딥러닝을 이용한 코로나 챗봇

이세훈*, 정지석O, 김영진*, 권형근*, 서희주*
OO인하공업전문대학 컴퓨터시스템과,
*인하공업전문대학 컴퓨터시스템과
e-mail: seihoon@inhatc.ac.kr*, wltjr137i@naver.comO, s564270@naver.com*,
ghg9612141@naver.com*, heejune25@naver.com*

COVID-19 Chat Bot by using Deep Learning

Se-Hoon Lee*, Ji-Seok JeongO, Young-Jin Kim*, Hyeon-guen Kwon*, Hee-Ju Seo*
ODept. of Computer System, Inha Technical College,
*Dept. of Computer System, Inha Technical College

● 요 약 ●

본 논문에서는 현재 이슈가 되고 있는 코로나에 대해서 사람들이 실생활에서 궁금해할 정보들을 Seq2seq 기술을 사용한 챗봇으로 정보를 제공한다.

키워드: 코로나19(COVID-19), Seq2seq(Sequence to sequence), 인코더/디코더(Encoder/Decoder)

I. Introduction

코로나가 발생 된 후, 2020년 3월 서울 구로구의 콜센터에서 발생한 코로나 집단 감염사태는 사람들이 밀집되어있고, 직업 특성상 비말이 튀어 감염 위험도가 높을 수밖에 없었다. 코로나의 감염이 갈수록 증가해가는 지금, 감염의 위험도를 낮추고 상담원의 피로도를 없애줄, 즉 일을 대신 해주는 프로그램이 개발되면 업무 부담이 줄어들지 않을까 고찰하였다. 상담원 업무와 가장 비슷한 프로그램이 챗봇 시스템이었고, 챗봇 시스템에 적용 가능한 딥러닝 모델을 조사해본 결과 Seq2seq 모델이 있었다.

본 논문의 1차 목적은 Seq2seq 아키텍처를 이용하여 상담원 업무와 감염 위험도를 덜어줄 코로나 상담 챗봇 시스템을 구현하는 것, 궁극적으로는 이 시스템을 사용하여 코로나 19의 정보를 쉽고 빠르게 찾고 이를 통하여 코로나 19의 전파를 줄이는 것이 궁극적인 목표이다.

II. System Architecture

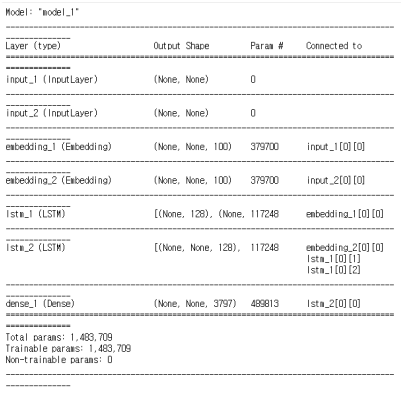
Fig1은 전체적인 시스템 구성도이다. Seq2seq 모델을 사용하여 학습시킨 후 파이썬의 Flask로 Rest API 서버를 구축하여 Spring과 연결하였다.

Fig. 1. System Architecture

III. Experiments

데이터 전처리는 Konlpy의 Komoran, Okt를 가지고 데이터를 전처리했다. 품사별로 문장을 나눈 뒤, 그 나눈 것들을 각각 숫자로 토큰화해주었다. 또한, 문장 안에 들어갈 단어의 개수를 100개로 설정해주었고, 그보다 짧을 경우, 0으로 Padding해 주었다. 그리고 단어사전에 포함되지 않는 단어가 있다면 3으로 표시해주었다. Decoder 입력, 출력은 각각 1,2로 태깅해주었다.

Fig. 2는 Keras를 이용한 Seq2seq 모델 구조 중 학습 모델의 구조를 나타낸 것이다. 입력문장을 input1, 목표문장을 input2 입력. Embedding Layer를 거쳐 LSTM으로 학습을 한다. 출력은 문장 단어 개수만큼 One-hot 인코딩하여 맞는 예측한 단어를 골라낸다.



Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, None)	0	
input_2 (InputLayer)	(None, None)	0	
embedding_1 (Embedding)	(None, None, 100)	379700	input_1[0][0]
embedding_2 (Embedding)	(None, None, 100)	379700	input_2[0][0]
lstm_1 (LSTM)	[(None, 128), (None,	117248	embedding_1[0][0]
lstm_2 (LSTM)	[(None, None, 128),	117248	embedding_2[0][0] lstm_1[0][1] lstm_1[0][2]
dense_1 (Dense)	(None, None, 3797)	489813	lstm_2[0][0]

Total params: 1,483,709
Trainable params: 1,483,709
Non-trainable params: 0

Fig. 2. Training Model

Fig. 3, 4는 예측 모델의 Encoder, Decoder의 구조를 나타낸 것이다. Encoder 모델은 입력문장의 Cell state를 예측, Decoder 모델은 목표문장을 Encoder의 Cell State를 이용해 예측한다.

```
Model: "model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, None)              0
_________________________________________________________________
embedding_1 (Embedding)      (None, None, 100)         379700
_________________________________________________________________
lstm_1 (LSTM)                [(None, 128), (None, 128) 117248
=================================================================
Total params: 496,948
Trainable params: 496,948
Non-trainable params: 0
_________________________________________________________________
```

Fig. 3. Predict Encoder Model

```
Model: "model_3"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            (None, None)         0
__________________________________________________________________________________________________
embedding_2 (Embedding)         (None, None, 100)    379700      input_2[0][0]
__________________________________________________________________________________________________
input_3 (InputLayer)            (None, 128)          0
__________________________________________________________________________________________________
input_4 (InputLayer)            (None, 128)          0
__________________________________________________________________________________________________
lstm_2 (LSTM)                   [(None, None, 128),  117248      embedding_2[0][0]
                                                                 input_3[0][0]
                                                                 input_4[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, None, 3797)   489813      lstm_2[0][0]
==================================================================================================
Total params: 986,761
Trainable params: 986,761
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fig. 4. Predict Decoder Model

Fig. 2 모델을 이용하여 각각의 Konlpy 형태소 분석기인 Komoran, Okt를 사용하여 학습시켰다. 아래의 Fig. 5는 Okt, Fig. 6은 Komoran을 가지고 학습시킨 그래프이다. Komoran과 Okt 둘 다 학습 정확도 0.99, 검증 정확도 0.85를 넘었고, 학습손실률, 검증 손실률 둘 다 0.05 1.5 이하였지만 Komoran이 검증 정확도 측면에서 0.2 정도 앞서 있는 것을 볼 수 있다.

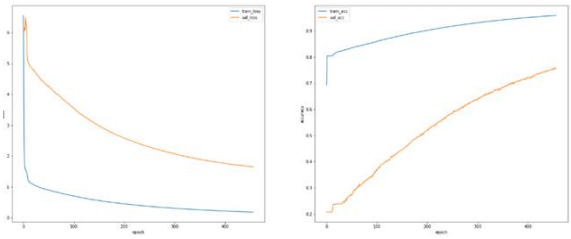


Fig. 5. Okt 학습 · 검증 정확도, 손실률

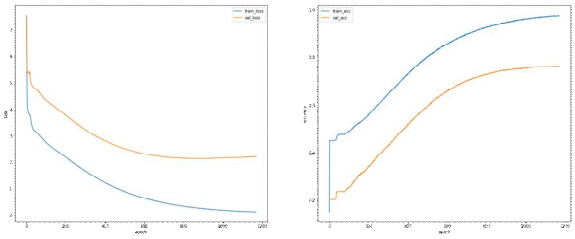


Fig. 6. Komoran 학습 · 검증 정확도, 손실률

IV. Conclusions

Konlpy의 형태소 분석기를 사용한 결과 Okt 보다 Komoran을 사용한 것이 더 나은 결과를 보여주었다. 이는 Komoran이 오탈자와 띄어쓰기에 대한 보정을 해주기에 이런 결과를 얻을 수 있는 것으로 판단된다. 학습 과정에서 발견한 문제는 한 문장 안에 들어가는 단어의 개수가 많으면 많을수록 학습 속도가 현저히 감소하는 것을 볼 수 있었다. 이는 한 문장 안에서 다음에 올 단어를 예측하는 형태이기 때문임이 확인되었다.

본 논문을 통하여 1장에서 제시한 1차 목적과 궁극적인 목적을 달성할 수 있다. 또한, 여기에서 STT와 TTS 기술을 활용하면 시, 청각 장애인들 또한 해당 챗봇으로 코로나에 대한 정보를 얻는 것을 기대할 수 있다.

REFERENCES

[1] A Study on Implementation of Patent Generation Chatbot Using Seq2Seq Model. Author: Jiho Lee, Seunghyun Bae, Kyunghoon Min, Deokgi Min (Konkuk University)

[2] https://github.com/deepseasw/seq2seq_chatbot github - Seq2Seq chatbot implementation