

딥러닝 기반 MBTI 성격 유형 분류 연구

김정민, 박지민, 이로운, 조서원
이화여자대학교 컴퓨터공학과

jeongmin0723@ewhain.net, pjm1124kr@gmail.com,
leerhoan@ewhain.net, seowon000123@gmail.com

Deep Learning Based MBTI personality type test

Jeongmin Kim, Jimin Park, Rhoan Lee, Seowon Cho
Department of Computer Science, Ewha Womans Univ.

요약

성격 및 심리 검사는 개인이 자신의 행동이나 성격을 이해하고 이를 바탕으로 개인의 문제를 해결하기 위해 수행된다. 기존의 MBTI 검사는 객관적으로 파악한 검사라고 확신할 수 없기 때문에, 본 연구를 통해 사용자가 작성한 글을 바탕으로 성격 및 심리를 자동으로 분석해 줌으로써 편의성을 높이고, 객관적 성격 유형 검사의 정확도를 높이고자 한다. 이를 위해 총 33,000개의 데이터셋을 구축한 후, 성격 유형 분류 알고리즘으로는 LSTM을 사용하여 최종적으로 한 유형당 최고 70.32%의 정확도를 도출하였다. 이 과정에서 충분한 양의 데이터를 전처리 과정을 거쳐 LSTM 분류 알고리즘을 수행했을 때 의미 있는 학습 데이터양이 증가할수록 정확도가 높아짐을 확인하였다. 성격 유형 검사를 통해 평소 개인적인 감정이나 생각들을 작성한 SNS 계정과 연동시킨다면 편의성과 객관성이 보장된 성격유형을 분석할 수 있는 서비스를 기대해 볼 수 있다.

1. 서론

성격 및 심리 검사는 개인이 자신의 행동이나 성격을 이해하고 이를 바탕으로 개인의 문제를 해결하는 것에 도움이 된다. 최근 다수의 MZ세대 사이에서 성격 유형 지표인 Myers-Brigg Type Indicator(MBTI)가 유행하고 있다. 자신이 생각하는 자신의 모습과 MBTI 검사 결과로 나온 특징이 비슷하기 때문에 이러한 점에서 자신을 소개하는 수단으로 활용하고 있다. 상대방이 어떤 행동을 좋아하고 싫어하는지 미리 파악하는 데에도 큰 도움이 되어 MBTI를 이용한 소통이 활발해지고 있다.

현재 MBTI 검사는 질문에 대해 ‘그렇지 않다’부터 ‘그렇다’ 사이를 5단계 혹은 7단계로 나누어 답변을 선택하는 방법으로 이루어진다. 검사는 약 10분에서 12분 동안 진행되어

시간이 다소 오래 소요된다는 단점이 있다. 또한 사람들은 자신이 생각하는 자신의 성격을 바탕으로 답변을 선택하는데, 무의식적으로 본인이 되고 싶은 성격 유형에 맞춰서 답변을 선택하는 경우가 종종 있기 때문에 이 선택이 자신의 성격을 객관적으로 파악한 선택이라고 확신할 수는 없다. 이러한 문제들을 해결하기 위해 본 논문에서는 사용자가 작성한 글에 나타나는 성격 및 심리를 자동으로 분석하여 MBTI를 분류해주는 프로그램을 제안한다. 해당 프로그램에서는 높은 편의성과 객관성을 띠며, 궁극적으로는 검사의 정확도를 높이고자 한다.

2. 관련 연구

참고 논문 [1]에 의하면, Dropout 기법은 MLP나 CNN 모형들이 좀 더 나은 이진 분류

성능을 갖는 데 도움을 줄 수 있다는 가능성을 보여준다. Dropout 기법을 적용한 CNN 모형이 가장 좋은 F1 Score를 보였다는 점에서 추후 CNN 알고리즘을 이진 분류 문제에 활용해볼 가치가 있다는 것을 확인할 수 있다. 하지만 [2]는 CNN 모형의 경우 Long-term dependency를 구조적으로 반영하기 어렵다는 문제점이 있기 때문에, 텍스트를 망각 없이 처리하기 위해 LSTM 방식의 Deep Neural Network 구조를 제안한다. 실험 결과 LSTM 방법이 단순한 CNN 네트워크 구조에 비해 전반적인 성능 향상을 보였고, 특히 감정이 드러나 있는 경우 분류가 더 잘 됨을 확인하였다.

3. 모델 설계 및 구현

의미 있는 검사를 위하여 데이터 크롤링과 라벨링을 직접 하였고, 텍스트 전처리 과정을 거친 후 LSTM 모델에 적용해 분류해보았다.

3.1 데이터 크롤링 (Data Crawling)

크롤링을 위해 동적 페이지에 적용이 가능한 웹 브라우저 자동화 도구인 Selenium 모듈을 이용하였다. 데이터는 약 10만 6천 명의 회원수와 26만 개 이상의 게시물이 있는 네이버 MBTI 심리 카페에서 수집하였다. HTML 코드를 파이썬이 인식할 수 있는 객체 구조로 변환하는 Parsing을 담당하는 BeautifulSoup과 Pandas 라이브러리를 통해 필요한 데이터만을 담은 Data Frame을 선언하여 딥러닝에 적용했다. [표 1]과 같이 게시글 번호, 댓글 번호, 게시글 텍스트, 댓글 텍스트, 작성자 ID를 크롤링 하였다. MBTI 성격유형의 4가지 속성 (E 또는 I), (N 또는 S), (F 또는 T), (P 또는 J)을 이분법적으로 나누어 각각 0과 1로 라벨링 하였다.

데이터 출처	텍스트 내용	작성자 아이디	라벨링
{INFJ&ENFJ} 게시글 617518 번째 게시글 1 번째 댓글	제 남자친구는 enfi인데 어느 집단에서도 자기가 좋은 사람이 되려고 해요. 상대방에게 도움을 주면서 자신의 자존감을 올리려고 하고 계산적인것 같아요. 그래서 가끔 답답하긴 하지만 그래도 다정하고, 말도 잘하고 표현도 잘해줘서 좋아요. 어디 여행가면 계획도 짜주는 리더형이라서 더 매력있네요.	ENT50F50J 3w2	E: 0 N: 0 F: 0 J: 1
{INTJ&ENTJ} 게시글 570569 번째 게시글 3 번째 댓글	카톡으로는 잘 판단하기가 쉽지 않습니다. 친한 지인이더라도 연락만도는 중요하게 생각하지 않아요. 공부를 할 때도 여행을 갈 때도 미리 계획을 세워야 마음이 편해요.	아인슈타인 ENTJ	E: 0 N: 0 T: 1 J: 1

[표 1] 데이터 라벨링 예시

3.2 텍스트 전처리 (Text Processing)

MBTI 분류 알고리즘의 높은 정확도를 위하여 전처리 과정을 수행하였다. 입력 텍스트 중 개인의 성격 또는 심리와 무관한 내용은 삭제하였고, 뜻이 명확하지 않은 것 또한 삭제하였다. 텍스트 내에 부가 설명이 필요한 경우 텍스트를 가공하여 의미를 명확하게 해주었다. 또한, 구축한 데이터셋이 한국어기 때문에 텍스트 전처리(Text Processing) 과정 중에서도 한국어 전처리 패키지 도구(Text Preprocessing Tools for Korean Text)를 사용하였다.

텍스트 전처리 과정을 위해 먼저 데이터열에서 중복인 내용이 있다면 중복을 제거하였다. 한글과 공백을 제외하고 모두 제거하는 정규 표현식을 수행하여 기존의 띄어쓰기는 유지되면서 온점과 같은 구두점 등을 제거하고, 공백만 있거나 빈 값을 가진 행이 있다면 Null 값으로 변경하고 제거하였다. 토큰화 과정에서 형태소 분석기인 KoNLPy의 Okt를 사용하여 토큰화를 하면서 불용어(stop words)를 제거하였다. 형태소 분석기 옵션을 stem으로 사용하여 각 단어에서 어간을 추출하여 일정 수준의 정규화를 수행하였다. 기계가 텍스트를 숫자로 처리할 수 있도록 데이터에 정수 인코딩을 수행하였다. 훈련 데이터에 대해서 단어 집합을 만들어 각 단어에 고유한 정수를 만들어 단어 집합이 생성되는 동시에 각 단어에 고유한 정수를 부여하였다. 전체 훈련 데이터에서 등장 빈도수가 높은 순서대로 정수가 부여되었기 때문에, 높은 정수가 부여된 단어들은 등장 빈도수가 매우 낮음을 의미한다. 등장 빈도가 threshold 값인 3회 미만의 단어들은 자연어 처리에서 배제한 후, 단어 집합을 Keras Tokenizer의 인자로 넘겨주고 텍스트 안의 단어들을 정수 시퀀스로 변환하였다.

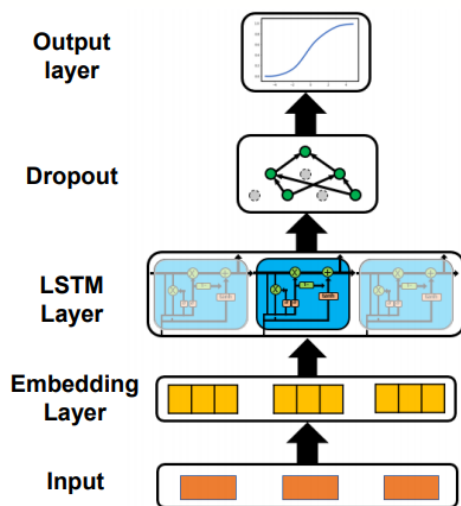
3.3 데이터 라벨링 (Data Labeling)

MBTI 분류 알고리즘의 정확도를 위해서는 보다 객관적인 라벨링 기준이 필요하다고 판단하여 3가지 기준에 따라 데이터를 객관적으로 라벨링 하였다.

- (1) 작성자 ID 기준으로 라벨링
- (2) 텍스트가 묘사하는 기준으로 라벨링
- (3) 1개의 텍스트 안에 여러 MBTI 특성이 드러나는 경우 분리하여 라벨링

3.4 LSTM 모델

본 연구에서는 대화 내 감정 분류를 하는 데에 있어 직간접적으로 드러나는 감정을 자동으로 학습하기에 효과적인 LSTM 방식의 Deep Neural Network 구조를 사용한다. LSTM 모델은 텍스트 시퀀스를 처리할 때 오랜 시간 동안 정보를 기억할 수 있기 때문에, 각 텍스트 데이터를 분류하는 작업을 하기에 적절하다고 판단하였다.



[그림 1] LSTM 모델 구성도

LSTM 모델은 파이썬의 Keras 패키지를 이용하여 구축하였다. [그림 1]과 같이 LSTM은 3개의 gate와 2개의 state로 이루어져 있고, 이러한 구조가 사슬과 같이 연결되어 동작한다. LSTM 구조 뒤에 출력층을 두었다.

본 연구는 MBTI의 예측을 4가지 특성인 (E 또는 I), (N 또는 S), (F 또는 T), (P 또는 J)에 대해 각각 이분법적으로 분석한다. 따라서 출력층으로 결과값을 전달할 때는 Sigmoid 함수를 사용하였다. 이 함수는 0부터 1 사이의 값을 출력층에 전달하는데, 출력층 결과값이 1에 가까울수록 I, S, T, J의 특징을 갖는 문장이며, 0에 가까울수록 E, N, F, P의 특징을 갖는 문장이라고 예측한다.

손실함수는 모델이 학습하는 과정에서 원래의 값과 예측한 값의 차이를 나타낸다. 이 함수는 모델의 오류를 줄이기 위해 사용된다. 이 논문의 프로그램에서는 0 또는 1의 2 가지 중 하나를 선택하는 작업을 하므로, 손실함수로 Binary cross entropy를 적용했다. Binary cross entropy 함수는 두 개의 class 중 하나를 예측하는 작업에 대한 손실 함수이므로 이분법적 분류에 적합한 손실 함수이다.

모델에서는 과적합을 막기 위해 여러 방법을 사용하였다. 먼저, weight regularization으로 L2 regularization을 적용했다. 그리고 학습 과정에서 dropout을 적용한다. Dropout은 랜덤으로 신경망의 일부를 삭제하는 것인데, 이를 통해 특정 뉴런이나 조합에 과하게 의존하는 것을 막고, epoch마다 랜덤하게 신경망을 삭제해주기 때문에 서로 다른 신경망들을 조합해서 사용하는 것과 같은 효과를 내면서 과적합을 방지한다. 또한, 검증 데이터의 정확도가 증가하지 않는 것이 4회가 되는 것은 과적합 징후이기에 그릴 경우 학습을 조기 종료(Early Stopping)하도록 하였다.

4. 연구 과정

연구를 위하여 train 데이터 중 일부를 validation 데이터로 두고, 각 epoch마다 validation 데이터를 테스트한다. Validation 데이터를 통해 과적합의 징후를 확인할 수 있다.

4.1 L2 Regularization 상수가 모델의 학습 과정에 미치는 영향

$$\text{Cost} = \sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} W_j)^2 + \lambda \sum_{j=0}^M W_j^2 \quad (1)$$

L2 Regularization은 위의 식 (1)에서 볼 수 있듯이 weight의 제곱값을 모두 더하고 L2 상수를 곱한 값이 손실에 더해진다.

λ	0.0005	0.0015	0.0045	0.0135	0.0405
a)	0.6594	0.6569	0.6573	0.4996	0.4996
b)					
c)					
a): 테스트 정확도, b): train, validation의 loss 그래프, c): train, validation의 정확도 그래프					

[그림 2] L2 상수 λ 에 따른 정확도 그래프

L2 상수인 λ 를 0.0005부터 3배씩 늘리며 L2 상수가 학습에 어떤 영향을 주는지 확인하는 연구를 했다. [그림 2]는 L2 상수에 따른 a)테스트 정확도, b)validation과 train의 loss 그래프, c)validation과 train의 정확도 그래프를 나타내준다.

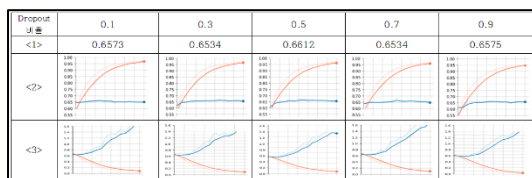
a) 0.0005부터 0.0045일 때까지의 테스트 정확도는 거의 동일하다. 하지만 0.0135일 때부터는 테스트 정확도가 급격히 줄어들었다.

b) validation과 train의 loss 그래프를 보면 L2 상수가 커짐에 따라 두 폭이 줄어드는 것을 확인할 수 있다. 하지만 L2 상수가 0.0135일 때부터는 validation과 train의 loss가 급격히 커졌다.

c) validation과 train의 정확도 그래프를 보면, 0.0005부터 0.0045일 때까지는 L2 상수가 달라져도 validation 정확도 그래프가 거의 동일하다. 그리고 0.0135일 때와 0.0405일 때는 epoch에 따라 validation과 train의 정확도가 개선되지 않는 것을 볼 수 있다.

위 세 가지 결과를 확인했을 때, L2 상수가 예측에는 딱히 영향을 미치지 않는다고 볼 수 있다. 또한 L2 상수가 어느 정도 이상으로 커지면 weight 값을 줄이는 것에만 집중되어서 학습이 제대로 이뤄지지 않는다.

4.2 Dropout 값이 모델의 학습 과정에 미치는 영향



[그림 3] Dropout 값에 따른 변화

Dropout 값을 0.1부터 0.9까지 0.2씩 늘려가며 비교해 보았다. Dropout 값에 따른 <1> 테스트 정확도, <2> validation과 train의 정확도 그래프, <3> validation과 train의 loss 그래프는 [그림 3]과 같다.

<1> 테스트 정확도는 5가지 경우에서 거의 동일하다.

<2> dropout의 비율을 달리해도 결국 마지막 epoch의 train 정확도는 0.85 부근으로 거의 동일하고, validation 같은 경우도 어느 시점부터 0.65 부근을 유지하게 된다.

<3> validation과 train의 loss 그래프를 보면 두 그래프의 폭에 큰 변화가 없다.

위 세 가지 결과를 확인했을 때, dropout 값을 달리 하는 것이 과적합을 줄이는 것과 예측에 영향을 주지는 않는 것으로 보인다.

5. 연구 결과

	라벨링(1)	라벨링(2)	라벨링(3)
E/I	25862/36519	3268/4174	13107/18964
N/S	35386/26995	12377/9766	16184/16352
F/T	32122/30259	3790/3778	13396/19088
P/J	33093/29288	3920/3636	19065/13385

[표 2] 시기별 데이터 양의 변화

[표2]는 연구 진행 시기별 각 지표의 데이터 양 변화를 나타낸 표이다. 라벨링 기준(1)에 따라 알고리즘을 실행시켰을 때 총 9,700개의 데이터로 도출한 전체 정확도는 13.79%였다. 이후 라벨링 기준 (2), (3)을 누적해가며 총 26,000개의 학습 데이터와 7,000개의 테스트 데이터를 구축하였다. 데이터의 노이즈들을 정제하고 데이터의 개수도 꾸준히 증가시킨 결과, 해당 데이터로 도출한 유형별 정확도로 E/I 유형 65.33%, N/S 유형 66.88%, F/T 유형 66.07%, P/J 유형은 70.32%를 얻었고, 최종적인 전체 정확도는 20.29%임을 확인하였다.

	접근 방법	분류 모델	분류 언어	데이터의 양	전체 정확도
[3]	binary	LSTM	영어	430,000	21%
[4]	binary	LSTM	영어	422,845	38%
[5]	binary	Linear/Logistic Regression	영어	17,640,062	45%
[6]	multiclass	LR	영어	8676 (frequency of data set)	66.59%

[표 3] 기존 MBTI 분류 연구 분석

[표 3]은 기존 MBTI 분류 연구와 본 연구 결과를 비교한 표이다. 비교 지표로는 접근 방법, 분류 모델, 분류하는 언어, 데이터의 양이 있다. 4가지 비교 논문은 모두 영어 텍스트 데이터를 사용하였고 [3], [4], [5]는 본

연구의 데이터양의 13배~532배 데이터를 사용하여 각 21%, 38%, 45%의 정확도를 얻었다. 이중 논문[5]은 binary 접근 방식으로 여러 데이터셋에 적용해본 결과 약 45% 정도의 정확도를 도출하였다. 또한, 논문 [6]에서는 multiclass 접근 방법으로 정확도를 66.59%까지 끌어올렸다. [표3]의 선행 연구와의 비교를 통하여, 본 연구의 정확도는 적은 데이터양임에도 불구하고 유의미한 수치임을 알 수 있다.

6. 결론

본 연구의 딥러닝 기반 텍스트를 분석하여 성격유형을 분류하는 프로그램은 300자 이상의 개인의 특성 및 감정이 잘 드러나는 텍스트만으로 객관적인 성격유형 파악이 가능하다. MBTI 성격유형검사의 4가지 항목 E/I, N/S, T/F, P/J 당 각각 65.33%, 66.88%, 66.07%, 70.32%의 정확도를 산출하였다. 본 연구를 통해 자신이 작성한 글을 바탕으로 자신의 객관적인 성격유형을 분석해주는 높은 정확도의 프로그램을 제작 가능하다는 결론을 얻었다. 또한, 본 연구는 한국어 텍스트로 딥러닝 기술을 적용한 최초의 MBTI 성격유형 분석 프로그램이라는 점에서 매우 의미 있다. 본 연구를 더 확장 시킨다면 각 성격유형에 따른 적절한 교수 방법을 개발하거나 적합한 직업 선택을 위한 진로 지도 및 상담에서의 사용을 기대할 수 있다. 또한, 데이터양을 증강하여 단어들 간의 의미적 분류(벡터 관계)를 이용한 후속 연구를 통해 더 발전시킬 계획이다.

참고문헌

- [1] 김기태 외 “이진 분류문제에서의 딥러닝 알고리즘의 활용 가능성 평가”, 지능정보연구 = Journal of intelligence and information systems v.23 no.1 , 2017년, pp.95 - 108
- [2] 신동원 외 "CNN-LSTM을 이용한 대화 문맥 반영과 감정 분류", 제28회 한글 및 한국어 정보처리 학술대회 논문집 , 2016년
- [3] R. Hernandez and I. S. Knight,

"Predicting Myers-Briggs type indicator with text classification", Proc. 31st Conf. Neural Inf. Process. Syst., pp. 4-9, Dec. 2017, Jul. 15, 2021

[4] B. Cui and C. Qi, Survey Analysis of Machine Learning Methods for Natural Language Processing for MBTI, Jun. 2017

[5] M. Gjurković, M. Karan, I. Vukojević, M. Bošnjak and J. Snajder, "PANDORA talks: Personality and demographics on Reddit", Proc. 9th Int. Workshop Natural Lang. Process. Social Media, pp. 1-15, 2021.

[6] S. Chaudhary¹, R. Singh, S. T. Hasan and M. I. Kaur, "A comparative study of different classifiers for Myers-Brigg personality prediction model", Int. Res. J. Eng. Technol., vol. 5, no. 5, pp. 1410-1413, 2018.