

임상문서 개체명인식 모델을 적용한 클라우드 의료정보시스템 구축

이정훈¹, 김상우¹, 김영욱¹, 김진우², and 최소운¹

¹컴퓨터정보공학부, 광운대학교

²컴퓨터공학과, 광운대학교

요 약

이 프로젝트를 통해 새로운 형태의 클라우드 의료 정보 시스템(이하 CHIS)을 제시한다. 우리는 HIS(의료전산망)의 구성요소 중 환자에 대한 진료/처방 데이터를 관리하는 서비스인 전자 의무기록(이하 EMR)에 집중한다. 이번 프로젝트에서는 EMR에 개체명 인식(이하 NER) 기능을 제공하는 모듈을 접목하여 방대한 분량의 임상 데이터 속 주요 정보만을 제시하는 방법으로 기존 EMR을 개선한다.

구현체는 *CHIS Cluster*와 *NER Module*로 구성된다. CHIS는 *MSA*를 차용하여 구성요소의 확장성을 개선한 HIS를 의미한다. 또한, 시스템 내에서 비식별화 및 Labeling된 의료데이터를 NER Module의 학습에 활용하여 서비스의 지속적 개선 및 통합이 가능하도록 한다.

NER Module은 *n2c2 2010 dataset*과 *BERT* 모델을 사용하여 구현한다. BERT모델에 의료 도메인 텍스트를 사용하여 *Fine tuning*을 진행하고, 개체명 인식을 수행하는 모델을 구현한다. 학습된 모델은 임상문서 내 진단 내역, 처방 내역, 치료 내역을 포함하는 개체명 인식, 발화자와의 연관도 분석, 개체명 간 관계분석을 통해 의료진이 중요한 정보를 빠르게 인지하고 분석하는 것을 돕게 된다.

제안한 CHIS를 통해 의료진의 업무부담 경감 및 의료서비스 질적 개선이 예상되며, 의료 데이터의 외부반출을 가능케 하여 의료 데이터를 활용한 의료인공지능 기술연구 환경개선을 가져올 것으로 기대된다.

주제어: 개체명인식, 의료인공지능, 클라우드, 헬스케어

차 례

제 1 장 서론	2
1.1 배경	2
1.2 개발목표	2
1.2.1 CHIS	2
1.2.2 NER 모듈	2
1.3 보고서 구성	2
제 2 장 관련연구	3
제 3 장 기술명시	3
3.1 개체명 인식	3
3.2 BERT: Bidirectional Encoder	3

3.3 DistilBERT	4
3.4 HIS	4
3.5 EMR	4
3.6 Web Application	4
제 4 장 시스템 구성도	4
4.1 web container	5
4.2 AI module	5
4.3 시스템 운용 시나리오	5
제 5 장 설계 및 구현	5
5.1 데이터셋 설명 및 전처리 결과	5
5.2 학습용 모듈 설계 및 구현	6
5.3 대화형 모듈 설계 및 구현	6

5.4	Web Service 구조	7
5.4.1	NginX	7
5.4.2	Django Backend server	7
5.4.3	web service 동작 시나리오	7
제 6 장	시스템 검증	8
6.1	F1 score	8
6.2	NER 모델 대한 통계적 검증	8
6.3	Ner added EMR 검증	9
제 7 장	결론	9

제 1 장 서론

1.1 배경

지난 수년간 전자의무기록의 전산화, 수집을 수행하는 데이터 웨어하우스의 실용화가 이루어 졌다. 하지만 축적한 임상데이터에 대한 인공지능 서비스의 실제 활용사례는 찾아보기가 어렵다. 이러한 배경에는 데이터 비식별화의 어려움을 들 수 있다. 대표적인 사례로 EMR은 환자 개인정보와 환자의 진찰기록이 같은 문서에 병기되어, 반출 및 활용에 어려움이 있다.

이와 동시에, 컴퓨팅 성능의 비약적 발달과 함께 덩어리로 대표되는 인공지능의 발전 또한 이루어졌다. 하지만 의료분야의 인공지능 서비스 도입은 더딘 편이다. 하지만, 인공지능이 의료현장에서 해결할 문제는 분명히 존재한다. 일례로, 많은 병원에서 영상분석 전문의의 부족으로 의료영상 분석에 막대한 시간이 소요된다. 또, 수많은 임상 이미지/텍스트 속에서 중요 정보만을 식별해 내는데 human-error의 가능성 또한 배제할 수 없다.

따라서 이 프로젝트에서는 의료데이터 전처리(비식별화 등)를 수행하는 CHIS와 그 구성요소로 동작하는 NER(개체명인식) 인공지능 모듈을 제안한다. 이를 통해 EMR내 주요 텍스트를 검출하여 의료진의 임상문서 분석을 돕는다.

1.2 개발목표

1.2.1 CHIS

CHIS는 인공지능 모듈, 웹서비스 컨테이너를 갖는 클라우드 클러스터의 형태로 구현한다. EMR 웹서비스를 통해 의료기록을 입력하면 클러스터 내부 큐(queue)를

통해 인공지능 모듈로 데이터가 전달되고, 전달된 데이터를 분석한 결과를 EMR 서비스에서 확인해 볼 수 있다.

이러한 구조를 취함으로써 종래의 Monolithic한 구조의 전산 시스템의 여러 한계점을 개선한다. 가변적인 work-load에 따라 컨테이너의 수를 scaling하여 대응할 수 있다. 또, 시스템 노후화에 의한 migration을 수행하거나 시스템 업데이트를 하는 경우에도 무중단 서비스를 제공할 수 있다.

1.2.2 NER 모듈

본 프로젝트에서는 EMR 서비스를 개선하기 위해 BERT를 이용하여 NER 모듈을 구현한다. BERT는 Transformer의 Encoder를 적층시켜 쌓아올린 사전학습 언어 모델로, 기존의 단방향 자연어 처리 모델들을 보완한 양방향 자연어 처리 모델이다. BERT를 사용하여 언어 이해에 필요한 학습량을 대폭 감소시킬 수 있었고, 본 연구에서는 BERT에 fine-tuning layer를 추가하여 NER(개체명인식)을 구현한다.

NER은 장문의 텍스트로부터 개체명을 추출하는 자연어 처리 기법이다. 이 연구에서는 EMR의 주요 정보를 추출하기 위해 질병명(PR), 진단명(TR), 검사명(TE) 3가지의 태그로 구분하였다. 또한, 토큰 구분을 위해 B/I/O 세가지 태그를 추가적으로 사용한다.

이렇게 정의된 모델 입력 규격에 알맞도록, 준비된 데이터셋(n2c2 2010 dataset)을 전처리하여 학습을 수행한다. 모델 학습에 대한 검증은 F1 score를 통해 이루어진다.

F1 score는 인공지능 모델의 실효성을 통계치로 제시한 것이므로, EMR서비스에서 NER모듈의 유용성을 입증하기 위해선 F1점수를 높여야 한다. 이번 프로젝트에서는 BERT를 대조군으로, DistilBERT, BioBERT 등을 비교군으로 하여 가장 유용한 모델이 무엇인지를 비교/분석 하였다.

1.3 보고서 구성

이 보고서의 2장에서는 프로젝트 관련 연구를 예시로, 주제 선정근거와 시스템 설계 배경 등에 대해서 다룬다. 3장에서는 프로젝트에서 요구되는 기술적 배경에 대해서 다룬다. 4장에서는 시스템 아키텍처를 다룬다. 5장에서는 요구사항 단위로 상세 설계된 CHIS, EMR 서비스, NER 모듈 등 시스템 구성요소와 구성요소간 상호작용을 중심으로, 사용자별 EMR / NER서비스의 사용 시나리오를 살펴본다. 6장에서는 CHIS와 NER 모듈을

통해 의료전산시스템을 얼마나 개선하였는지를 다룬다. 7장에서는, 프로젝트 요약 및 결론을 다룬다.

제 2 장 관련연구

BERT를 기반으로 하는 BioBERT 모델을 사용하여 EMR에서 Problem, Test, Treatment에 해당되는 개체명 추출 선행 연구[1]가 있다. I2B2 2010 challenge dataset을 사용하여 연구를 진행하였으며 여러 모델의 조합으로 실험하여 Precision, Recall, F1 score를 비교한다. 먼저, 임상 텍스트를 수치 벡터로 변환하기 위해 의료 관련 분야의 말뭉치에 대해 사전 학습을 시킨다. 그 후에 BiLSTM-CRF 모델을 사용하여 처리된 벡터를 통해 학습을 하고 태깅한다.

실험 결과에 따르면 BioBERT, BiLSTM, CRF 모델을 사용했을 때 F1 score가 87.10%로 제일 높게 나왔다. 이를 통해 임상 문서 내 개체명 인식의 정확도를 높이기 위해서는 어떤 모델을 사용해야 좋은 성능을 낼 수 있는지 확인할 수 있었다. 하지만 해당 연구에서는 모델 개선(F1 score 개선)에만 포커스를 맞추고 있어서, 다른 서비스와의 연동에서 해당 모델이 사용자 경험(UX) 관점에서 유의미한지에 대한 분석은 언급되어 있지 않았다. 따라서, 개체명 인식을 EMR 서비스에 연동하여 해당 서비스를 사용자 관점에서 실제로 개선하는 것을 이번 프로젝트에서 진행하였다.

의료 인공지능 연구/개발과 실용화를 위한 지능형 병원정보시스템 모델은 데이터 형식에 따른 시스템간의 연계 및 결과 UI 등을 독립적으로 개발하여 구현한다. 하지만 인공지능 개발과 실용화까지의 전반적인 시스템에 대한 연구는 부족하다. 선행 연구[2]에서 제시하는 실시간 센서기기와의 연계 방안으로 AI Module과 EMR의 실시간 연계가 있다. 구현한 모델은 AI Platform에서 분석을 실시간으로 처리하고 결과를 EMR DB에 저장하는 형식이다. 인공지능과 조화된 병원정보시스템은 의료 서비스질 향상을 제공한다. 이번 프로젝트는 EMR과 NER 모듈이 연계된 모델을 웹서비스로 구현하여 통해 유기적 결합이 고려된 지능형 모델을 제안한다. 이를 통해 기존의 장점에 이어 실용성, 사용자 요구의 정확성, 최종 사용의 편리성을 제공한다.

제 3 장 기술명시

3.1 개체명 인식

개체명 인식(Named-Entity Recognition, NER)이란 태그를 가진 개체를 인식하여 분류하는 정보 추출의 일종이다. 개체명 인식 학습은 지도 학습이므로, 사용되는 데이터는 분류 기준을 바탕으로 Tagging이 진행되어 있어야 하며, Tag의 종류로 품사 태깅, 청크 태깅, 개체명 태깅 등이 존재한다. 정확도를 평가하는 기준으로는 precision, recall, F1 score 등이 존재한다.

tagging의 기법에 있어서 BIO system과 BIESO system이 존재한다. BIO system의 경우 개체명 시작에 B-(type), 토큰이 개체명 중간에 위치할 경우 I-(type)를, 개체명이 아닐 경우 O를 붙이는 방법이다. BIESO의 경우 BIO에 tag를 추가한 형태로, 개체명의 마지막에 위치할 경우 E-(type), 하나의 토큰이 하나의 개체명인 경우 S-(type)을 붙이는 형태의 기법이다.

개체명 인식의 모델 구조는 총 3단계로, Distributed Representations for input(Input 데이터에 대해 벡터 형태로 변경), Context Encoder(문맥 정보를 Encoding하는 영역으로 CNN, RNN 등 사용), Tag Decoder(태그 정보를 Decoding하는 영역으로 Softmax, CRF 등 사용)라는 총 3개의 프로세스로 구성된다.

3.2 BERT: Bidirectional Encoder

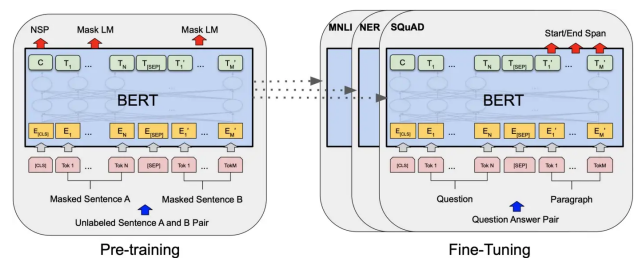


그림 1: BERT모델을 이용한 자연어 처리

BERT[3]는 특정 영역을 학습하기 위해 새로운 신경망을 추가하는 파인튜닝을 통해 해당 작업을 수행할 수 있다. 데이터를 활용하거나 하이퍼 파라미터를 조정하거나 추가 네트워크를 구성하여 파인튜닝을 할 수 있으며, 특정 도메인에 사전 학습된 모델을 구체적인 용도에 맞게 사용하기 위해 추가적인 학습을 시킬 수 있다. BERT의 파생 모델로는 DistilBERT, BioBERT 등이 있으며 본 연구에서는 학습과 추론 과정에서 큰 비용과 자원이 필요하다는 한계를 극복한 경량화된 BERT 모델인

DistilBERT를 이용하였다.

3.3 DistilBERT

DistilBERT[4]는 BERT에 KD(Knowledge Distillation)를 적용한 모델로, 큰 사이즈의 모델인 BERT의 동작을 재현하는 압축된 작은 사이즈의 모델을 훈련시킨다. 기존 BERT 대비 언어 이해 능력을 97%로 유지하면서 크기는 40% 가까이 줄이고, 추론 속도는 60% 더 빠르다. KD의 학습 방법으로는 잘 학습된 모델의 특정 클래스에는 높은 확률을, 학습이 잘 안된 모델에는 0에 가까운 확률을 부여한다. 큰 모델이 출력하는 확률 분포를 작은 모델이 배워 반영할 수 있도록 softmax-temperature를 적용하였다. T 값이 높을수록 확률분포의 평활도(smoothness)를 조정할 수 있으며, T는 큰 모델과 작은 모델에게 적용되며 추론 시에는 제외한다.

$$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \quad (1)$$

그림 2: softmax-temperature

수많은 텍스트 데이터들을 처리하기 위해서 DistilBert를 이용한다. RadioBERT는 흉부사진의 특징 텍스트들을 wiki와 Books로 학습된 distilbert를 이용하여 학습시켜서 문장들로 만들어 판독하는 기술을 이용한 사례가 있다. 이와 같이 bioBERT, ClinicalBioBERT와 같이 DistilBERT를 사용하여 비교분석하는 사례를 찾아볼 수 있다.

3.4 HIS

HIS는 병원의 의료 및 행정 업무를 비롯, 의약품 관리 및 재무 관리를 포함한 운영의 모든 측면을 관리할 수 있도록 설계 포괄적인 통합 정보 시스템을 의미한다. 표준화된 형식은 없지만 데이터 형식 및 데이터 교환을 위한 HL7등의 표준들이 존재한다. 환자 정보 흐름, 의료 제공자의 데이터 접근성의 간소화, 분리된 의료 구성 시스템 통합 등을 위해 설계되었다. 대표적인 구성 시스템은 LIS(검사 정보 시스템), PACS(사진 영상 시스템), CIS(영상 정보 시스템)이 있다.

3.5 EMR

EMR(전자의무기록)은 병원 내 환자에 대한 진료기록을 비롯한 정보들을 전자 문서 등의 형태로 전산화된 의무 기록이다. 입원, 수술, 의약품 처방, 외래 기록 등을

기록하는 진료 EMR, 간호 기록과 임상 관찰 기록 등을 기록하는 간호 EMR, 장비 Interface 및 공인인증의 영역도 이에 포함될 수 있다.

3.6 Web Application

웹 애플리케이션은 어떤 시스템의 기능을 웹 기술을 통해 제공해 놓은 형태를 의미한다. 가장 일반적인 구현 방식은, 하나의 웹 서버에 모든 기능을 집약시키는 Monolithic Architecture을 들 수 있다. 하지만, 이런 고전적인 구조를 개선한 많은 패러다임이 제시되었다.

이 중 프로젝트에 적용된 대표적인 패러다임으로는, EDA(Event Driven Arch), MSA(Micro Service Arch)를 들 수 있다. 가장 기초가 되는 MSA는 복잡한 시스템을 원자적(atomic)인 작은 서비스들로 구분하여 그들 간의 상호작용을 통해 전체 시스템을 구현하는 형태이다. 이러한 방식을 통해 전체 시스템의 모듈화가 이루어져 시스템 복잡성을 낮추는 효과가 있고, 이를 통해 유지보수가 간단해진다는 장점을 갖게 된다.

EDA는 시스템 내에서 발생하는 어떤 이벤트를 추상화하여, 이에 대해 관심 있는 다른 구성요소들이 해당 이벤트를 소비할 수 있도록 하는 구조이다. 이벤트의 수집과 전달은 Broker(중개자)를 통해 이루어지며, Consumer는 Broker가 이벤트 메시지를 전달하는 경우에만 비동기적으로 이를 처리한다. 이와 같은 구조로, MSA에서 특히 문제가 되는 연결 복잡성을 Producer와 Consumer간의 종속성을 낮추는 방식으로 개선할 수 있다.

Vue와 Django의 통신은 기존의 웹에서 이용하는 Session 방식이 아닌 JWT(Json Web Tokens)를 통해 암호화 토큰 방식을 이용한다 토큰 자체에 사용자의 권한 정보와 서비스에 이용하는 정보를 포함시켜 통신하기 때문에 RESTful과 같은 무상태(Stateless)한 환경에서 데이터를 주고받는 경험을 사용자에게 제공한다.

제 4 장 시스템 구성도

시스템을 구성하는 구성요소는 크게 세 가지로 분류해 볼 수 있다. 사용자와 직접 상호작용을 하여 EMR / Authentication 등의 기능을 제공하는 [web container], EMR을 통해 생산된 데이터를 분석하여 결과물을 제공하는 [AI module] 그리고 구성요소 간의 이벤트를 중개하는 [Backbone Layer]이다.

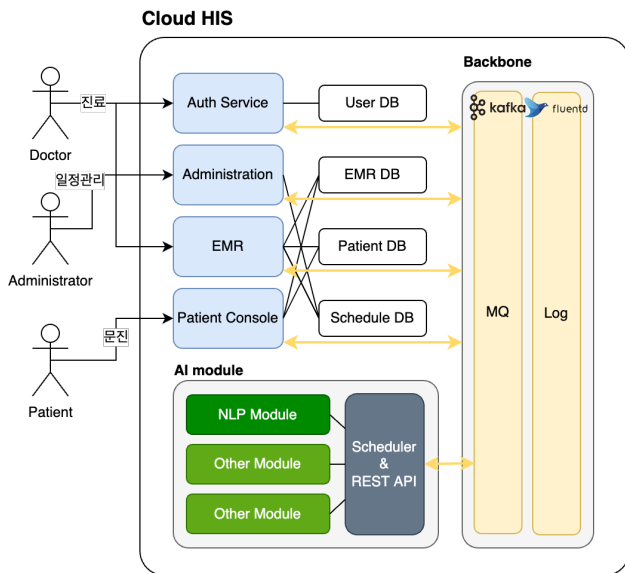


그림 3: CHIS 시스템 아키텍처

4.1 web container

도식에 파란색으로 표시되어 있는 웹 컨테이너들은, HIS가 제공할 기능들을 관심사에 따라 분리한 Micro Service Container이다. 이번 프로젝트의 중추가 될 서비스는 EMR 서비스로, 환자의 진료 / 처방 / 병력 조회 등의 기능을 제공한다. 이러한 과정에서 EMR을 통해 생산된 정보들은 Kafka Producer API를 통해 Kafka cluster의 Broker에게 전달된다.

4.2 AI module

도식에 녹색으로 표시된 AI module은 Kafka Consumer API를 통해, 분석 가능한 Topic을 처리한다. 예를 들어, emr prescript라는 topic을 가진 메시지가 Kafka에 enqueue 되는 경우, 이에 대한 분석이 가능한 NER AI module은 해당 메시지를 분석하여 사용 가능한 형태로 가공한다.

4.3 시스템 운용 시나리오

시스템에서 제공하고자 하는 가장 큰 기능은 clinical text를 입력받아 keyword를 추출하는 기능이다. 진료 시 환자와 의료진의 comment를 텍스트 형식으로 입력받고, 입력받은 텍스트는 Kafka API를 통해 AI module로 전달된다. AI module은 전달받은 텍스트에서 keyword를 추출하고, 그 결과를 EMR database에 저장한다. 이를 도식화하면 아래와 같다.

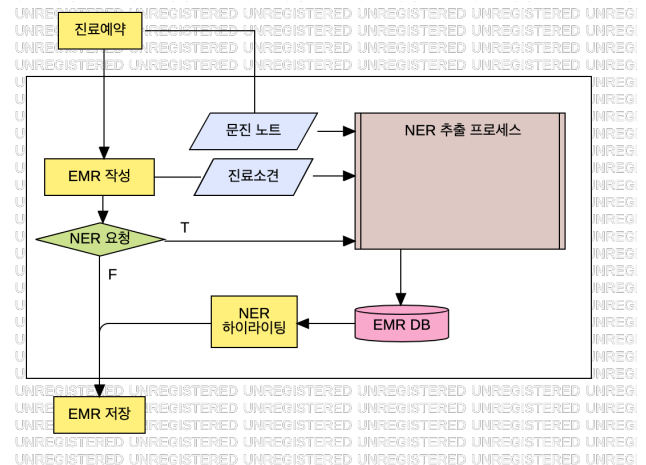


그림 4: CHIS 서비스 사용 시나리오

진료 예약을 미리 한 경우의 시나리오에 대해 살펴본다.

1. EMR서비스를 통해 EMR 문서를 작성하여, 진료 소견정보를 텍스트 형태로 얻어낸다.
2. 진료소견 정보를 이전 진료예약에서 작성한 문진노트와 함께 NER 추출 프로세스로 전달한다.
3. 데이터를 전달하는 과정에 Kafka MQ를 사용한다.
4. NER추출 프로세스에서 얻어낸 NER token을 EMR DB에 저장한다.
5. EMR DB에 저장된 token정보를 바탕으로, EMR 웹 서비스 상에 텍스트 하이라이팅을 수행한다.
6. 완성된 EMR 문서를 저장한다.

제 5 장 설계 및 구현

5.1 데이터셋 설명 및 전처리 결과

NER_dataset.csv라는 파일은, NER_train_module.py에서 load하는 데이터셋이다. 해당 데이터셋은 n2c2 NLP Research Data Sets 2010 Relations Challenge[5]를 원본 데이터셋으로 하여, 전처리한 데이터셋이다.

원본 데이터셋은, 426개(73개 + 97개 + 256개)의 txt 확장자로 작성된 영문 EMR 문서(text 데이터)와 해당 EMR문서 내에서 태깅된 모든 개체명들의 목록과, 특정 개체명이 나타나는 행 번호, 열 번호를 인덱스 범위로 표현한, 426개의 con 확장자로 작성된 개체명 문서(label

데이터)로 구성되어 있다. 426개의 EMR문서들을 구성하는 문장들을 전부 합치면 그 갯수는 4만여개이며, 개체명의 종류로는 질병명(PR), 검사명(TE), 진단명(TR)의 세 종류가 존재한다.

단, 문자열이 주어졌을 때, 이러한 형식의 라벨링 데이터는, BERT가 바로 학습이 가능한 형식이 아니므로, BIO 태그 방식인 text속 word와 label 속 tag들 간의 순차적인 일대일 연결 결과를 나타내는 형식의 라벨링 데이터로 형변환해야 한다. 따라서, 태깅 가능한 가짓수는 B-PR, I-PR, B-TE, I-TE, B-TR, I-TR, O로 총 일곱 가지이다.

해당 데이터셋 내 EMR문서는 정형화된 자연어 형식의 문장들과, 메모에 가까운 비정형화된 단어의 나열들로 구성되어, 정형화 된 텍스트와 정형화 되어 있지 않은 텍스트를 모두 담고 있다는 점에서, 해당 데이터셋으로 학습을 진행한다면 robust한 모델을 얻을 수 있다는 장점이 있다.

5.2 학습용 모듈 설계 및 구현

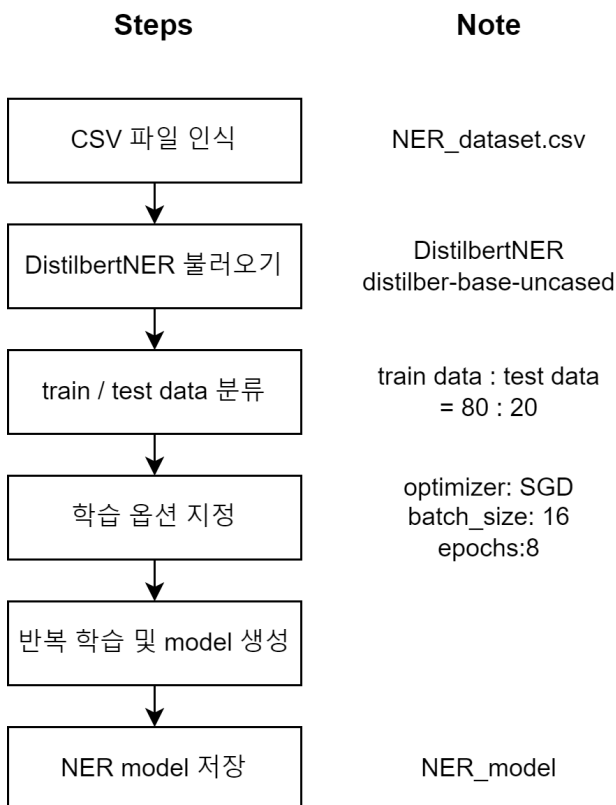


그림 5: NER_train.module.py의 흐름도

NER_train.module.py는 전처리가 완료된 NER_dataset.csv 데이터셋을 통하여 학습을 진행

하고, 학습 결과를 NER_model라는 파일로 export하는 프로그램이다. 따라서 입력이 데이터셋 파일, 출력이 학습이 완료된 모델 파일로 볼 수 있는, jupyter notebook에서 pytorch를 통해 작성된 학습용 프로그램이다.

NER_train.module.py에선 DistilBertForTokenClassification라는 모델을 load한다. 해당 모델은 distilbert-base-uncased를 토큰 분류를 위해 파인 튜닝이 완료된, HuggingFace가 제공하는 Classifier 형태의 모델이다. 임의의 문자열을 해당 모델(인공 신경망)에 입력시키기 위해선, 문자열을 텐서의 형태로 바꿔주는 토큰라이저가 필요하며, 토큰라이저는 DistilBertForTokenClassification과 인터페이스가 일치하는 AutoTokenizer를 이용하였다. 또한, 해당 모델(인공 신경망)의 출력은 임의의 문자열을 구성하는 word들의 태깅 결과(tag)가 된다. 따라서, word들의 목록과 tag들의 목록을 join하면, 임의의 word들로 구성된 입력 문자열로부터, word와 label 속 tag들간의 순차적인 일대일 연결 결과를 출력하는 모델을 얻을 수 있다.

학습은 RTX3080으로 진행되었으며, 초매개변수는 optimizer로 SGD(Stochastic Gradient Descent), batch size로 16, epochs로 8을 사용하였으며, 학습 데이터와 테스트 데이터의 비율을 4:1로 두었다. 학습에는 약 60분이 소모되었고, 학습 결과, 0.819라는 F1점수를 얻었다. 학습 결과는 NER_model이라는 파일로 export하여, 결론적으로는 대화형 프로그램에서 학습된 모델만을 이용하므로, 해당 프로그램은 CHIS 서버 구동을 위한 선행 조건(모델 생성)을 만족시키기 위한 1회용 프로그램이라 볼 수 있다.

최종 학습 결과 분석은 “기능 테스트” 항목에서 진행 하겠다.

5.3 대화형 모듈 설계 및 구현

NER_interact.module.py는 입력 받은 list를 Ner Model 인스턴스가 개체명 인식을 진행하고 화면 상에서 tagging 할 범위를 개체명 종류별 세 list에 담아 반환하는 jupyter notebook에서 pytorch를 통해 작성된 대화형 프로그램이다.

해당 모듈은 NLPmodule이라는 class를 통해 외부(Kafka)로부터 입력을 받고 출력(DB 어댑터)으로 보낼 수 있도록 설계 하였고, GetNerToken 함수를 list형태의 문자 배열을 인자로서 call하는 형태로 이를 구현하였다. 우선 해당 객체는 미리 생성된 NER model 파일을 load 하여 인공신경망을 재구축 한다. 이후 임의의 입력 문

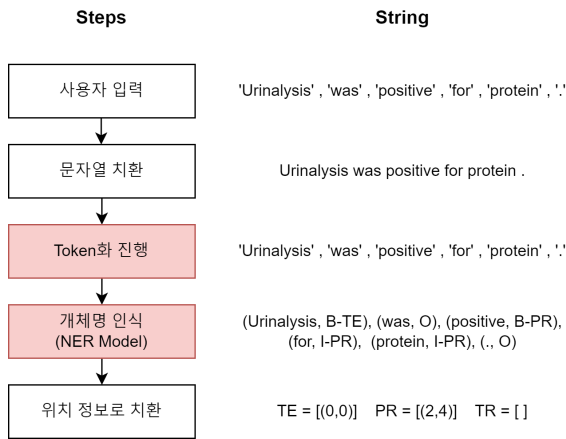


그림 6: NER_interact_module.py의 흐름도

자열에 대해 AutoTokenizer를 기반으로 한 Tokenize를 진행한다. 동시에 BIO System을 기반으로 생성된 label을 인식 후, load한 model을 사용하여 개체명 인식을 진행한다. 실행 결과로 입력 문자열 내 word와 tag 간의 일대일 연결 관계를 의미하는 리스트가 얻어지게 된다. 이를 기반으로 개체명의 순번을 나타낼 수 있도록 각각 Problem, Test, Treatment를 의미하는 세 개의 리스트에 위치를 개체명 각각마다 범위 형태의 정보로 저장한다.

5.4 Web Service 구조

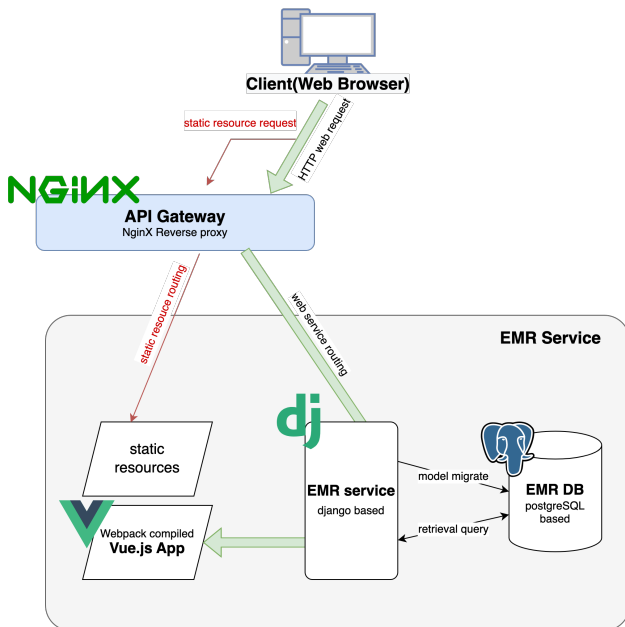


그림 7: Web Service 구성도

앞서 상위 설계에서 언급한 여러 웹 서비스들은 대체로 위와 같은 구조를 갖는다.

5.4.1 NginX

URL에 따라 각자 다른 ip, port 주소에서 실행되고 있는 서비스에 요청을 routing하는 한다. 또한, static resource를 client에게 제공하는 static resource web server의 역할 또한 수행한다.

5.4.2 Django Backend server

webpack compile된 vue앱을 제공하는 웹서버로서의 역할과, data base에 접근할 수 있는 API를 제공한다. 그 과정에서 JWT token을 검증하고, 권한에 따른 접근 제어를 수행한다.

5.4.3 web service 동작 시나리오

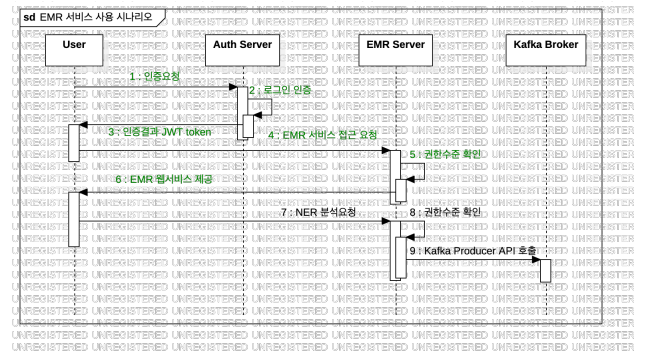


그림 8: Web Service 시퀀스 다이어그램

위의 시퀀스 다이어그램은, 웹서비스 전반에 대한 API 요청 시퀀스와, NER 모듈 호출을 예시로 Kafka 연동부를 표현하고 있다. 녹색으로 표시한 1-6번 시퀀스는 일반적인 웹 서비스의 접근 시퀀스이고, 7-9번 시퀀스는 EMR/NER 호출을 위한 추가적인 시퀀스를 나타낸 것이다.

일반적인 웹 서비스 시퀀스

지금까지 구현된 CHIS의 모든 웹 서비스는 일정수준 이상의 권한을 가진 회원에 한해 제공된다. 따라서, 특정 웹 서비스에 접근하기 위해서는 Auth Server에 인증요청 먼저 보내고, 토큰을 획득하여 서비스에 접근한다. 각 시퀀스에 대한 설명은 아래와 같다.

1. 다른 웹 서비스에 접근하기위해 로그인(인증요청)을 수행한다
2. 인증서버(Auth Server)는 로그인 요청을 처리하여 JWT토큰을 생성한다. 토큰은 RSA hash로 암호화된다.

- 로그인 요청을 보낸 client에게 JWT을 포함하여 응답한다.
- 특정 웹 서비스에 접근 시도한다
- JWT token을 RSA pub key로 복호화하여 권한 수준을 확인한다.
- client가 충분한 권한을 소유하고 있다면, 웹 서비스를 제공한다.

EMR에서의 추가적인 시퀀스

EMR서비스는 Kafka Producer로 동작하여, NER서비스가 분석가능한 EMR 문서가 생성되었음을 Kafka에게 알린다. 그 과정을 설명하면 아래와 같다.

- client는 웹서비스를 통해 EMR 서버에 NER 분석 요청을 보낸다.
- client가 갖고있던 JWT토큰을 확인하여 올바른 권한을 갖는지 확인한다.
- 충분한 권한을 갖고 있다면 Kafka Producer API를 통해 Kafka Broker에게 EMR텍스트를 전달한다.

제 6 장 시스템 검증

6.1 F1 score

F1 Score는 분류 모델에서 사용되는 머신러닝의 평가 지표로 데이터 불균형(Imbalanced data)에 대처하기 위해 개발되었다. 데이터 분류에 대해 정확도를 측정하는 평가 지표이며, 0 이상 1 미만의 값을 가진다. 해당 결과는 precision과 Recall를 기반으로 계산된다.

- Accuracy : 전체 샘플들 중 올바르게 예측한 샘플들의 비율
- Precision: 모델이 True라고 분류한 것 중 실제로 True인 것의 비율
- Recall: 실제 True인 것 중 모델이 True라고 예측한 것의 비율

위 개념을 이용하되 Precision과 Recall은 데이터 불균형의 영향을 받는다는 점을 고려, F1 Score는 아래와 같이 정의된다.

$$F1Score = \frac{2 * (recall * precision)}{(recall + precision)} \quad (2)$$

그림 9: F1 score의 정의

Epoch	Accuracy	Loss	F1 score	Precision	Recall
1	0.848	0.0277	0.631	0.711	0.623
2	0.898	0.0181	0.759	0.82	0.75
3	0.916	0.0147	0.802	0.85	0.794
4	0.928	0.0124	0.831	0.873	0.823
5	0.939	0.0103	0.861	0.896	0.854
6	0.945	0.0091	0.873	0.905	0.867
7	0.946	0.0092	0.873	0.902	0.868
8	0.955	0.0072	0.897	0.922	0.8931

그림 10: train data epoch별 model 변화

Epoch	Accuracy	Loss	F1 score	Precision	Recall
1	0.893	0.0193	0.752	0.827	0.735
2	0.904	0.0172	0.772	0.834	0.758
3	0.908	0.0170	0.786	0.831	0.779
4	0.912	0.0160	0.8	0.833	0.804
5	0.918	0.0162	0.812	0.853	0.81
6	0.922	0.0162	0.819	0.861	0.812
7	0.92	0.0162	0.822	0.851	0.825
8	0.92	0.0175	0.819	0.844	0.822

그림 11: test data epoch별 model 변화

6.2 NER 모델 대한 통계적 검증

F1 score는 NER_train_module.py가 학습을 완료하고 그 결과로 생성된 NER_model이라는 모델에 대한 통계적 유의미성을 명시한다. 최종 F1 score로 0.819를 얻었으며, 이는 동일 데이터셋을 다른 관련 연구 사례의 최종 F1 score인 0.871보다는 다소 미흡한 점수이다. 가장 큰 이유로는, 이번 프로젝트에서 최종적으로 확정 지은 모델은 DistilBert(모든 도메인에서 사전학습된 Bert의 경량화 모델)뿐 인 것에 비해, 관련 연구 사례에서 최종적으로 확정 지은 모델은 BioBert(의학 도메인에서 사전학습된 Bert의 하위 모델)에 BiLSTM(Bidirectional Long Short-Term Memory)과 CRF(conditional Random Field)를 적층시킨 하위 모델들의 합으로 구성되어 있기 때문이다. 이에 대한 개선 여지들은 “개선 방안”에서 설명하겠다.

6.3 Ner added EMR 검증

EMR서비스에서 NER AI module을 호출하는 과정에 누락없이 값이 전달되고, 하이라이팅이 이루어지는지 검증하도록 하겠다.

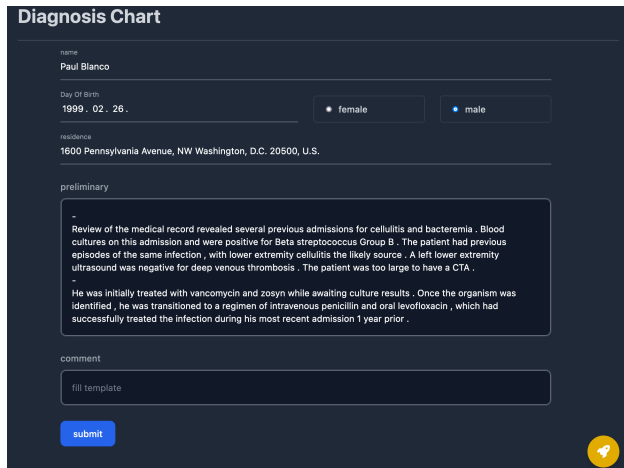


그림 12: EMR text 입력

위의 화면은 EMR서비스의 DIagnosis console이다. 의료진은 해당 화면에서 특정 환자에 대한 기본정보와, comment를 남길 수 있다. 우측 하단의 황색 버튼을 클릭하면, comment를 MQ로 전달하여 NER module을 통한 분석을 수행하고, 그 결과는 아래 화면과 같이 text-highlighting을 통해 확인할 수 있다.

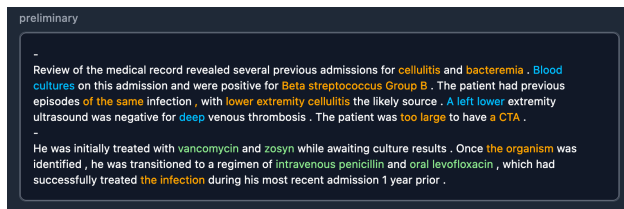


그림 13: NER을 통해 highlight된 EMR text

위의 결과를 통해 각 태그(황색: problem / 청색: test / 녹색: treatment)에 해당하는 highlight가 올바르게 이루어지는것을 알 수 있다. 앞서 입력시에 plain-text와 highlight된 text의 가독성을 비교해보면, 후자에서 원하는 정보를 더 빠르게 인지할 수 있을것으로 보인다.

제 7 장 결론

NER 모듈을 적용한 의료 정보 시스템(HIS)을 설계한다. EMR 서비스는 MSA 구조로 각 기능들을 구현하여 환자 예약부터 의사 처방 및 진단까지 제공한다. 또한

EMR 서비스는 EDA 구조를 차용하여 이벤트를 관리하는 웹 서비스를 구현한다. EMR 서비스에서 kafka api로 텍스트를 NER 모듈로 넘겨준다. 넘겨받은 텍스트를 분석하기 위한 의료 NER 모듈은 n2c2 NLP Research Data Sets을 이용하여 DistilBERT를 파인튜닝하여 개발한다. 모델 검증으로는 F1 score를 이용하였으며 해당 수치는 81.9% 이다.

AI를 이용하여 환자 데이터를 분석하고 유의미한 부분을 의료진에게 제공하는 것은 의료진들의 진단 정확성 향상을 야기하고 확정 진단 시간을 줄여줄 것이다. 이는 더 많은 환자들에게 진료 제공을 가능하게 하고 더 나아가 경제적인 이득도 기대할 수 있다. 실제로 이번 프로젝트를 진행하면서, plain text로 이루어진 EMR문서에 개체명들을 highlight하여, 가시성과 직관성을 유의미한 오차 범위 내에서 개선시켰다.

그럼에도 개선할 사항은 있다. 위 모델에 사용된 DistilBert은 general 도메인에 보편화 된 BERT모델을 성능 감소를 통해 경량화한 모델이다. Bert의 또다른 하위 모델로, BioBert라는 의학 도메인을 기반으로 사전학습된 모델이 있다. BioBert[6]는 의학 도메인 상에서 파인튜닝 되는 것을 주 목적으로 하는 만큼, EMR에 적용시 기존 모델보다 정확도 향상이 나타난다는 것이 관련 연구 결과에서 제시되고 있다. 또한, BERT만을 인공 신경망의 메인 모델로 두는 것이 아닌, BiLSTM이나 CRF 같은 다른 모델들 또한 인공 신경망에 추가한 구조를 가지게 할 수도 있다. 해당 구조를 차용하면 정확도가 더 향상된다는 것 또한 관련 연구 결과에서 제시되고 있다.

참고 문헌

- [1] Xin Yu, *BioBERT Based Named Entity Recognition in Electronic Medical Record*, 2022
- [2] 손병은, 의료인공지능 연구/개발 및 실용화를 위한 지능형 병원정보시스템 모델, 2019
- [3] Ashish, *Attention is All you Need*, 2017
- [4] Victor Sanh, *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, 2019
- [5] Ozlem Uzuner, *2010 i2b2VA challenge on concepts, assertions, and relations in clinical text*, 2011
- [6] Jinhyuk Lee, *BioBERT: a pre-trained biomedical language representation model for biomedical text mining*, 2019