텍스트 스타일로 MBTI 성격 예측

Predicting MBTI Personality through Text Styles

CUAI 4기 NLP팀
유승욱 중앙대학교 소프트웨어학부
김상렬 중앙대학교 컴퓨터공학부
김중훈 중앙대학교 응용통계학부
박경빈 중앙대학교 소프트웨어학부

[요약] 성격 유형 지표로서 사람의 특성을 나타내는 MBTI로 텍스트 분류를 진행했다. 국내 MBTI 커뮤니티에서 크롤링으로 데이터를 수집하고, XGBoost 등의 머신러닝 모델과 LSTM 등의 딥러닝 모델을 활용해 텍스트 분류 모델을 구현했다. 높은 정확도를 보인 텍스트 분류 모델을 웹에 적용하여 텍스트로 MBTI 예측 결과를 실시간으로 확인할 수 있다.

1. 서 론

MBTI 성격 유형 검사는 사람의 성격 유형을 네가지 지표로 구분한다. 외향형(E)과 내향형(I), 직관형(N)과 감각형(S), 감정형(F)과 사고형(T) 그리고 판단형(J)과 인식형(P)까지 이들을 조합하면 총 16가지다. 예를 들어 ENFJ는 정의로운 사회운동가로서 충만한 열정을 지닌 타고난 리더형, ISTP는 만능 재주꾼으로서 왕성한 호기심을 가진 재주꾼형 등으로 16Personalities[1]에서 지표 조합에 따른 사람의 특성(이하 MBTI)을 소개하고 있다.

최근 온라인 커뮤니티가 발달하며 서로의 MBTI가 무엇인지 소통하는 사람들이 늘어났다. 내가 어떤 사람인지 알아가려는, 또 상대방을 이해하고 싶다는 욕구가 반영되며 누리꾼들의 입소문에 오르내린 것이다. 이런 상황 속 작성자에 대한 정보 없이 글만으로 상대 MBTI를 파악한다면 작성된 텍스트를 글쓴이의 시선으로 바라볼 기회가 생길 것이다. 상대방의 관점에서 텍스트 맥락을 이해하며 서로의 글에 더욱 공감하는 상황은 곧 원활한 의사소통으로 이어진다.

또한 작성한 글로 파악된 MBTI가 실제와 다르다면 글쓴이의 평소 작문 성향을 파악해볼 수 있다. 본인의 성격이 반영되지 않는 글을 작성한다는 사실과, 실제 글에 반영된 성격이 무엇이었는지를 깨우치게 함으로써 자신의 텍스트 스타일에 대한 성찰이 가능해진다.

본 연구에서는 국내 MBTI 커뮤니티 중 거대 규모인 MBTI & HEALTH 심리 카페[2]의 게시글을 크롤링하여 성격 유형에 맞게 분류하고자 한다. 10년 넘게 유지되었으며 2020년에는 대표 인기카페로 선정된 커뮤니티의 글을 각종 머신러닝 및 딥러닝 NLP 모델에 학습하고, 새로 제시된 텍스트 데이터의 MBTI 예측 정확도를 검증한다.

2. 본 론

1) 활용데이터 정의

본 연구에서는 MBTI & HEALTH 심리 카페의 게시판 11개에서 약 44,000개 정도의 본문 텍스트 및 작성자 닉네임을 수집하였으며. 크롤링 과정에서 Selenium과 BeautifulSoup을 활용하였다. 데이터를 선정함에 있어 다음의 고려사항을 반영하였다.

첫째, 서로의 MBTI를 염두하며 대화가 진행되어온 커뮤니티 게시판을 선정하였다. 작성자의 MBTI를 예측하는 것이 목적이니만큼 얻을 데이터 역시 작성자의 MBTI가 명시된 글을 선택했다.

둘째, 얻은 텍스트가 모델링에 유의미하게 적용될지 고려하였다. 짧지 않은 기간동안 글이 쌓여온 디시인사이드 MBTI 갤러리를 고려하였으나 작성된 텍스트의 퀄리티가 일정 수준만큼 보장되지 않는 점으로 인해 활용데이터에서 배제하였다.

마지막으로, 충분한 기간 동안 축적된 데이터로 가능한 많은 양을 모으려했다. 외국과는 달리 아직 국내에 MBTI 라벨링 데이터셋이 없어 크롤링을 통해 데이터를 직접 얻어야했는데, MBTI를 밝힌 유명인의 SNS 텍스트 수집을 고려하였으나 일부 텍스트를 특정 MBTI로 라벨링하기 애매하다는 점으로 인해 활용데이터에서 배제하였다.

2) 데이터 라벨링 및 전처리

축적한 텍스트를 16가지나 되는 MBTI에 맞게 분류하려면 작성자의 MBTI 특성이 글에 고스란히 담겨있다는 전제가 필요하다.

<표1> 작성자 MBTI에 따른 텍스트 예시

본문	작성자 MBTI
가식이라도 하면 좋을까요?	ISFP

그러나 실제 게시판 글을 살펴본 결과 모든 글에 작성자의 MBTI가 그대로 반영되지 않았으리라 판단했으며, 이에 따라 16가지나 되는 MBTI로 텍스트 분류는 어려울 것으로 생각하였다.

텍스트 분류 기준을 달리해 16가지 MBTI의 개별적인 특성을 고려하기보다는 이들의 세세한 분류를 가능하게 해준 네가지 지표, 외향형(E)과 내향형(I), 감각형(S)과 직관형(N), 사고형(T)과 감정형(F) 그리고 판단형(J)과 인식형(P)의 이진분류를 통해 MBTI를 예측하기로 결정하였다.

또한 정규표현식과 텍스트 전처리, 정규화를 통해 데이터를 정제했으며 '엔프제', '잇팁' 등 MBTI를 지칭하는 표현이 한글로 작성되어 있는 경우 'ENFJ', 'INTP' 등 원래의 영어 MBTI 명칭으로 바꾸는 작업을 거쳤다.

3) 머신러닝 모델

본격적인 머신러닝 모델 사용 전 전처리 및 벡터화 방법에 대해 소개하겠다.

텍스트 데이터의 특징을 추출하고자 TF-IDF를 사용하여 단순 Count Vectorizer가 가진 문제점을 보완하였다. 특정 단어가 한 데이터 안에서 등장하는 횟수를 의미하는 TF와, 특정 단어가 다른 데이터에 등장하지 않는 지표를 나타내는 IDF를 사용해 조사나 지시대명사처럼 중요도가 낮지만 자주 등장하는 단어를 효과적으로 배제하였다. Okt 라이브러리로 단어를 형태소 분리한 상태에서 TF-IDF를 사용하여 더욱 정교한 텍스트 표현을 가능하게 했다.

Random Forest[3]는 전체 데이터에서 Bagging으로 샘플링해 여러 개의 결정 트리 분류기가 개별적으로 학습을 수행한 뒤, 최종적으로 모든 분류기가 Voting을 통해 예측 결정을 하는 모델이다. 대중적인 머신러닝 모델로서 자주 사용되며 앙상블 알고리즘 중 비교적 빠른 수행 속도와 다양한 영역에서의 높은 예측 성능을 이유로 해당 모델을 선택하였다.

XGBoost[4]는 Random Forest의 Bagging과 동일한 원리를 사용하나 Boosting 방식을 적용해 오답에 가중치를 부여하는 방향으로 예측 방향을 결정을 하는 모델이다. 기존 GBM(Gradient Boosting Model)의 느린 수행 시간 및 Regularization 부재 문제를 해결하며, CPU 환경에서의 병렬학습을 통해 빠른 학습이 가능한 이유로 해당 모델을 선택하였다.

SVM[5]은 데이터를 분류하기 위해 분류될 클래스 사이의 Margin을 최대화하는 초평면을 찾는 것으로, 선형 분류기로 이진 분류 데이터를 학습할 경우 0과 1 두개의 클래스를 분할하는 직선을 찾는 모델이다. 직선을 통한 구분이 불가능할 때 비선형 분류기를 사용하는데, 결과적으로 비선형 SVM을 사용하는 것이 성능 평가에서 더 나은 결과를 보여주었다.

4) 딥러닝 모델

LSTM[6]은 forget gate를 사용하여 기존 Simple RNN이 가졌던 장기기억 손실 문제를 극복한 모델이다. Bi-LSTM[7]은 Vanilla LSTM과는 달리 양방향 모델로 정방향과 역방향 두가지 모두에서 텍스트를 파악한다. Simple RNN과 GRU 등의 RNN 기반 모델 역시 모델을 사용해 보았으나 LSTM과 Bi-LSTM만큼 기대한 성능을 확인할 수 없었다.

GPT2[8]는 다양한 영역의 텍스트를 활용해 사전 학습이 진행됨으로써 모델이 기존보다 다양한 문맥과 영역의 글을 이해할 수 있다는 것이 특징이다. 기존 버전인 GPT1을 기반으로 모델 구조를 더 효율적인 방향으로 변형했다는 점과, 더 많은 사전학습 데이터를 이용하여 성능을 발전시켜 왔다는 점에서 해당 모델을 선택하였다.

3. 결 론

1) 머신러닝 모델 구현 결과

Random Forest에서의 성능은 아래와 같다.

<표2> Random Forest 성능평가

	E or I	N or S	F or T	J or P
accuracy	0.68	**0.77**	0.67	0.73
f1 score	0.34	**0.86**	0.52	0.43

XGBoost에서의 성능은 아래와 같다.

<표3> XGBoost 성능평가

	E or I	N or S	F or T	J or P
accuracy	0.62	**0.75**	0.55	0.67
f1 score	0.17	**0.85**	0.65	0.8

선형 SVM에서의 성능은 아래와 같다.

<표4> 선형 SVM 성능평가

	E or I	N or S	F or T	J or P
accuracy	0.67	**0.76**	0.64	0.71
f1 score	0.29	**0.86**	0.69	0.36

비선형 SVM에서의 성능은 아래와 같다.

<표5> 비선형 SVM 성능평가

	E or I	N or S	F or T	J or P
accuracy	0.68	**0.77**	0.66	0.72
f1 score	0.34	**0.86**	0.71	0.40

네가지 머신러닝 모델 모두에서 직관형(N)과 감각형(S)을 가장 잘 분류해냈으며, 이를 제외한 각 모델의 세가지 이진분류에서는 모두 비슷한 수준의 성능을 보였다. 또한 선형 SVM보다는 비선형 SVM에서 근소한 차이로 나은 결과를 확인했다.

2) 딥러닝 모델 구현 결과

LSTM에서의 성능은 아래와 같다.

<표6> LSTM 성능평가

	E or I	N or S	F or T	J or P
accuracy	0.63	**0.75**	0.66	0.69
f1 score	**0.62**	0.43	**0.62**	**0.62**

Bi-LSTM에서의 성능은 아래와 같다.

<표7> Bi-LSTM 성능평가

	E or I	N or S	F or T	J or P
accuracy	0.61	**0.76**	0.62	0.69
f1 score	0.59	0.52	**0.61**	0.59

GPT2에서의 성능은 아래와 같다.

<표8> GPT2 성능평가

	E or I	N or S	F or T	J or P
accuracy	0.67	**0.69**	0.65	0.58
f1 score	0.78	**0.86**	0.66	0.81

세가지 딥러닝 모델 모두에서도 마찬가지로 직관형(N)과 감각형(S)을 비교적 잘 분류해냈으며, GPT2에 비해 LSTM 계열에서 accuracy와 f1 score가 조금 더 높게 관측되었다.

3) 결론 및 보완

전반적으로 딥러닝 모델들보다 머신러닝 모델들의 accuracy와 f1 score가 높은 편으로 관측되었는데, 이러한 점에 있어 데이터 전체 개수가 영향을 주었으리라 생각하였다. 학습한 딥러닝 모델 깊이가 상당한 만큼 더욱 많은 텍스트 레이블 데이터가 있었다면 딥러닝 모델 성능 역시 머신러닝 모델보다 더욱 뛰어난 수준을 보였을 것으로 추측했다.

이처럼 작문 성향을 MBTI에 맞게 분류하는 모델을 만들어내자는 소기의 목적은 달성하였으나, 같은 MBTI를 가졌다고 한들 사람마다 글 스타일의 차이가 있을 수 있다. 또한 카페 게시판 특성상 다른 사람들에게 질문하는 글, 특정 주제에 대해 토론하는 글 등 본인의 텍스트 스타일이 반영되지 않았을 글들이 학습 데이터에 그대로 반영되었다. 카페 게시판의 글이 어떠한 형식으로 작성되는지 자세히 살펴본 후 전처리 기준을 기존보다 엄격하게 세워 데이터를 정제해간다면 전체적으로 더 높은 수준의 모델 성능을 기대할 수 있을 것으로 생각된다.

마지막으로 영어로 된 MBTI 텍스트 분류 데이터셋은 온라인상에 다수 존재하지만, 한글로 된 MBTI 텍스트 분류 데이터셋의 부재로 활용데이터를 고르는 단계부터 난항을 겪었다. 사람들이 MBTI에 더욱 많은 관심을 가져 MBTI 성격 특성이 담긴 글이 주기적으로 업데이트되는 커뮤니티가 앞으로 더욱 많이 형성된다면, 텍스트 스타일로 MBTI 성격를 예측함에 있어 좋은 데이터셋으로 활용될 것이다.

참고 문헌

[1] 16Personalities, https://www.16personalities.com/ko

[2] MBTI & HEALTH 심리 카페, https://cafe.naver.com/mbticafe

[3] Breiman, L. "Random Forests" Machine Learning 45, 2001

[4] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System" In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016

[5] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf. "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, July-Aug. 1998

[6] Hochreiter, S., & Schmidhuber, J. "Long short-term memory" In Proceedings of the Neural Computation, 1997

[7] Chenbin Li, Guohua Zhan, Zhihua Li. "News text classification based on improved Bi-LSTM-CNN" In Proceedings of 9th International Conference on Information Technology in Medicine and Education (ITME), 2018

[8] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. "Language models are unsupervised multitask learners", OpenAIblog, 2019