MBTI Personality Type Based on Social Media Record

Xinyi Cai
2018533085
`caixy@`

Beiyuan Yang
39132991
`yangby@`

Wenhui Qiao
57425238
`qiaowh@` *

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

2 Related Works

3 Feature Extraction

3.1 Dataset

We accquired the dataset from kaggle. It contains record of social media of 8675 users, each of the data contains 50 posts on social media, as well as the type label for those 8675 users. The general distribution over these types is summarized in the graph.

3.2 Word2vec

Before we dig into the classifiation tasks for these users, we must convert our text data to vector, so that it can be applied to by classification algorithm. In our project, we adapted three methods for converting vectors to text data, particularly they are one-hot coding, CBOW model and N-gram model.

3.2.1 One-hot Coding

Suppose we have a dictionary W, which contains N words, i.e.$|W| = N$. Then one-hot coding map each word to a vector $x \in \{0, 1\}^N$, where $x_i = 1$ if $x = W_i$, and $x_i = 0$ otherwise. The picture shows a simple example of the principle of one-hot coding.
One possible improvement for one-hot coding is *one-hot hash trick*, whose dimension is reduced by hash function. We also tried to improve the performence by applying hash function. One major shortcoming is unavoidable hash collision, which could map different values to same target.

3.2.2 CBOW Model

CBOW Model stands for continues bag-of-word model, which was illustrated by Xin Rong in his publication. COBW takes the words before and after the target word, to predict the target word, which is actually a way of dimension reduction. In this way, we would be able to get vectorized word

*Email suffix: @shanghaitech.edu.cn

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

representation in much lower dimension, which make it possible and efficient for future classification. The picture shows a simple structure of the network used in CBOW model. One thing to be noted is that the activation function of the hidden layer is linear, which is more similar to *projection layer*.

3.2.3 N-gram Model

Unlike CBOW model, N-gram model intend to predict the target word based on the N word before the target words. Suppose we have a sentence $S = (w_1, w_2, \cdots, w_n)$, we have

$$p(S) = p(w_1 w_2 \cdots w_n) = p(w_1) p(w_2 \mid w_1) \cdots p(w_n \mid w_{n-1} \cdots w_2 w_1)$$

To reduce the parameter space, we adopt *Markov assumption*, which state that the word's appearence only depend on the first N words before it:

$$p(w_1 \cdots w_n) = \prod p(w_i \mid w_{i-1} \cdots w_1) \approx \prod p(w_i \mid w_{i-1} \cdots w_{i-N+1})$$

. Then we could estimate the conditional probability with MLE, which takes the frequencies of words to calculate:

$$p(w_n \mid w_{n-1} w_{n-2}) = \frac{C(w_{n-2} w_{n-1} w_n)}{C(w_{n-2} w_{n-1})}$$

3.3 Paragraph Vectorization

So far, we have make it possible to get the feature vetor from words. What we want to do is to get the feature vector for each user, which represent the feature for whole paragraph.
We decide to apply a naive but efficient method to compute the feature for each paragraph, which is take the weight sum of all word vectors. Suppose the paragraph as K different words, we have $V = \sum_{i=1}^{K} w_i v_i$ where v_i stands for the word vector for word i and $w_i = \frac{\text{\# of word i}}{\text{total length of paragraph}}$.

4 Model for Prediction

4.1 Logistic Regression

Applying logistic regression, we get the model $Z = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n$. Then we use sigmoid function to transform Z to $g(Z) = \frac{1}{1+e^{-z}}$, where $g(Z) \in [0, 1]$. Take $g(Z)$ into the posterior probabilities, and use sum of squares as the loss function, we get $J(\Theta) = \frac{1}{n} Loss(h_\Theta(X), y)$. Repeat the update of $\theta_j^{new} = \theta_j^{old} - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$ until covergence, then we got the paraments needed by the model.

In logistic regression, we firstly find the posterior probabilities of the K classes via linear function in x, which yields $Pr(G = k \mid X = x) = \frac{exp\left(\beta_{k0} + x^T \beta_k\right)}{1 + \Sigma_{l=1}^{K-1} exp(\beta_{l0} + x^T \beta_l)}$, $k = 1, ..., K-1$, then we get the log-likelihood function $l(\theta) = logPr(g \mid X; \theta)$, and use maximum likelihood estimation (MLE) to estimate parameter set $\theta = \{\beta_{10}, \beta_1, ..., \beta_{(K-1)0}, \beta_{K-1}\}$. Then we use Newton-Raphson algorithm to update each β by $\beta_{new} = \beta_{old} - \frac{f'(x_{old})}{f''(x_{old})}$ until covergence.

5 Experiments and Results

6 Discussion

7 Conclusion

8 Contribution

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.