확률 식에 Sequence-to-Sequence(Seq2Seq) 모델에서 사용하는 인코더의 마지막 Vector인 Conditional Context Vector(c_t)를 추가하여 일반화한 모델이며, 수식은 다음 Equation (3)과 같다.

$$P(W) = P(w_n | c_t, w_1, w_2, \dots, w_{n-1}) \quad (3)$$

3.2 단어 임베딩

3.1의 언어 모델을 구현하기 위해서는 문장의 각 단어들을 숫자로 변환할 필요가 있다. 여러 변환 방법들 중 단어 임베딩 기법은 단어를 컴퓨터가 이해하기 쉬운 벡터로 표현하고, 서로 유사한 단어의 벡터는 임베딩 공간에 가까운 위치에 배치 되도록 고안된 기법이다. 이는 모델이 단어의 의미 정보를 이해하고 문장을 생성하는데 장점을 가진다. 이를 제안한 기법으로는 Word2Vec, Glove, FastText 등이 있다[10,13,14]. 본 논문에서는 여러 임베딩 기법들 중 학습 속도가 빠르고, 판독보고서의 휴먼 오류를 완화하기 위해 Subword로 단어들을 임베딩 하여 오타문제와 학습한 단어가 아닌 단어가 입력되었을 때 발생하는 OOV(Out-Of-Vocabulary)문제에 강한 FastText모델을 채택하였다[10].

3.3 Recurrent Neural Network

단어 임베딩을 통해 벡터의 형태로 임베딩 된 데이터들을 신경 언어 모델(Neural Language Model)을 사용하여 학습한다[15]. 여러 신경 언어 모델들 중 Audio, Text 등의 Sequential 데이터를 처리하는 모델은 일반적으로 RNN(Recurrent Neural Network)을 사용하여 구축한다[16]. RNN은 가변적인 길이 시퀀스를 입력으로 사용할 수 있고, 데이터에서 장기적인 종속성(long-term dependency)을 학습할 수 있다는 장점이 있다. 그러나 기울기 소실 혹은 폭주문제가 발생하는 경우가 존재여, 문장의 길이가 길어질수록 문장의 앞부분을 충분히 활용하지 못하는 문제가 있다. 판독 보고서의 경우 길이가 짧은 문장부터 긴 문장까지 다양함으로 RNN을 사용한 모델은 적절하지 않다. 이를 개선한 모델로 LSTM(Long Short-Term Memory)과 GRU(Gated Recurrent Units)모델이 제안되었다[17-19]. 문장 생성 문제에서 LSTM과 GRU 모델을 비교하였을 때, LSTM이 성능이 경미하게 좋았지만, GRU 모델 또한 크게 차이 나지 않는다. 다만 학습 속도 측면에서는 GRU 모델의 구조가 더 간단하여 더 빠르기 때문에 본 논문에서는 LSTM과 성능 차이가 크지 않고 속도가 빠른 GRU 모델을 채택하였다[20].

3.4 Sequence-to-Sequence 학습

Seq2Seq은 인코더-디코더 모델로써, 각 모델에 RNN 모

렐을 사용하여 입력 시퀀스를 받은 후, 출력 시퀀스를 생성하는 모델이다. 이는 가변적인 입/출력 시퀀스 길이를 처리할 수 있는 장점이 있어 NMT, ChatBot, Dialog System 등 다양한 분야에서 사용되고 있다[4-6].

Seq2Seq에서 인코더는 입력 문장으로부터 Context Vector를 생성한다. 디코더는 출력 시퀀스를 생성하기 위해서 입력 시퀀스에서 압축된 정보인 Context Vector를 디코더의 첫 번째 hidden state로 활용한다. 각 타임 스텝 단어는 다음 Equation (4)와 같은 조건부 확률로 정의된다.

$$P(Y) = \prod_t P(y_t | c_t, y_{<t}) \quad (4)$$

이를 log-최대 우도법(Maximum Likelihood Estimation)을 위한 조건부 확률로 변환하면 다음 Equation (5)와 같이 정의된다.

$$\log P(Y) = \sum_t \log P(y_t | c_t, y_{<t}) \quad (5)$$

3.5 Attention 기법

Attention 기법은 디코더의 각 타임 스텝마다 인코더를 다시 참조하는데 이는 각 타임 스텝마다 인코더의 전체 부분을 동일하게 보는 것이 아니라, 현재 예측해야 할 디코더의 부분과 연관 있는 인코더 부분을 집중적으로 참조한다. 이를 통해 문장을 생성하면서 다음 단어를 예측할 때, 이전 단어 중 가장 관련 있는 단어와 있는 단어를 선택하게 되어 문장의 길이가 길어질 경우 성능이 떨어지는 Seq2Seq모델의 단점을 완화시키며 성능을 향상할 수 있다[7].

본 연구에서 사용하는 모델을 Fig. 3의 형태로 도식화하였다. 기존 Seq2Seq 모델과 다르게 디코더 모델에서 문장이 아닌 단어만 출력 후 이를 반복하는데, 디코더 모델에서 Attention Mechanism을 사용함으로써 Seq2Seq 모델의 단점을 완화함과 동시에 장점을 유지하였다.

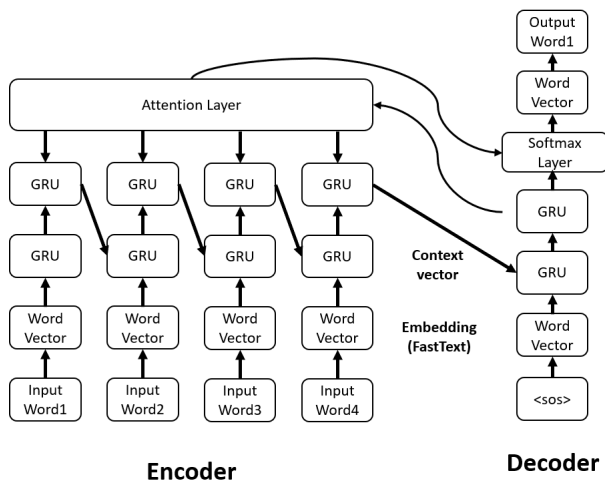


Fig. 3. Seq2Seq+Attention Model Graphical Sample

3.6 단어 기반 모델 훈련 방법

인코더에 입력된 단어들의 다음 단어를 디코더에서 출력하도록 학습이 진행된다. 먼저 전체 데이터의 배치 사이즈의 개수 만큼 반복하며, 입력된 단어들로 인코더의 Context Vector를 구한 후 디코더에서 각 단어의 다음 단어를 예측하도록 학습을 하는데, 디코더에서 예측한 단어를 기존 입력받은 단어들과 합쳐 다시 인코더의 Context Vector를 구한 후 다음 단어를 학습하는 과정을 반복한다. 디코더가 문장의 끝을 나타내는 토큰인 < eos >를 예측하도록 학습하게 되면 하나의 배치가 끝나게 된다. 각 예측마다 손실함수를 최소화시키는 방향으로 경사하강법과 역 전파 알고리즘을 사용하며 학습을 진행한다.

4. 단어 기반 문장 생성을 위한 디코딩 방법

4.1 Greedy 탐색 기법

일반적으로, 문장 생성은 단어들의 시퀀스를 통해 아웃풋으로 단어를 예측하는 방식이다. 이를 위해 Seq2Seq 모델의 디코더는 각각의 타임 스텝에서 전체 단어 사전에 대한 확률 분포를 예측한다. 따라서 단어를 생성하기 위해 모델의 예측 확률 분포를 이용해 단어로 변환하는 과정이 필요하다. 문장 생성에 있어 최적의 값을 찾는 것은 어렵기 때문에 우리는 문장 생성을 위해 휴리스틱 탐색 기법을 사용해야 한다. 일반적으로 사용되는 Greedy 탐색 기법은 각 단어를 예측할 때, 단어들의 확률 분포에서 가장 확률이 높은 단어를 선택하기 때문에 탐색 속도가 빠른 알고리즘이다. 다만, 최적의 해를 보장하지 않아 앞의 시퀀스가 잘못되었을 경우 원하는 값과 멀어진다는 단점이 있다. Greedy 탐색 기법의 수식은 다음 Equation (6)과 같으며, Equation (6)에서 \hat{y} 은 예측 값을 의미한다.

$$\hat{y} = \operatorname{argmax} \log P(y_t | c_t, y_{t-1}) \quad (6)$$

4.2 Beam 탐색 기법

Greedy 탐색 기법의 특징인 국소적으로는 최적의 단어를 선택하지만 전역적으로는 아닐 수 있는 단점을 개선한 알고리즘으로 Beam 탐색 기법이 있다. Beam 탐색 기법은 매 순간 가장 높은 확률의 단어를 찾는 Greedy 탐색 기법과 달리, 가장 높은 확률의 문장 혹은 단어를 예측한다. Beam 탐색 기법은 Greedy 탐색기법에서 타임 스텝마다 하이퍼 파라미터인 K개의 문장들로 확장하고, 예측한 각 단어들의 확률의 점수를 계산한 후, 가장 확률이 높은 K개의 문장을 남기고 나머지를 제거한다. 매 타임 스텝마다 K개의 문장을 유지하여 최종적으로 K개의 문장이 남게 되고, 이것이 우리가 예측한 K개의 문장들이 된다. K의 개수는 5~10이 적절하다고 한다[21]. Beam 탐색 기법의 계산식은 각 단어의 예측 확률 f_θ 을 모두 합하여 문장의 확률을 구하는 방식이며 다음 Equation (7)과 같다.

$$\hat{Y} = f_{\theta}(c_t, y_{<t}^1) \cup f_{\theta}(c_t, y_{<t}^2) \cup \dots \cup f_{\theta}(c_t, y_{<t}^n) \quad (7)$$

Beam 탐색을 진행하면서 점수는 log 값을 더하여 계산되기 때문에 문장의 길이가 길어질수록 확률 점수가 낮아진다는 단점이 있다. 이는 긴 문장을 제대로 출력하지 못한다는 문제로 이어진다. 이를 방지하기 위해 출력 문장의 길이에 상관없도록 점수를 타임 스텝의 크기만큼 나누어 정규화 함으로써 단점을 개선하였다. 최종적으로, 점수 계산법은 다음과 같으며 소문자 t는 타임 스텝을 의미한다.

$$score(y_1, \dots, y_t) = \log(y_1, \dots, y_t | c_t) = \frac{1}{t} \sum_{i=1}^t \operatorname{argmax} \log P(\hat{Y}) \quad (8)$$

4.3 단어 기반 모델 Beam 탐색 기법

기존 Beam 탐색 기법은 인코더에 문장이 입력된 후 디코더의 출력 층에서 진행되는 방식이며, 본 논문에서 제안하는 시스템은 문장이 아닌 단어 하나가 입력되었을 때도 문장을 추천해주는 시스템으로 구성되었기 때문에 단어 단위로 탐색하는 Beam 탐색 기법을 제안한다. 제안한 알고리즘의 의사 코드는 Fig. 4와 같다.

의사 코드를 줄 단위로 설명하자면 다음과 같다. 사용자가 단어 혹은 문장을 입력한다(2줄). Beam_k 개의 문장이 모두 <eos>로 끝날 때 까지 밀의 과정 반복한다.(3줄) <eos>로 끝

```

1 Procedure
2 sentence = [input()] (User Input Word or sentence)
3 while sentence[:Beam_k][-1] == <eos> (All sentence)
4     temp_sentence=[]
5     for i in Beam_k
6         if sentence[i]-1 == <eos>
7             temp_sentence += sentence[i]
8             continue
9         context_v = encoder(sentence[i])
10        result = decoder(context_v) (1 word * beam_k,
Score ascending)
11        for j in Beam_k
12            temp_sentence += sentence[i]+result[j]
13        sentence = temp_sentence[:Beam_k] (Score
ascending)
    
```

Fig. 4. Decoding Pseudo Code

난 문장은 모든 문장이 <eos>로 끝날 때 까지 대기한다(5 줄~8줄). 입력 받은 문장을 인코더에 입력 후 Context Vector를 받는다.(9줄) 디코더에 인코더에서 받은 Context Vector를 넣어 높은 확률 순으로 정렬된 단어리스트를 받는다.(10줄) 기존 문장과 Beam_k의 개수만큼 단어들을 합친다(11줄~12줄). 생성된 문장들 중 높은 확률 순으로 Beam_k 개수만큼 저장 후 위 과정을 반복한다(13줄).

5. 실험 및 결과

5.1 실험 환경 및 하이퍼 파라미터

본 논문에서 제안한 시스템을 실험한 컴퓨터 사양은 CPU: Intel Xeon Gold 6134 3.2GHz, GPU: NVIDIA Quadro GP100, RAM: 32GB이다. 모델구축에 사용한 라이브러리는 Pytorch-cuda, FastText를 사용하였으며 구현 언어는 Python이다.

데이터 셋은 총 데이터 셋에서 학습용 70%, 검증용 15%, 평가용 15%로 분리하여 사용하였다. 학습 시 설정한 하이퍼 파라미터는 다음과 같다. 훈련 데이터의 batch_size는 32, epoch은 28, 인코더의 layer는 3, 디코더의 layer은 2, hidden_size는 512로 설정하였으며, 실험에 사용한 컴퓨터의 사양과 학습 데이터의 크기를 고려하여 설정하였다. epoch는 검증 데이터로 모델 평가를 진행 하였을 때 가장 높은 유사도를 보이는 횟수로 설정하였다.

5.2 실험 방식 및 평가지표

실험에 사용한 평가용 데이터 셋의 문장의 길이는 짧게는 3 단어로 이루어진 문장부터 최대 50 단어로 이루어진 문장으로 이루어져 있다. 실험에 사용한 판독고보고서의 경우 외부 유출이 금지된 보호 데이터이기 때문에 Fig. 5는 가공된 샘플 데이터로 진행한 결과를 보여준다. Fig. 5의 Beam 탐색기법의 하이퍼 파라미터(Beam_k)는 5이며, army_type (지상군, 해군, 공군, 전력), location(좌표)은 입력할 문장의 가장 앞에 더해져 모델에 입력된다.

모델에 대한 평가지표는 평가용 데이터 셋에서 목표 문장을 하나씩 꺼내온 후, 각 문장에서 한 단어씩 증가 시키며 모델에 입력 후 생성한 문장과 목표 문장의 유사도를 측정하였다. 문장의 유사도 평가는 2-gram 방식으로 진행하였고, 문장의 유사도가 높을수록 정확한 문장을 추천한다.

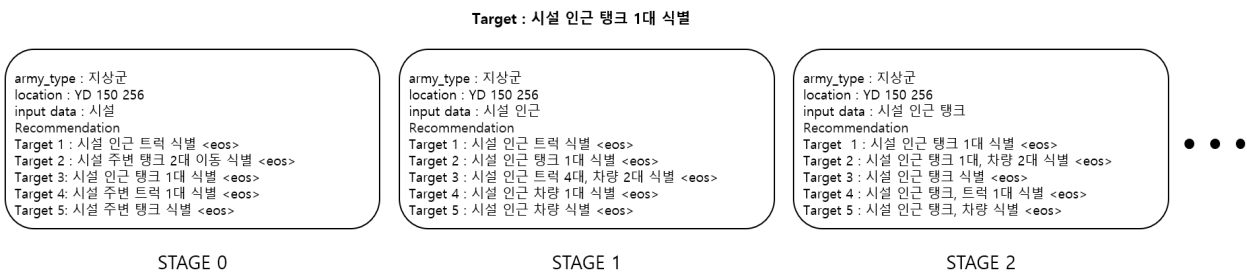


Fig. 5. An Example of Result

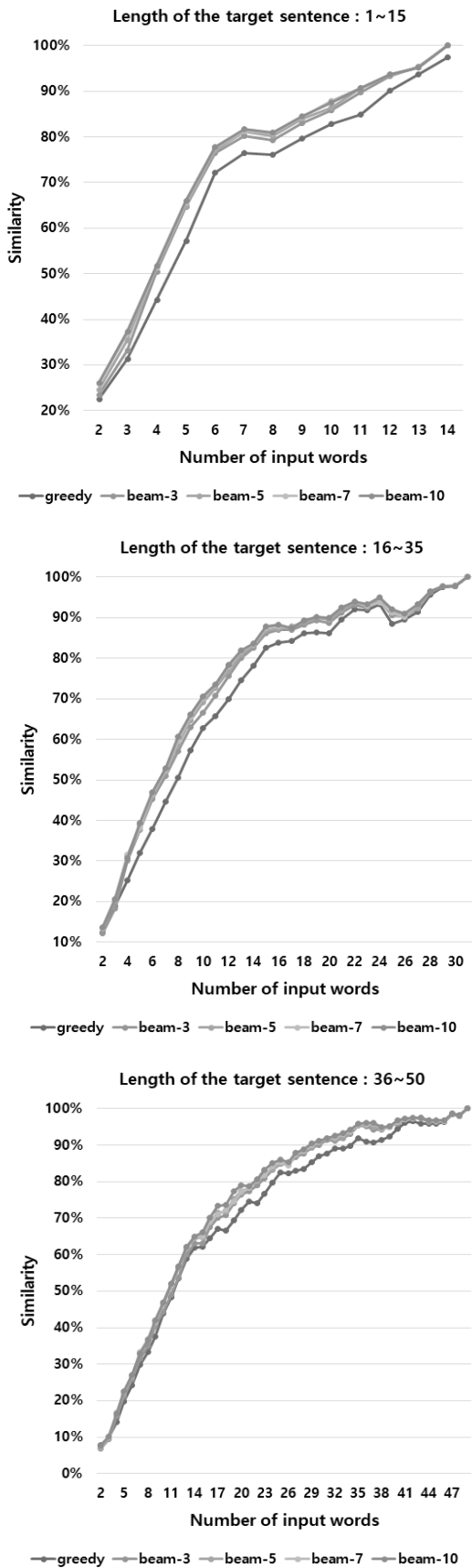


Fig. 6. Similarity by length of sentences

5.3 결과 및 토론

자연어 생성 문제는 문장의 길이에 영향을 많이 받기 때문에, 문장의 길이가 1~15단어, 16~35 단어, 36~50 단어를 가지는 문장들을 그룹화 하여 각 그룹 당 입력된 단어의 길이에 따른 유사도를 Fig. 6 그래프로 시각화 하였다. Fig. 6 그래프를 보면 문장의 총 길이와는 관계없이 입력되는 단어의 수가 증가할수록 유사도가 증가하고, 평가용 문장의 길이가 짧을수록 각 기법들의 유사도 편차가 큰 것을 확인 할 수 있다. 입력된 단어가 적을 때 유사도가 작은 이유는 첫 번째 단어로 유추할 수 있는 문장들이 무수히 많기 때문이다.

디코딩 기법들 종류들을 살펴보면 Beam 탐색기법의 하이퍼 파라미터의 숫자는 4.2절 에서 언급한 것과 같이 5 이상부터는 크게 차이가 나지 않는 것을 확인할 수 있다. 또한 기존 Greedy 탐색 기법 보다 Beam 탐색 기법이 유사도 점수가 높은 것을 확인할 수 있다.

하지만 Beam 탐색 기법의 하이퍼 파라미터의 크기가 커질수록 시간 복잡도와 공간 복잡도가 기하급수적으로 커지므로 적절한 크기의 하이퍼 파라미터를 설정할 필요가 있다.

6. 결론 및 향후 계획

본 논문은 영상 판독보고서를 기반으로 단어를 입력할 시 문장을 추천해주는 지능형 시스템을 제안하였다. 제안한 시스템은 과거 데이터를 토대로 특정 지역의 패턴을 고려한 문장을 추천하고 문장 추천 시스템에 Beam 탐색 기법을 적용함으로써 보고서에 입력될 확률이 높은 여러 문장들을 함께 추천한다. 또한 기존의 문장 단위가 아닌 단어 단위로 문장을 추천해주며, 이는 군사용 영상 판독관들의 업무 효율성 증가와 업무 과부하를 줄일 수 있을 것으로 판단된다. 또한 다년간 축적된 판독보고서를 활용한다는 점에 있어서 의미가 있다.

향후 계획으로는 추가 데이터 확보와 모델의 성능을 더 높일 수 있는 기법들을 추가로 적용하고 영상 판독관들이 사용할 수 있도록 UI를 적용한 프로그램을 제작하고자 한다. 프로그램에는 문장 추천 기능과 입력하고자 하는 지역의 과거 정보를 제공하는 기능을 탑재할 예정이다.

References

- [1] T. H. Jeon, H. S. Na, J. H. Ahn, and D. H. Im, "Pre-processing and implementation for intelligent imagery interpretation system," *Proceedings of the Korea Information Processing Society Conference*, Vol.28, pp.305-307, 2021.
- [2] Y. D. Kim and H. J. Gwon, "A study on defense command and control system AI application," *Korea Information Processing Society Review*, Vol.24, No.1, pp.13-18, 2017.
- [3] I. Sutskever, O. Vinyals, and V. L. Quoc, "Sequence to sequence learning with neural networks," In: *Advances in neural Information Processing Systems*, pp.3104-3112, 2014.

- [4] M. Zhang, Z. Li, G. Fu, and M. Zhang, "Syntax-enhanced neural machine translation with syntax-aware word representations," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol.1 (Long and Short Papers), 2019.
- [5] K. Palasundram, N. M. Sharef, N. Nasharuddin, K. Kasmiran, and A. Azman "Sequence to sequence model performance for education chatbot," *International Journal of Emerging Technologies in Learning (ijET)*, Vol.14, No.24, pp.56-68, 2019.
- [6] K. Qian and Z. Yu. "Domain adaptive dialog generation via meta learning," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [7] D. Bahdanau, C. Kyunghyun, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *3rd International Conference on Learning Representations, ICLR 2015*.
- [8] S. Wiseman and A. M. Rush. "Sequence -to-sequence learning as beam-search optimization," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.
- [9] pyhwp Documentation 2013. [Internet], <https://pythonhosted.org/pyhwp/ko/> (accessed August 2, 2021.)
- [10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, Vol.5, pp.135-146, 2017.
- [11] E. J. Park and S. Z. Cho, "KoNLPy: Korean natural language processing in Python," *Annual Conference on Human and Language Technology*, pp.133-136, 2014.
- [12] Comparison of Korean stemming analyzer performance (2018). [Internet], <https://iostream.tistory.com/144> (accessed August 2, 2021)
- [13] T. Mikolov, K. Chen, G Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [14] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp.1532-1543, 2014.
- [15] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, Vol.3, 1137-1155, 2003.
- [16] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," *In Eleventh Annual Conference of the International Speech Communication Association*, Vol.9, pp.1045-1048, 2010.
- [17] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult." *IEEE Transactions on Neural Networks*, Vol.5, No.2, pp.157-166, 1994.
- [18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol.9, No.8, pp.1735-1780, 1997.
- [19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling." *NIPS 2014 Workshop on Deep Learning*, Dec. 2014.
- [20] S. Mangal, P. Joshi, and R. Modak, "Lstm vs. gru vs. bidirectional rnn for script generation," *arXiv preprint arXiv:1908.04332*, 2019.
- [21] K. Cho, "Noisy parallel approximate decoding for conditional recurrent language model," *arXiv preprint arXiv:1605.03835*, 2016.

나 형 선



<https://orcid.org/0000-0002-3941-3959>
 e-mail : nayosk@kw.ac.kr
 2021년 호서대학교 컴퓨터공학과(학사)
 2021년 ~ 현 재 광운대학교
 인공지능융합학과 석사과정
 관심분야 : NLP, 추천 시스템

전 태 현

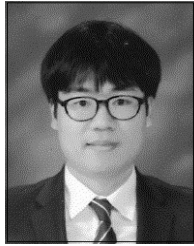


<https://orcid.org/0000-0003-1437-5682>
 e-mail : 20161557@vision.hoseo.edu
 2016년 ~ 현 재 호서대학교 컴퓨터공학과
 학사과정
 관심분야 : 데이터마이닝, 이상탐지, 추천
 시스템

강 형 석



<https://orcid.org/0000-0001-6389-3326>
 e-mail : kanghs@add.re.kr
 2008년 ~ 2012년 포항공과대학교(학사)
 2012년 ~ 2015년 포항공과대학교(석사)
 2015년 ~ 현 재 국방과학연구소 연구원
 관심분야 : 영상처리, 이상탐지



안 진 현

<https://orcid.org/0000-0002-2331-004X>

e-mail : jha@jejunu.ac.kr

2005년 고려대학교 컴퓨터교육과
(이학학사)

2007년 고려대학교 컴퓨터교육과
(이학석사)

2017년 서울대학교 의료경영과정정보학(컴퓨터공학박사)

2017년 ~ 2018년 서울대학교 의생명지식공학연구실 선임연구원

2018년 ~ 현 재 제주대학교 경영정보학과 조교수

관심분야: 분산/병렬 컴퓨팅, 지식공학, 온톨로지, 비식별화



임 동 혁

<https://orcid.org/0000-0002-0290-755X>

e-mail : dhim@kw.ac.kr

2003년 고려대학교 컴퓨터교육과
(이학학사)

2005년 서울대학교 컴퓨터공학과
(공학석사)

2011년 서울대학교 컴퓨터공학과(공학박사)

2011년 ~ 2012년 서울대학교 의생명지식공학연구실 선임연구원

2013년 ~ 2020년 호서대학교 컴퓨터공학과 조교수/부교수

2020년 ~ 현 재 광운대학교 정보융합학부 부교수

관심분야: 빅데이터, 머신러닝, 시공간 데이터 분석