JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 25, No. 11: 1505~1511, Nov. 2021

# 웹 말뭉치에 대한 문장 필터링 데이터 셋 구축 방법

남충현[1] · 장경식[2*]

# Sentence Filtering Dataset Construction Method about Web Corpus

**Chung-Hyeon Nam[1] · Kyung-Sik Jang[2*]**

[1]Graduate Student, Department of Computer Engineering, Korea University of Technology and Education, Cheonan 31253 Korea
[2*]Professor, Department of Computer Engineering, Korea University of Technology and Education, Cheonan 31253 Korea

## 요 약

자연어 처리 분야 내 다양한 작업들에서 높은 성능을 보인 사전 학습된 모델은 대량의 말뭉치를 이용하여 문장들의 언어학적 패턴을 스스로 학습함으로써 입력 문장 내 각 토큰들을 적절한 특징 벡터로 표현할 수 있다는 장점을 갖고 있다. 이러한 사전 학습된 모델의 학습에 필요한 말뭉치를 구축하는 방법 중 웹 크롤러를 이용하여 수집한 경우 웹 사이트에 존재하는 문장은 다양한 패턴을 갖고 있기 때문에 문장의 일부 또는 전체에 불필요한 단어가 포함되어 있을 수 있다. 본 논문에서는 웹으로부터 수집한 말뭉치에 대해 신경망 모델을 이용하여 불필요한 단어가 포함된 문장을 필터링하기 위한 데이터 셋 구축 방법에 대해 제안한다. 그 결과, 총 2,330개의 문장을 포함한 데이터 셋을 구축하였다. 또한 신경망 모델을 이용하여 구축한 데이터 셋을 학습시켜 성능을 평가하였으며, BERT 모델이 평가 데이터에 대해 93.75%의 정확도로 가장 높은 성능을 보였다.

## ABSTRACT

Pretrained models with high performance in various tasks within natural language processing have the advantage of learning the linguistic patterns of sentences using large corpus during the training, allowing each token in the input sentence to be represented with appropriate feature vectors. One of the methods of constructing a corpus required for a pre-trained model training is a collection method using web crawler. However, sentences that exist on web may contain unnecessary words in some or all of the sentences because they have various patterns. In this paper, we propose a dataset construction method for filtering sentences containing unnecessary words using neural network models for corpus collected from the web. As a result, we construct a dataset containing a total of 2,330 sentences. We also evaluated the performance of neural network models on the constructed dataset, and the BERT model showed the highest performance with an accuracy of 93.75%.

**키워드** : 자연어 처리, 딥러닝, 문장 필터링, 말뭉치 구축

Keywords : Natural language processing, Deep learning, Sentence filtering, Corpus construction

**Received** 16 August 2021, **Revised** 18 August 2021, **Accepted** 4 September 2021

*** Corresponding Author** Kyung-Sik Jang(E-mail:ksjang@koreatech.ac.kr, Tel:+82-41-560-1352)
Professor, Department of Computer Engineering, Korea University of Technology and Education, Cheonan, 31253 Korea

Open Access http://doi.org/10.6109/jkiice.2021.25.11.1505 print ISSN: 2234-4772 online ISSN: 2288-4165

## Ⅰ. 서 론

최근 자연어 처리 분야에서는 대량의 말뭉치로 사전 학습된 신경망 모델을 수행하고자 하는 세부 분야에 맞게 추가 학습하는 방법으로 사용하고 있는 추세이다[1]. 이러한 방법은 말뭉치 내 문장들의 언어학적 패턴들을 사전에 학습한 신경망 모델이 문장 내 각 토큰들을 유의미한 특징 벡터로 표현할 수 있기 때문에 각 작업에 맞게 신경망 모델을 처음부터 학습하는 방법보다 더 높은 성능을 보여주고 있다[1,2].

다양한 입력 문장에 대해 적절한 표현을 할 수 있는 사전 학습 모델을 만들기 위해서는 학습 과정에서 정제된 대량의 말뭉치가 필수적으로 요구되어지며, 대표적인 사전 학습 모델 중 하나인 BERT(Bidirectional Encoder Representations from Transformers) 모델의 경우 약 16GB 정도의 위키피디아, 책 등을 포함한 말뭉치를 이용하여 학습하였다.

사전 학습에 필요한 대량의 말뭉치는 위키피디아와 같은 인터넷 백과사전에서 공개한 말뭉치를 이용하는 방법[3], 웹 크롤러를 이용하여 웹상의 문장들을 직접 수집하는 방법 등을 이용하여 구축할 수 있다. 공개된 말뭉치를 이용하는 방법은 대부분 데이터에 대한 필터링 작업이 사전에 수행되어 있기 때문에 추가적인 전처리 과정을 최소화할 수 있다는 장점을 갖고 있지만, 말뭉치가 사전 학습에서 사용되기에는 소량으로 이루어져 있거나, 특정 주제에 대해 편향된 문장만을 포함하고 있을 수 있다.

다른 방법으로 웹 크롤러를 이용하여 문장을 수집하는 방법은 웹 크롤러가 수많은 웹 사이트를 방문하여 HTML 파일들을 수집하고, 수집된 파일 내 태그나 자바스크립트 코드와 같은 불필요한 요소 제거 등의 정제 작업과 파일에 속한 문장 추출 작업을 수행하면서 말뭉치를 구축하는 방법이다.

이러한 일련의 과정을 수행하는 웹 크롤러 방법은 다수의 불특정 웹 사이트에 방문하기 때문에 공개된 데이터를 이용하는 방법보다 다양한 주제에 대한 문장을 수집할 수 있으며, 시간이 지남에 따라 대량의 말뭉치를 지속적으로 구축할 수 있다는 장점을 갖고 있다. 하지만 HTML 파일 내 태그 쌍이 정확하지 않아 태그의 경계가 모호해져 파싱이 제대로 되지 않거나, 태그 내 다른 태그가 존재하는 경우 문장 추출 시 문장에 불필요한 단어가 포함되는 등 HTML 문법의 유연성으로 인해 문장의 일부에 불필요한 노이즈가 포함되거나 노이즈를 문장으로 인식할 수도 있다.

이러한 이유로 수집된 말뭉치는 노이즈가 포함된 문장 필터링 작업이 추가적으로 수행해야 하며, 이 작업은 구두점, 품사 확인 등 특정 패턴을 사전에 정의한 후 문장에 대해 패턴과 일치하는 문장과 아닌 문장들로 분류하는 작업을 포함한다. 하지만 사전에 정의된 특정 패턴을 이용한 필터링 과정은 문장을 이루는 HTML 내 다양한 패턴들에 대해 모두 대응하기엔 현실적으로 불가능하기 때문에 모든 노이즈를 제거하기 어렵다.

본 논문에서는 특정 패턴을 사전에 정의하지 않고 신경망 모델이 문장 필터링 처리 과정을 수행할 수 있도록 학습에 필요한 문장 필터링 데이터 셋을 구축하는 방법에 대해 제안한다.

자세히는 웹으로부터 수집된 문장들에 대해 문장 필터링을 적용하여 정확한 문장만을 추출해내는 것을 목표로 하며, 완전한 문장과 불완전한 문장에 대한 정의와 신경망 모델의 문장 필터링 분류 방법 학습을 위한 데이터 셋 구축 방법을 제시하고, 최종 구축된 데이터 셋에 대해 소개하고자 한다. 또한 문장 분류를 위해 주로 사용되는 대표적인 신경망 모델들을 이용하여 구축된 데이터 셋을 이용하여 학습하여 각 모델에 대한 성능 평가를 하고자 한다.

그 결과, 자체 개발한 문장 태깅 툴을 이용하여 완전한 문장 474개와 불완전한 문장 1,856개가 포함된 총 2,330개의 태깅된 문장으로 이루어진 데이터 셋을 구축하였다.

또한 구축한 데이터 셋을 이용하여 대표적인 신경망 모델인 합성 곱 신경망(Convolutional Neural Network)[4] 모델, 순환 신경망(Recurrent Neural Network)[5] 모델, BERT[1] 모델을 각각 학습하여 성능을 평가하였으며, 학습 결과 BERT 모델이 검증 데이터와 평가 데이터에 대해 각각 약 95.30%, 93.75%로 가장 높은 성능을 보였음을 확인하였다.

## Ⅱ. 관련 연구

웹 크롤러를 이용하여 말뭉치를 구축하는 과정은 수집한 HTML 파일로부터 문장을 추출하는 과정과 추출

된 문장 중 노이즈가 포함된 문장을 필터링 하는 과정으로 나눌 수 있다.

먼저 수집한 HTML 파일로부터 문장을 추출하는 과정에 대한 기존 연구들은 DOM(Document Object Model) 트리로 HTML 파일을 변환 후 트리 내 노드들 내 포함된 문장을 추출하는 구조적 특징 해석을 이용한 추출 방법[6]과 SVM (Support Vector Machine), 신경망 모델 등 기계 학습 기반의 추출 방법[7,8]을 주로 제안하였다.

먼저 구조적 특징 해석을 이용한 방법은 전체 DOM 트리 노드 내 문장들을 추출하는 경우, 불필요한 단어들도 함께 추출되는 문제점이 발생할 수 있으며, 이를 완화하기 위해 제안된 연구[6]는 문장이 포함되어 있을 가능성을 지닌 후보 블록들을 추출하기 위한 통계 기반의 영역 구분 단계와 후보 블록의 앞, 뒤 블록들의 태그를 분석하여 본문을 찾아내는 규칙 기반의 구조 분석 단계로 이루어진 2단계 추출 방법을 제안하였다.

기계 학습 기반의 추출 방법 중 SVM 알고리즘을 이용한 연구[7]는 DOM 트리 내 각 노드들의 문장 정보, 노드들의 위치와 노드가 갖고 있는 속성을 입력 특징으로 사용하여 문장을 포함한 노드들을 추출하였다. 다른 방법으로 노드들이 문장을 포함하고 있는지 아닌지 이진 분류(Binary Classification) 작업을 통해 문장을 추출한 연구[8]는 합성 곱 신경망 모델을 사용하였으며, DOM 트리의 노드에 해당하는 태그를 입력으로 사용하였다.

## Ⅲ. 웹 말뭉치 수집 및 구축 방법

문장 필터링 데이터 셋을 구축하기 위한 말뭉치를 구축하기 위해 본 논문에서는 말뭉치 수집을 위한 웹 크롤러를 자체적으로 제작하였으며, 이에 대한 웹 크롤러의 구성 요소 및 동작 방식에 대해 간단히 언급하고자 한다.

말뭉치 수집을 위한 웹 크롤러는 웹 사이트에 접근하여 HTML 파일들을 수집하는 에이전트(Agent), 수집한 HTML 파일 내 포함된 다음 에이전트가 방문할 URL 목록과 문장 추출 및 저장을 위한 작업자(Worker)와 에이전트가 다음 방문할 URL들을 결정하는 스케줄러(Scheduler)로 세 가지 구성 요소를 포함하고 있다.

웹 크롤러는 초기에 처음 방문할 Seed URL들을 입력해주어야 하며, 총 세 가지 동작을 반복적으로 수행하여 말뭉치를 수집한다. 먼저 스케줄러는 에이전트가 방문할 URL들을 결정하여 에이전트에게 전송하며, 두 번째로 에이전트는 전달 받은 URL들을 이용하여 HTML 파일을 수집하고 작업자에게 수집 결과를 전송한다. 마지막으로 작업자는 HTML 파일 내 포함된 URL 목록을 추출하여 스케줄러에게 전송하고, 문장을 추출하여 말뭉치 저장소에 저장한다. 이러한 일련의 과정이 반복되면서 시간이 지남에 따라 웹 크롤러는 수많은 웹 사이트를 방문하여 다양한 도메인에 대한 대량의 말뭉치를 수집할 수 있다.

## Ⅳ. 문장 필터링 데이터 셋 구축 방법

문장 필터링 데이터 셋 구축 목표는 웹으로부터 수집 및 필터링 과정을 거친 말뭉치 내 문장들 중 각 문장이 완전한 문장인지 아닌지 신경망 모델이 학습 및 평가할 수 있는 문장 필터링 데이터 셋을 구축하는 방법을 제안하는 것이며, 문장 필터링 데이터 셋 구축 방법을 각 문장에 대해 문장 속성에 따른 필터링 조건과 HTML 문법에 따른 필터링 조건으로 세분화하여 필터링할 수 있도록 제안한다.

본 논문에서 정의한 완전한 문장은 한국어 문장으로 한정하며, 하나의 주어부와 서술부로 이루어져 있으며, 말뭉치 내 각 문장들을 독립적인 문장들로 보았을 때 “누가”, “무엇을”, “했다”등과 같이 정확한 의미 전달이 가능한 문장이여야 한다. 또한 문장의 구성 요소 중 동사가 타동사인 경우 동사의 행위자, 동사의 대상이 되는 사람 또는 사물이 모두 포함되어 있는 문장을 포함해야 한다.

여기서 문장은 다양한 사용자들이 작성한 웹 사이트 내 문장들은 다양한 패턴으로 이루어져 있기 때문에 조사 일부 생략, 따옴표로 문장 일부 또는 문장 전체를 감싸는 경우 등에 대해서는 따로 기준을 두지 않고 유연하게 앞서 언급한 조건들에만 해당한다면 완전한 문장으로 간주한다.

표1은 말뭉치 내에 나타날 수 있는 문장과 해당 문장이 완전한 문장인지 아닌지에 대한 태그에 대한 예시이다. 표에 있는 태그 중 O는 완전한 문장, X는 불완전한 문장을 나타낸다.

표의 1번 문장은 하나의 주어부와 서술부로 이루어

져 있으며, 정확한 의미를 전달할 수 있기 때문에 완전한 문장이다. 2번 문장은 “이 꽃의 향기가 좋다.”라는 문장에서 조사가 생략된 경우이며, 조사가 일부 생략되더라도 정확한 의미를 전달할 수 있기 때문에 완전한 문장이라 할 수 있다.

**Table. 1** Each Cases about Sample Sentences

| No | Sentences | Tag |
|---|---|---|
| 1 | 나는 영희와 밥을 먹었다. | O |
| 2 | 이 꽃 향기가 좋다 | O |
| 3 | 얼떨결에 제안을 승인하였다. | X |
| 4 | 서울에 사는 철수는 학교를 | X |
| 5 | 공부를 위해 책을 골랐다. | X |
| 6 | 과목 선택을 위해 나는 샀다. | X |
| 7 | I like to study natural language processing | X |

3번과 4번 문장은 각각 주어와 서술어가 빠져 있으며, 하나의 주어부와 서술부로 이루어져야하는 완전한 문장 기준에 부합하지 않기 때문에 불완전한 문장이다. 5번, 6번 문장은 각각 동사에 대한 행위자 또는 대상이 빠져 있기 때문에 불완전한 문장이라 볼 수 있다. 또한, 3, 5, 6번 문장은 문서 또는 문단 관점에서 생략된 단어들을 유추할 수 있지만, 본 논문에서 완전한 문장은 하나의 문장에 대해 독립적인 의미 전달이 가능해야하기 때문에 완전한 문장이 되기 어렵다.

마지막으로 정의한 완전한 문장은 한국어 문장으로 제한하였기 때문에 7번 문장처럼 올바른 문장 구조를 갖고 있고, 의미 전달에 문제가 없더라도 타 언어로 이루어진 문장은 완전한 문장에 해당하지 않는다.

또한 완전한 문장에 부합하더라도 다양한 문장 패턴이 포함될 수 있는 말뭉치에 대해 본 논문에서는 문장과 HTML 문법에 따라 예외 조건을 추가적으로 설정하였으며, 예외 조건을 하나라도 만족할 시 완전한 문장이 아닌 것으로 간주한다.

### 4.1. 문장에 따른 필터링 조건

4.1.1. 중의적 문장인 경우

모호한 의미를 가진 중의적 문장은 완전한 문장의 조건인 정확한 의미 전달 조건에 부합하지 않고, 사람에 따라 각기 다른 해석이 존재할 수 있기 때문에 불완전한 문장으로 설정하였다. 예를 들어, “홍길동은 어제 그와 축구한 친구를 만났다.” 라는 문장이 주어졌을 때, 홍길동이 어제 시점에 친구를 만났는지, 어제 함께 축구한 친구를 현재 시점에서 만난건지 다양한 해석을 할 수 있기 때문에 완전한 문장에서 제외한다.

4.1.2. 맺음말이 없는 경우

웹 페이지에 작성된 문장의 경우, 사용자들이 작성한 문체에 따라 맺음말이 생략될 수 있다. 예를 들어 “정부의 사회적 거리두기가 단계별로 대응”이라는 문장에서 “대응”이라는 단어는 “대응한다.”라는 단어에서 “한다.”라는 단어가 생략된 경우일 수 있다. 완전한 문장은 유연한 기준을 갖고 일부 조사 생략은 가능하지만 이처럼 맺음말이 없는 문장의 경우는 완전한 문장이 아니다.

4.1.3. 타 언어와 혼용된 경우

정의한 완전한 문장은 한국어로 한정하였기 때문에 문장의 전체가 타언어로 이루어지지는 않지만, 문장 또는 단어의 의미를 정확하게 전달하기 위해 문장 내 일부 한글 단어를 타 언어 단어가 수식하는 경우가 존재한다. 이러한 경우는 제시한 완전한 문장의 조건에 부합한다면 후처리 과정에서 수식 단어를 제거할 수 있기 때문에 완전한 문장으로 간주한다. 하지만 단어를 한글 단어가 아닌 타 언어의 단어를 사용하여 문장을 표현한 경우는 불완전한 문장으로 간주한다. 예를 들어 “우리는 문장 필터링(Filtering) 방법을 제안한다.”라는 문장은 “Filtering”이라는 영단어가 한글 단어를 수식하고 있기 때문에 완전한 문장으로 볼 수 있다. 또한 “문장 분리를 위해 Filtering 방법을 사용한다.” 라는 문장에 대해서는 Filtering이라는 영어 단어를 사용하여 문장을 표현하고 있기 때문에 이러한 경우는 불완전한 문장이다.

### 4.2. HTML 문법에 따른 필터링 조건

4.2.1. 중의적 문장인 경우

문장을 일렬로 정렬하여 목록 형식으로 표현하거나 범주별로 문장들을 묶어 테이블 형식으로 사용하기 위해 HTML 태그 중 <ul>, <li>, <table> 등의 태그들이 사용되어지며, 이러한 태그들은 태그를 제거하더라도 태그 내의 문장이 남아 있을 수 있다.

예를 들어 “<li>면접 : </li><li>본 회사는 3단계 면접 프로세스를 따릅니다.</li>”라는 HTML 소스 코드에 대해 <li>, </li> 태그를 제거할 시 “면접 : 본 회사는 3단계 면접 프로세스를 따릅니다.”라는 문장이 추출되어질 수

있으며, 이러한 경우 불완전한 문장의 일부가 완전한 문장으로 이루어져 있을 수 있다. 본 논문에서는 목록 또는 테이블 형식에 대해 ":", "-"와 같은 특정 기호가 포함되어 있고 기호를 기준으로 분리하였을 때 분리된 두 문장 모두 완전하지 않은 문장인 경우 불완전한 문장으로 간주하며, 예시 문장처럼 ":" 기호를 기준으로 분리하였을 때 오른쪽에 위치하는 문장이 완전한 문장이라면 후처리 과정을 통해 문장을 분리할 수 있기 때문에 완전한 문장이다.

또한 다른 예시로 "(1) 회사는 회원에게 발생한 손해에 대한 책임을 부담하지 않습니다." 와 같이 목록의 숫자와 함께 포함된 문장의 경우 또한 후처리 과정을 통해 "(1)" 과 같은 특정 숫자들은 제거할 수 있기 때문에 완전한 문장이다.

4.2.2. 중요 정보를 포함하지 않은 경우

HTML 파일에서 추출된 문장들에 대해 문장 분리기를 이용하여 문장을 분리하는 과정에서 문장 분리기의 오류로 인해 문장의 일부가 따로 분리되는 경우가 발생할 수 있다. 또한 이 경우, 분리된 문장들이 완전한 문장일지라도 문장 의미를 해석하는데 있어서 중요 정보가 누락되어 있는 경우 불안전한 문장으로 간주한다.

또한 "이러한 정보는 아래 목록과 같습니다."와 같은 문장처럼 목록, 테이블 형식 이전에 주로 나타나는 문장인 경우 완전한 문장일지라도 문장이 특정 중요 정보를 갖고 있지 않은 경우 불완전한 문장이다. 하지만 예외로 "이러한 정보는 아래와 같이 4차 산업 혁명에 관련된 것입니다."와 같이 아래 목록에 대한 내용을 언급하면서 목록 또는 테이블 정보의 목표 또는 속성과 같은 정보를 포함하고 있는 경우는 완전한 문장이다.

4.2.3. 문장이 생략된 경우

웹 페이지 내 문장을 표현하기에 공간이 한정된 경우에는 텍스트 오버플로우 속성을 사용하여 긴 문장의 일부를 잘라내는 경우가 있다. 주로 뉴스나 블로그의 제목에서 나타나는 경우가 많으며, 잘려진 문장은 대부분 문장의 마지막에 "..."과 같이 생략 기호가 추가되어진다. 텍스트 오버플로우 속성을 사용하면 한국어의 구조에 따라 대부분 문장의 마지막에 위치하는 서술어가 생략될 가능성이 높고, 중요 정보가 제외될 수 있기 때문에 불완전한 문장으로 간주한다.

예를 들어 원 문장 "저희 회사는 엔지니어 신입사원을 채용합니다."에서 텍스트 오버플로우 속성을 사용하는 경우 "저희 회사는 설계 엔지니어 신입 사원을 채용..."과 같이 처리될 수 있으며, 이는 문장의 서술어가 제거되었기 때문에 불완전한 문장이다.

4.2.4. 명사형 단어가 나열된 경우

명사형 단어가 한 문장의 일부 나열되거나 전체를 이루고 있는 경우는 HTML 파일에서 문장을 추출하는 과정에서 빈번하게 나타나며, 특히, HTML 코드 내 태그들이 한 줄로 이루어져 있는 경우에 주로 나타난다. 이처럼 명사형 단어가 문장의 일부 또는 문장 전체가 쉼표와 같은 기호 없이 명사형 단어 나열로 이루어져있는 경우는 불완전 문장으로 간주한다. 하지만 완전한 문장의 조건에 위배되지 않으며, 의미 전달이 가능하면 일부 명사형 단어를 복합 명사로 간주하여 완전한 문장으로 본다.

예를 들어, "<a>로그인</a><a>회원가입은 이 버튼을 눌려주세요</a>"라는 HTML 코드가 주어졌을 때, "로그인 회원가입은 이 버튼을 눌려주세요"라는 문장이 추출될 수 있다. "로그인 회원가입"은 존재하지 않은 복합 명사이며, "로그인"과 "회원가입"을 구분 짓는 쉼표 기호 또한 없기 때문에 불완전한 문장이다. 하지만 "(개인 사생활 보호) 우리 회사는 개인의 사생활 보호를 노력 한다."와 같은 문장 내 특정 기호로 감싸져 있는 명사형 단어들이 나열되어 있는 경우 타 언어와 혼용된 경우와 동일하게 후처리 과정에서 제거할 수 있기 때문에 완전한 문장이다.

## Ⅴ. 문장 필터링 데이터 셋 구축 결과

앞서 언급한 문장 필터링 데이터 셋 구축 방법을 이용하여 본 논문에서는 문장 필터링 데이터 셋을 자체적으로 구축하였다.

그림1은 데이터 셋 구축을 위해 개발한 문장 태깅 툴이며, 수집한 말뭉치 내 문장 리스트, 태깅하고자 하는 문장, 완전한 문장인지 아닌지에 대한 선택할 수 있는 버튼으로 구성되어 있다.

태깅 작업은 총 두 명의 작업자가 태깅, 검증 작업으로 나누어 수행하였으며, 데이터의 신뢰도를 높이기 위해 한 작업자가 태깅 작업을 하면 다른 작업자가 태깅된 결과를 검증하는 방식으로 구축 작업을 실시하였다.

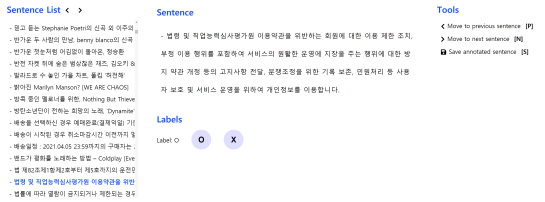


**Fig. 1** Sentence Filtering Annotation Tool

구축된 데이터 셋은 웹으로부터 수집한 말뭉치 내 각 문장들과 레이블 정보를 포함한다. 표2는 구축한 데이터 셋에 대한 통계 정보를 보여준다. 데이터 셋에 포함된 문장은 총 2,330개이며, 완전한 문장 474개와 불완전한 문장은 1,856개를 포함한다. 수집된 말뭉치 내 다양한 노이즈가 포함된 문장이 본 논문에서 정의한 완전한 문장에 비해 많이 포함되어 있었기 때문에 상대적으로 완전한 문장의 개수가 적다.

## Ⅵ. 신경망 모델 성능 평가

완전한 문장을 필터링하는 작업은 완전한 문장인지 아닌지 분류하는 문장 분류 작업으로 볼 수 있으며, 구축한 문장 필터링 데이터 셋에 대해 성능 베이스라인을 설정하기 위해 문장 분류 모델들의 성능을 비교하였다. 모델은 문장 분류 모델은 대표적인 신경망 모델인 합성곱 신경망 모델, 순환 신경망 모델, BERT 모델을 사용하여 성능 평가를 실시하였으며, 성능은 정확도를 이용하였다.

### 6.1. 실험 환경

학습은 구축한 데이터 셋을 무작위로 섞은 후, 8:1:1 비율로 나누어 학습, 검증, 평가 데이터 셋으로 사용하였으며, 학습은 총 15회 진행하였다. 학습 과정에서 조기 종료(Early Stopping) 방법은 사용하지 않고, 검증 데이터로 각 학습 횟수마다 성능 평가를 하여 가장 높은 성능을 보인 모델에 대해 평가 데이터로 성능 평가를 진행하였다. 문장의 최대 길이는 256으로 설정하였으며, 문장을 모델에 입력하기 위해 합성 곱 신경망과 순환 신경망은 형태소 분석 전처리 과정을 수행하여 토큰으로 분리하였으며, BERT 모델은 기존 BERT 모델에서 사용한 토큰 분리 방법을 동일하게 사용하여 문장을 분리한 뒤, 문장 분류를 위한 [CLS] 토큰을 함께 추가하여 사용하였다.

각 모델에서 사용되는 파라미터에 대해 먼저 합성 곱 신경망 모델은 임베딩 층의 크기를 64로 설정하였으며, 윈도우 사이즈를 3, 4, 5로 설정하고, 필터의 크기를 100으로 설정하였다. 다음으로 순환 신경망 모델은 임베딩 층의 크기를 64로 설정하였고, 은닉 상태를 256으로 설정하였다. 또한 순환 신경망 모델의 구조 중 GRU[5] 모델을 사용하였으며, 2층 양방향 구조를 사용하였다. 마지막으로 BERT 모델은 Huggingface[9]에서 제공한 BERT 모델을 사용하였다.

### 6.2. 신경망 모델 성능 평가

표2는 검증 데이터와 평가 데이터에 대한 각 모델의 성능 평가 결과이다. 사전 학습된 BERT 모델은 검증 데이터와 평가 데이터에 대해 각각 약 95.30%, 93.75%로 3개의 신경망 모델 중 가장 높은 성능을 보였으며, 순환 신경망 모델은 약 92.74%, 90.00%로 가장 낮은 성능을 보였다.

**Table. 2** Accuracy of Each Models(%)

| Models | Valid Dataset | Test Dataset |
|---|---|---|
| CNN | 93.16 | 92.50 |
| RNN | 92.74 | 90.00 |
| BERT | 95.30 | 93.75 |

## Ⅶ. 결 론

사전 학습된 모델은 학습 과정에서 대량의 말뭉치 내 문장들의 언어학적 패턴을 스스로 학습함으로써, 입력 문장 내 각 토큰들을 적절한 특징 벡터로 표현할 수 있다는 장점을 갖고 있으며, 작업에 맞게 처음부터 학습하는 모델보다 더 높은 성능을 보였다. 또한 높은 성능을 보인 대표적인 사전 학습 모델인 BERT 모델의 경우 약 16GB 정도의 말뭉치로 학습했을 정도로 대량의 말뭉치는 필수적으로 필요하다.

본 논문에서는 신경망 모델이 웹으로부터 수집한 말뭉치에 대해 완전한 문장과 불완전한 문장을 필터링할 수 있도록 학습을 위한 문장 필터링 데이터 셋 구축 방법에 대해 제안하였다. 또한, 웹 사이트에서 나타나는 문장은 다양한 사용자들로부터 작성한 다양한 패턴들

로 이루어져 있기 때문에 유연한 기준을 세워 완전한 문장에 대한 정의를 하였다.

데이터 셋 구축은 문장을 문장에 따른 방법과 HTML 문법에 따른 방법에 따라 완전한 문장과 불완전한 문장을 구분하여 총 2명의 작업자가 자체 개발한 문장 태깅 툴을 이용하여 태깅 작업을 수행하였다.

그 결과, 문장 474개와 불완전한 문장 1,856개가 포함된 총 2,330개의 태깅된 문장으로 이루어진 데이터 셋을 구축하였다. 또한 구축한 데이터 셋을 이용하여 대표적인 신경망 모델인 합성 곱 신경망 모델, 순환 신경망 모델, BERT 모델을 각각 학습하여 성능을 평가하였다.

학습 결과, BERT 모델이 검증 데이터와 평가 데이터에 대해 각각 약 95.30%, 93.75%로 가장 높은 성능을 보였음을 확인하였다.

현재는 문장 필터링 데이터 셋이 약 2천개의 완전한 문장과 불완전한 문장을 포함하고 있지만, 추후 지속적으로 태깅 작업을 실시하여 데이터 셋을 확장하고자 하며, 실험 결과 중 가장 높은 성능을 보인 BERT 모델보다 더 높은 성능을 보이는 모델에 대한 연구와 데이터 불균형에 대한 문제에 대한 연구를 진행하고자 한다.

## ACKNOWLEDGEMENT

This work was supported by the 2020 sabbatical year research grant of KoreaTech.

## REFERENCES

[ 1 ] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, San Francisco, pp. 4117-4186, 2019.

[ 2 ] Z. Yang, D. Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized Autoregressive Pretraining for Language Understanding,” in *Proceedings of the 33rd Conference on Neural Information Processing System (NeurIPS)*, Vancouver, pp. 5754-5764, 2019.

[ 3 ] Wikipedia, Wikipedia Dump Data [Online]. Available: https://www.wikipedia.org/.

[ 4 ] P. W. Park, “Text-CNN Based Intent Classification Method for Automatic Input of Intent Sentences in Chatbot,” *Journal of Korean Institute of Information Technology*, vol. 18, no. 1, pp. 19-25, Jan. 2020.

[ 5 ] J. M. Kim and J. H. Lee, “Text Document Classification Based on Recurrent Neural Network Using Word2vec,” *Journal of Korean Institute of Intelligent Systems*, vol. 27, no. 6, pp. 560-565, Dec. 2017.

[ 6 ] H. J. Jeon and C. Koh, “Text Extraction Algorithm using the HTML Logical Structure Analysis,” *The KDCS Transactions*, vol. 16, no. 3, pp. 445-455, Jun. 2015.

[ 7 ] N. Utiu and V. S. Lonescu, “Learning Web Content Extraction with DOM Features,” in *Proceedings of the 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing(ICCP)*, Doha, pp. 1724-1734, 2014.

[ 8 ] B. D. Nguyen-Hoang, B. T. Pham-Hong, Y. Jin, and P. T. V. Le, “Genre-Oriented Web Content Extraction with Deep Convolutional Neural Networks and Statistical Methods,” in *Proceedings of 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong, pp. 476-485, 2018.

[ 9 ] Huggingface. Transformers [Internet]. Available: https://www.github.com/huggingface/.

**남충현(Chung-Hyeon Nam)**

2019년 한국기술교육대학교 컴퓨터공학부 공학사
2019년 ~ 현재 한국기술교육대학교 컴퓨터공학과 공학석사과정
※관심분야 : Natural Language Processing, Image Processing, Incremental Learning, Data Mining

**장경식(Kyung-Sik Jang)**

1987년 고려대학교 전자공학과 공학사
1989년 한국과학기술연구원 전기전자공학과 공학석사
1998년 동경공업대학 전기전자공학 공학박사
1999년 ~ 현재 한국기술교육대학교 컴퓨터공학부 교수
※관심분야 : Embedded System, Natural Language Processing, Incremental Learning