머신러닝

최신기술 특론

term project 결과보고서

- FF를 이용한 POS tagging

M2017517 조영훈

20153204 이가영

목차

1.역할 분담

1.1 팀원 역할 분담

1.2 Git Commit Graph

2.프로젝트 목표

2.1 프로젝트 개요

2.2 추진 배경 및 필요성

3.Train Method

3.1 개발 & 실험 환경

3.2 Input Data

3.3 Model

4.Train Result

4.1 Result

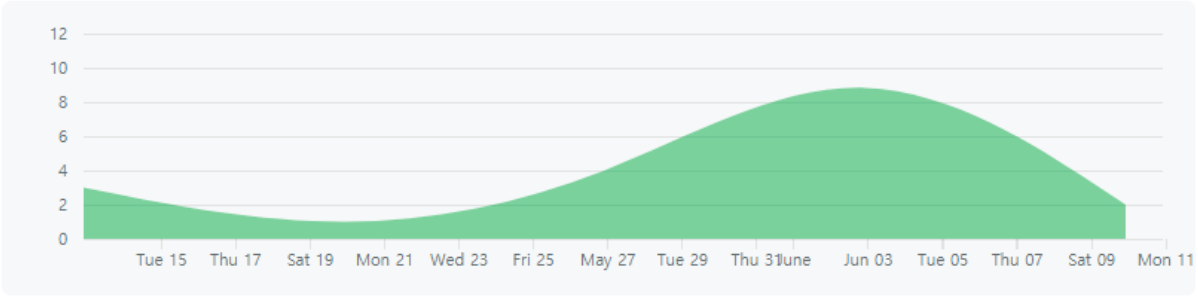
4.2 개선 사항

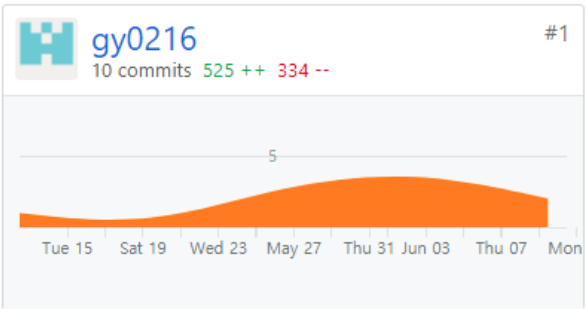
1. 역할 분담

1.1 팀원 역할 분담

학번	이름	역할
M2017517	조영훈	학습 모델 구축, Data 전처리, Test
20153204	이가영	데이터 수집, Data 전처리, Test

1.2 Git Commit Graph





2. 프로젝트 목표

2.1 프로젝트 개요

자연어는 어휘, 구문, 의미 수준에서 본질적으로 중의성을 포함하고 있다. 따라서 Raw corpus를 사용할 때 정확한 언어 정보에 대한 추출의 어려움이 발생하기 때문에 부가적인 언어 정보를 부착해야 한다. 이러한 언어의 중의성을 해소하기 위해서 부가적인 언어 정보를 부착하는 작업을 '태깅'이라고 하고, 형태소 분석의 단위로 쪼개어 각 형태소에 품사 정보를 '태깅'하는 작업을 'pos tagging'이라고 한다.

이와 같이 Pos tagging은 자연어 처리 분야에서 형태소 분석은 중요한 역할을 하고 있다. 그래서 우리는 현재 개발되어 있는 Hannanum, Kkma, Komoran, Mecab 등과 같은 pos tagging 라이브러리의 tagging format을 따르되, 통계적인 방식을 사용한 이것들과는 차별화 되도록 Deep Learning을 적용하여 pos tagging이 가능한 소프트웨어를 개발할 계획이다.

2.2 추진 배경 및 필요성

자연어 처리 분야의 Part-Of-Speech tagging (이하 pos tagging)은 Rule-Base 혹은 Statistical 방식을 이용한다. 보통 새로운 규칙에 대해 적용되지 않는 Rule-Base 방식보다 기존의 데이터를 통해서 규칙을 찾아내어 tagging을 하는 Statistical 방식을 많이 사용한다. 이 중에서도 Hidden Markov Model(이하 HMM)을 많이 사용하고 있다.

하지만, 과거에 비해 최근 PC 성능의 향상으로 Deep Learning의 성능 또한 함께 향상되었다. 따라서 기존의 pos tagging에 사용되는 통계학적 방식인 HMM을 대신하여 Deep Learning을 적용한다면 더 좋은 성능을 도출할 수 있을 것이라는 생각을 했다.

그에 따라 우리는 Deep Learning 기술을 적용하여 한국어 형태소 분석을 하여 tagging을 해주는 pos tagging 소프트웨어를 개발하기로 했다.

3. Train Method

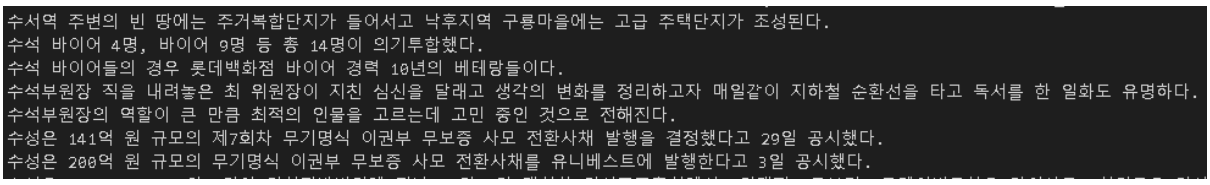
3.1 개발 & 실험 환경

TensorFlow Version : 1.8

OS : Ubuntu 16.04

Python Version : 3.5

3.2 Input Data



Train 시 사용한 Raw corpus의 Data format은 하나의 line에 한 문장으로 한다. 이 Data는 총

75MB, 561,125 line, 8,564,269 단어로 이루어져 있다.

```
수서역/N 주변의/N_j 빈/V_e 땅에는/N_j 주거복합단지가/N_j 들어서고/V_e 낙후지역/N 구룡마을에는/N_j 고급/N 주택단지가/N_j 조성된다/N_t_e ./q
수석/N 바이어/N 4명/N ,/q 바이어/N 9명/N 등/N 총/D 14명이/N_j 의기투합했다/N_t_f_e ./q
수석/N 바이어들의/N_s_j 경우/N 롯데백화점/N 바이어/N 경력/N 10년의/N_j 베테랑들이다/N_s_c_e ./q
수석부원장/N 직을/N_j 내려놓은/V_e 최/N 위원장이/N_j 지친/N 심신을/N_j 달래고/V_e 생각의/N_j 변화를/N_j 정리하고자/N_t_e 매일같이/N_j 지하철/N 순환선을/
수석부원장의/N 역할이/N_j 큰/V_e 만큼/U 최적의/N_j 인물을/N_j 고르는데/V_e 고민/N 중인/N_c_e 것으로/U_j 전해진다/V_e ./q
수성은/N_j 141억/N 원/N 규모의/N_j 제7회차/N 무기명식/N 이권부/N 무보증/N 사모/N 전환사채/N 발행을/N_j 결정했다고/N_t_f_e 29일/N 공시했다/N_t_f_e ./q
수성은/N_j 200억/N 원/N 규모의/N_j 무기명식/N 이권부/N 무보증/N 사모/N 전환사채를/N_j 유니베스트에/N_j 발행한다고/N_t_e 3일/N 공시했다/N_t_f_e ./q
```

Raw corpus를 konlp 라이브러리를 이용하여 pos tagging을 하여 Train의 input 데이터로 활용한다.

이때 분석된 형태소의 형태는 메모리의 문제로 각 품사가 '_'로 이어져 있다. 예를 들어 명사 + 조사 조합의 경우 세 가지 '로봇은', '수업은', '수성은' 을 분석해보면 다음과 같은 결과를 도출할 수 있다.

'로봇/명사 + 은/조사'

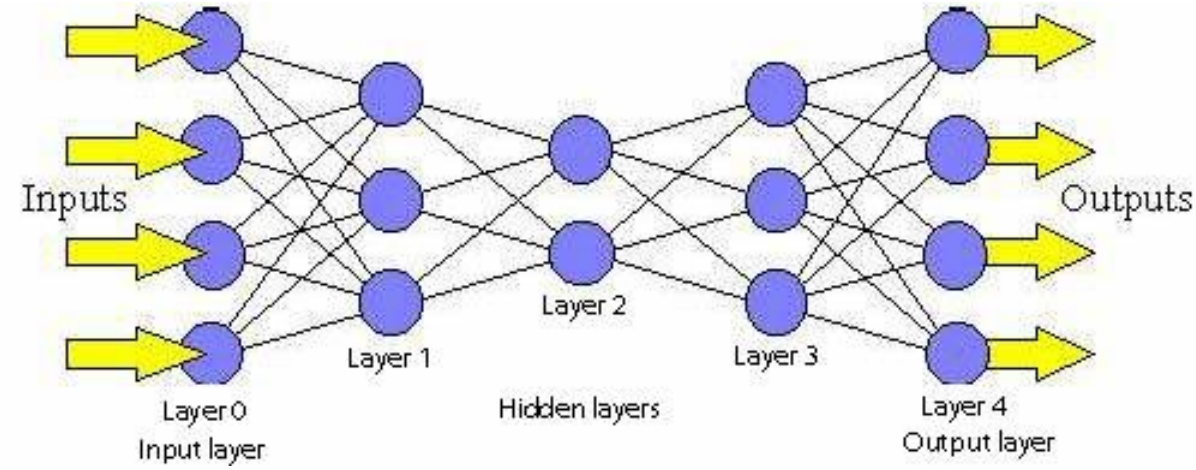
'수업/명사 + 은/조사'

'수성/명사 + 은/조사'

그러나 이와 같이 결과가 나올 경우 각각이 모두 다른 tag의 set으로 처리되어 결과적으로 tag의 set이 매우 많아지게 된다. 따라서 형태소를 제외하고 품사만을 '_'를 이용해 concatenation해서 표현하여 input data를 구성했다.

3.3 Model

이 프로젝트에 사용된 모델은 Depp Neural Network에서 가장 기초적인 모델이라고 할 수 있는 Feed Forward Neural Network(FF)를 사용했다. 다음은 모델의 도식화 그림이다.



Fully Connect 된 input layer, hidden layers, output layer로 모델이 구성이 되어 있다. 이 프로젝트에서는 Hidden layer의 수를 여러 번의 조정을 거쳐 가장 높은 결과를 도출할 수 있는 3개로 했다.

Input layer의 사이즈는 [feature vector의 x의 수 + 1 * embedding size]로 정의했고, hidden layer의 사이즈는 100으로 고정해서 학습을 진행했다. Output layer의 사이즈는 corpus의 tag의 수로 정의된다.

Input vector를 생성하는 방식은 random으로 생성된 수를 word vector로 가정하여 진행했고, input vector는 과거의 n개의 단어, label은 현재 단어의 품사이다.

Ex> 나는 집에 간다. N = 3

나는 0 0 '나는'의 tag

집에 나는 0 '집에'의 tag

간다 집에 나는 '간다'의 tag

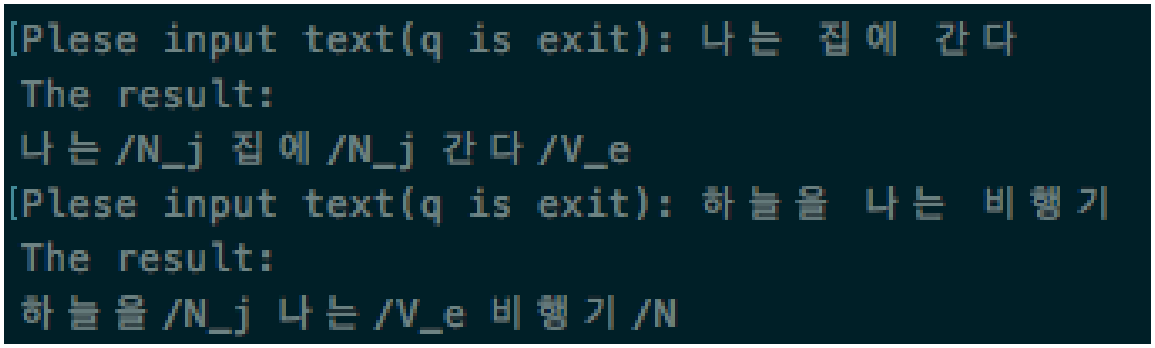
Result vector는 결과 tag에 대한 tag index를 가지고 있다. 각 layer간의 activation function은 ReLU를 이용했고, 결과를 위해 softmax classifier로 처리했다. Cost 함수로는 softmax cross entorpy를 사용했다.

4. Train Result

학습의 정확도는 결과적으로 94%가 나온 것을 확인할 수 있다.

학습의 결과를 확인하기 위해 가장 대표적인 형태소 분석 문장으로 사용이되는 "나는 집에 간다."와 "하늘을 나는 비행기" 두 문장을 실행했다.

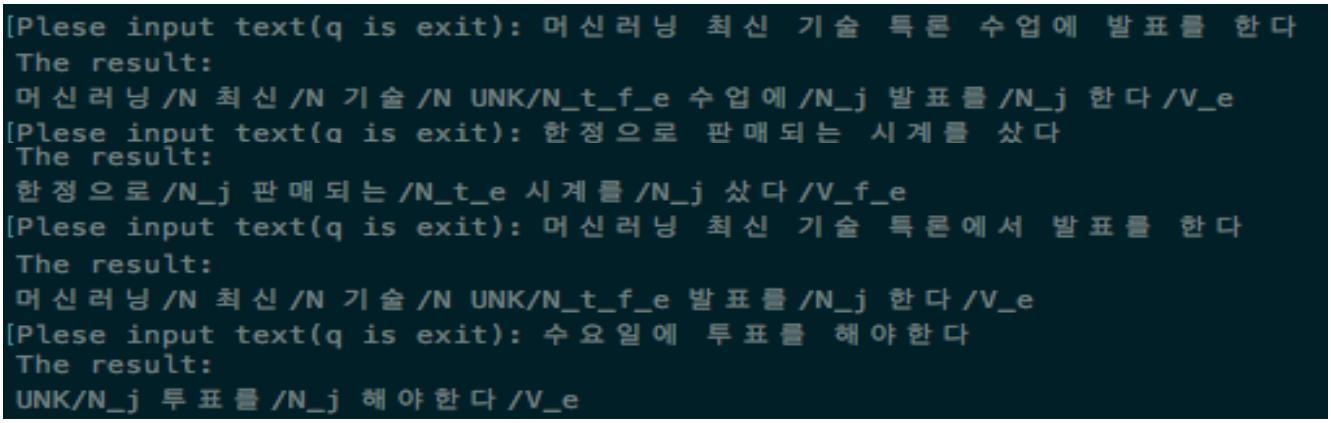
이 두 문장은 '나는'이라는 단어가 공통적으로 출현한다. 하지만, 이 단어는 각각의 문장에서 다른 뜻을 지니는 중의적 표현이다.



- 첫 번째 input text인 "나는 집에 간다."에서는 명사 + 조사인 N_j로 형태소가 분석이 되고 있다.
- 두 번째 input text인 "하늘을 나는 비행기"에서는 동사 + 어말어미인 V_e로 분석이 되고 있다.

다음과 같이 두 문장에서 '나는'이라는 단어가 다른 형태소로 분석이 되고 있는 것을 알수 있다.

하지만, 실험 결과 정확한 결과가 도출되지 않는 것도 확인할 수 있다.



이와 같이 UNK로 표현이 되고 있는 형태소는 학습 문서에 없는 단어이거나 vocab size에서 벗어난 경우로 확인된다. 이것은 학습에 필요한 Input data가 충분하지 않았다는 것이 가장 큰 요인으로 분석하고 있다.