2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST 2017)

Deep Learning based Personality Recognition from Facebook Status Updates

Jianguo Yu
Human Interface Lab
The University of Aizu
Fukushima, Japan
d8182103@u-aizu.ac.jp

Konstantin Markov
Human Interface Lab
The University of Aizu
Fukushima, Japan
markov@u-aizu.ac.jp

***Abstract*—Many approaches have been proposed to automatically infer users personality from their social networks activities. However, the performance of these approaches depends heavily on the data representation. In this work, we apply deep learning methods to automatically learn suitable data representation for the personality recognition task. In our experiments, we used the Facebook status updates data. We investigated several neural network architectures such as fully-connected (FC) networks, convolutional networks (CNN) and recurrent networks (RNN) on the *myPersonality* shared task and compared them with some shallow learning algorithms. Our experiments showed that CNN with average pooling is better than both the RNN and FC. Convolutional architecturewith average pooling achieved the best results 60.0±6.5%.**

***Index Terms*—Big Five model, Automatic Personality Recognition, Convolutional Neural Networks, Social Media.**

I. Introduction

Social networks such as Facebook, Twitter, and Weibo have become essential components of everyday life and hold rich sources that reflect individual's personality. Our personality affects our life choices, well-being, and many other behaviors. During the social interaction, people have to interact with unknown individuals. In order to achieve effective cooperation, it is important to predict the preferences and behaviors of the people we deal with. Such predictions can be found everywhere in the daily life and are often based on the personality of that person. For example, interviewers also consider whether the interviewee's personality is suitable for their company. A girl may consider marriage based on her boyfriend's personality.

Automatic recognition of person's personality from his/her social network activities allows to make predictions about preferences across contexts and environments [1] and has many important practical applications, such as products, jobs, or services recommendation [2] [3], word polarity disambiguation, mental health diagnosis, etc.

Many approaches have been proposed to automatically infer users' personality from the content they generate in social networks. However, the performance of these approaches depends heavily on the data representation which often is based on hard-coded prior knowledge.

Recently, deep learning approaches have obtained very high performance across many different natural language processing (NLP) tasks. Unlike traditional methods, deep learning approaches can learn suitable representation automatically.

In this work, we implemented several deep learning algorithms including fully-connected neural networks (FC), convolutional neural networks (CNN) and recurrent neural networks (RNN) in our personality recognition system and evaluated it on the task from the "Workshop on Computational Personality Recognition (Shared Task)" [4]. For classification performance comparison, we used the same task results from some traditional shallow machine learning methods published elsewhere.

II. Related Work

In the context of this study, personality is formally described by five dimensions known as the Big-Five personality traits [5]:

- **EXT**raversion vs. Introversion (sociable, assertive, playful vs. aloof, reserved, shy).
- **NEU**roticism vs. Emotional stability (calm, unemotional vs. insecure, anxious).
- **AGR**eeableness vs. Disagreeable (friendly, cooperative vs. antagonistic, faultfinding).
- **CON**scientiousness vs. Unconscientious (self-disciplined, organised vs. inefficient, care-less).
- **OPE**ness to experience (intellectual, insightful vs. shallow, unimaginative).

Automatic recognition of personality typically involves binary classifications of which trait types an user belongs to given the content generated by him/her. The true labels are usually obtained by self-assessment questionnaire [6]. A variety of approaches have been proposed for this task utilizing different classifiers and feature spaces. Until recently, most of the models were based on shallow learning approaches such as Support Vector Machine (SVM) [7] [8], Naive Bayes classifier (NB) [9], K-Nearest Neighbors (kNN) [10], and Logistic Regression (LR) [11]. In the early studies, text features were typically extracted by tools like Linguistic inquiry and word count (LIWC) [12] and good results were usually achieved by selecting features from a very large feature space like [13], which achieved a very high classification performance on the myPersonality task using ranking algorithms for feature selection and SVMs and Boosting as learning algorithms. Deep

978-1-5386-2965-9/17/$31.00

learning based method was also proposed in [14] recently, where convolutional neural networks are applied to extract n-gram information from stream-of-consciousness essays [15].

III. System Description

A. Word Embedding

Currently, word embedding has become a standard component for the DNN based natural language processing. It converts the one-hot representation of the word to a distributed representation [16], which has many benefits and allows to map words with similar meaning to similar values: the learning of one word can indirectly help the learning of the other words with similar meaning. This is especially helpful for tasks with small training data.

In order to utilize the statistical knowledge of the text, we pre-train the word embedding matrix with the text data using the skip-gram method [17]. We didn't use Google pre-trained word2vec because the statuses contain internet-slang, emoticons (e.g., :-D), acronyms (e.g., BRB-be right back) and various shorthand notations, which carry rich information about personality, but are not included in the Google model.

B. Network Architecture

In our neural networks, the first several layers are intended to automatically extract features from the raw text. Then, extracted features can be concatenated with other features and fed to the output layer for final classification. These are the main components of our networks:

- **Inputs:** our network takes input pairs of the form $\{(x_i, a_i)\}_{i=1}^{n}$, where n is the number of statuses. The text input $x_i \in R^{W \times V}$ contains W one-hot vectors of words with V vocabulary size. The $a_i \in R^A$ is nonverbal information data of A dimensions.
- **Target:** learning targets $\{t_i\}_{i=1}^{n}$ are binary classification labels for one trait (there are five traits in total) and $t_i \in R^2$ is represented as 2 dimensional one-hot vector.
- **Embedding layer:** the text input x_i is be transformed by the embedding matrix W_E to a distributed representation $e_i = W_E \cdot x_i$, where $W_E \in R^{V \times E}$ and E is the dimension of word embedding.
- **Convolution layer:** multiple convolutional filters extract n-gram information from e_i as shown in Fig.1. The green box represents a bigram CNN filter which considers two words each time and slides over all words to create a feature map.
  A convolutional filter of size $n \times E$ is applied on status $e_i \in R^{W \times E}$ to extract the *n*-gram features, where n =1, 2, 3 for the unigram, bigram, and trigram. In every convolutional layer, K filters are applied to each status e_i producing a matrix $F_n^{conv} \in R^{K \times n \times E}$ to which bias $B_n^{conv} \in R^K$ is added, resulting in $FM_n^{conv} \in R^{K \times (W-n+1) \times 1}$. A Rectified Linear Unit (ReLU) function is then applied on FM_n^{conv} to introduce non-linearity.
- **Avg / max pooling layer:** it performs a max or average pooling operation on the convolutional layer output FM^{conv} to obtain a feature vector $PFM_n \in R^K$. All *n*-grams CNN feature maps will be concatenated into $PFM_{all} \in R^{(K \times n)}$. It seems that max pooling extracts the most important information from the status but doesn't take into account how many times such information appears in a status while average pooling does the opposite.

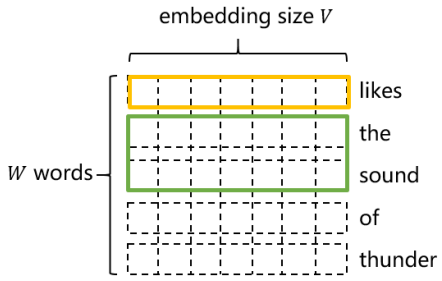

Fig. 1. Illustration of unigram and bigram convolutional filters.

- **Non-lexical features:** recognition of personality which depends only on text features could be misleading since words meanings vary in different contexts. Therefore, it is beneficial to use other non-lexical features, if such are available. This can be done by concatenating the text features $PFM_{all} \in R^{(K \times n)}$ extracted using convolutional nets with the non-lexical features $a_i \in R^A$, resulting in $concat(PFM_{all}, a_i) \in R^{(K \times n + A)}$.
- **Fully connected layer:** to transform the combined features $concat(PFM_{all}, a_i)$ to higher-level representation which may be shared across different samples, we added several fully-connected layers. The activation function of these layers is also ReLU. The output of two fully-connected layers is computed as:

$$x^{fc_1} = \sigma(W_{fc_1} \cdot concat(PFM_{all}, a_i) + b_{fc_1}) \quad (1)$$

$$x^{fc_2} = \sigma(W_{fc_2} \cdot x^{fc_1} + b_{fc_2}) \quad (2)$$

where $W^{fc_1} \in R^{(K \times n + A) \times F}$, $W^{fc_2} \in R^{F \times F}$, and σ is $max(x, 0)$.
- **Output layer:** a softmax output layer is added to maximize the probabilities of the trait being yes and no. It's output is obtained from:

$$y_i = \sigma(W_o \cdot x^{fc_2} + b_o) \quad (3)$$

where $W_o \in R^{F \times 2}$, and σ is the softmax function.
- **Loss function:** we use cross entropy (4) as the loss function to minimize:

$$H(t_i, y_i) = -\sum_{j=1}^{2} t_{ij} \log y_{ij} \quad (4)$$

The block diagram of our system using two n-grams (unigram and bigram) CNN is illustrated as Fig.2-A. Using this system trained 5 separate networks, one for each of the 5 traits.

978-1-5386-2965-9/17/$31.00 © 2017 IEEE

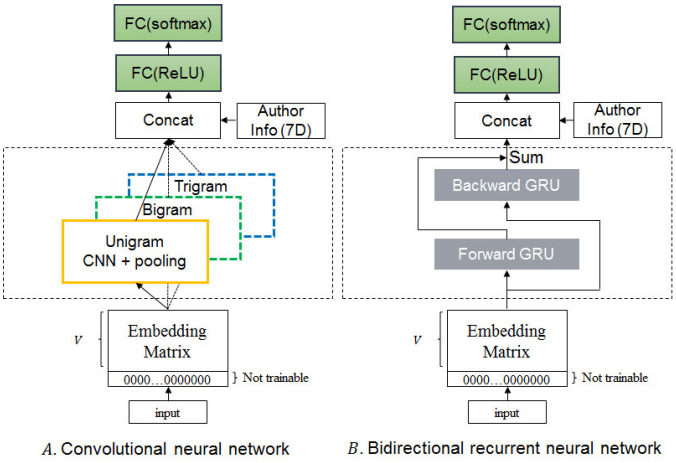

Fig. 2. Network architecture.

IV. Experiments and Results

A. Dataset

The dataset used in our experiments is a subset (250 users with 9917 status updates) of the database released by organizers of the "Workshop on Computational Personality Recognition (Shared Task)" [4]. It contains Facebook statuses in raw text, author information (network size, betweenness, nbetweenness, density, brokerage, nbrokerage, and transitivity) and gold standard Big-5 personality labels (obtained using self-assessments questionnaire). The personality labels have both scores and classes. Classes have been derived from scores with a median split. In this work, we focus on personality classification.

It is suggested in the shared task guidelines to split the data as train (66%) and test (33%). Because each author has multiple statues in this dataset, the train set can see all examples from 250 authors if randomly splitting 9917 statuses into train/test sets. So we divided 250 authors into 3 parts, each of which contains about 3300 statuses. We used 3-fold cross validation (CV) for our performance evaluation.

During tokenization, we treat each internet-slang word as an unique word and map all web addresses to the same word "*URL". Such processing was also applied to digit numbers, time and currency. This way, we got about 15000 unique words and maximum word length of a single status was 78. We set the vocabulary size to 4400, which is the number of words appearing more than 2 times in the data. Uncommon words then are replaced by "UNK". We concatenated a zero-likes vector $z \in R^{1\times V}$ to embedding matrix W_E which is not trainable as shown in Fig.2, and pad all statuses to the length of 78 with 4401 index.

We trained the embedding matrix W_E learning in advance from the raw text of this dataset using skip-gram. The skip window was set to 1 and embedding size to 128 dimensions. The embedded matrix is kept fixed during the network training.

B. Results

In our experiments, each network was trained by 100 epochs using Adam [18] update method with a learning rate of 1e-4. Then, using their predictions on the test data we calculated the classification accuracy and F1 score metrics results for test set.

Each of our architectures can be seen as three consecutive parts: embedding part (embedding layer), feature extraction part and classification part (one fully connected layer). We experimented feature extraction part with Convolutional architecture (Fig.2-A), bidirectional recurrent architecture (Fig.2-B) and fully-connected architecture where the module in the dotted box is replaced one fully-connected layer. Convolutional architectures are also tested with max pooling and average pooling layers. The output of feature extraction part is also concatenated with 7 dimensional author information. The Overall DNN settings are shown in Table I. All results are given as mean and standard deviation of 3-fold cross validation experiments.

TABLE I
Overall DNN settings

Length of status	78
Embedding size	128
loss function	Categorical Cross-entropy
Output activation	SoftMax
hidden activation	ReLU
Nodes of FC layer	32
# CNN filters per n-gram	32
GRU (per direction) nodes	32

- **Fully-connected architecture**
  Table II shows the accuracy and F1 score results when the module in the dotted box (Fig.2) is replaced one fully-connected layer.

TABLE II
CLASSIFICATION TEST RESULTS WITH DIFFERENT FEATURES USING FULLY-CONNECTED ARCHITECTURE.

Trait	ACC%	F1%
Author Information Only		
EXT	68.4±3.0	63.3±2.7
NEU	62.4±1.7	55.5±5.1
ARG	57.6±3.6	55.8±3.6
CON	52.0±4.3	49.8±4.2
OPN	70.4±4.2	42.5±2.5
Overall	61.7±7.4	52.5±7.7
Text Only		
EXT	51.8±1.6	49.9±1.6
NEU	53.0±1.7	49.3±1.3
ARG	49.7±1.4	49.4±1.3
CON	49.8±1.6	49.3±1.7
OPN	64.5±5.7	49.1±0.5
Overall	53.8±6.0	49.3±1.1
Author+Text		
EXT	59.5±0.2	58.0±1.3
NEU	60.7±4.8	55.6±3.5
ARG	53.8±2.8	53.0±2.0
CON	51.4±0.9	51.2±1.4
OPN	72.4±8.5	53.1±3.4
Overall	59.6±8.2	53.8±3.1

978-1-5386-2965-9/17/$31.00 © 2017 IEEE

When using a fully-connected layer for feature extraction, it overfits heavily. The help of dropout is tiny. We also tested models that use only text or author information as input. The results using author information were tested on author level (250 examples).

The results shows that it gives better results to combine both text and author information as input. However, models with only text input didn't perform well. So the following results are all models with both text and author information.

We tried to initialize embedding matrix randomly and learn it during the training of classification task. We also used the embedding matrix pre-trained on Wikipedia using FastText [19]. But both were worse than the embedding pre-trained using test from this dataset. Perhaps, the expression people use for chatting is quite different from the one for writing.

- **Convolutional architecture**
  Table III shows the accuracy and F1 score results when using a uni-gram convolutional layer with average pooling. We tuned the L2 regularization and drop out rates to get better results for each trait.

TABLE III
CLASSIFICATION RESULTS USING CONVOLUTIONAL ARCHITECTURE WITH AVERAGE OR MAX POOLING

Big-5	Average pool		Max pool	
Trait	ACC%	F1%	ACC%	F1%
EXT	65.8±4.9	65.2±5.0	59.7±2.4	58.6±1.8
NEU	67.9±2.1	62.7±4.7	59.0±3.1	54.9±3.8
ARG	59.8±4.9	57.0±3.4	51.9±0.3	51.4±0.6
CON	53.7±1.1	53.3±1.1	52.8±1.0	52.6±1.2
OPN	74.0±12.2	61.3±11.1	64.9±5.4	54.0±4.6
Overall	64.2±8.7	60.0±6.5	55.4±5.4	54.1±3.4

We found that no improvement was achieved when the combination of unigram, bigram, trigram CNN filters was used, but it did enhance the learning capacity. We also found that max pooling can learn faster than average pooling, but overfits heavily. Convolutional architecture with average pooling achieved the best results 60.0±6.5%, which is better than competition results of F1 58.6% from two teams [9] [10] on *myPersonality* shared task in 2013.

- **Recurrent architecture**
  With this architecture, we replaced the module in the dotted box by a bi-directional GRU layer [20] where forward GRU and backward GRU are summed up in the output of this layer. The RNN result is listed in Table IV. The RNN results showed that recurrent architecture is able to extract some sequential meanings from the text for personality recognition, but still not as good as convolutional layer with average pooling. It seems that the selection of words reflects more personality than the meaning itself.

V. CONCLUSION AND FUTURE WORK

In this work, we applied deep learning approaches including convolutional neural networks and recurrent neural networks on the shared task from "Workshop on Computational Personality Recognition (Shared Task)" [4].

TABLE IV
CLASSIFICATION RESULTS FOR RECURRENT OF ARCHITECTURES

Trait	ACC%	F1%
EXT	60.9±0.3	59.8±0.9
NEU	61.1±4.5	55.5±3.1
ARG	54.3±3.7	53.7±3.0
CON	50.9±0.2	50.6±0.3
OPN	70.6±7.9	56.4±9.1
Overall	59.6±7.6	55.2±4.8

The results showed that FC models with only text are worse than the ones with author information. DNN with both text and author information achieves the best results. CNN, RNN and FC are able to automatically extract useful features for personality recognition and the best result of 60.0±6.5% F1 score was obtained using CNN with average pooling. We found that Bi-gram, tri-gram and recurrent architecture didn't get better results. It may indicate that words chosen by the author tell more about author's personality than the meaning author expresses. We also noticed that applying regularization barely restrains the overfitting, the network learns the patterns that only exist in the training set. This may be improved by collecting more data or by transforming the text into a representation which is stable for personality by external knowledge.

In the future work, we plan to apply unsupervised learning on the text data and use external knowledge about personality to cluster the text.

REFERENCES

[1] A. Vinciarelli, and G. Mohammadi, "A survey of personality computing." IEEE Transactions on Affective Computing, vol. 5, no. 3, pp. 273-291, 2014.
[2] L. Piwek, D. A. Ellis, S. Andrews, and A. Joinson, "The rise of consumer health wearables: promises and barriers," PLoS Medicine, vol. 13, no. 2, pp. e1001953, 2016.
[3] M. Denscombe, The good research guide: for small-scale social research projects. McGraw-Hill Education (UK), 2014.
[4] F. Celli, F. Pianesi, D. Stillwell, and M. Kosinski, "Workshop on computational personality recognition (shared task)," in Proceedings of the Workshop on Computational Personality Recognition, 2013.
[5] L. R. Goldberg, "The structure of phenotypic personality traits." American psychologist, vol. 48, no. 1, p. 26, 1993.
[6] P. T. Costa Jr and R. R. McCrae, "Domains and facets: Hierarchical personality assessment using the revised neo personality inventory," Journal of personality assessment, vol. 64, no. 1, pp. 2150, 1995.
[7] T. Polzehl, S. Moller, and F. Metze, "Automatically assessing personality from speech," in Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on. IEEE, 2010, pp. 134140.
[8] B. Verhoeven, W. Daelemans, and T. De Smedt, "Ensemble methods for personality recognition," in Proceedings of the workshop on computational personality recognition, 2013, pp. 3538.
[9] F. Alam, E. A. Stepanov, and G. Riccardi, "Personality traits recognition on social network-facebook," WCPR (ICWSM-13), Cambridge, MA, USA, 2013.
[10] G. Farnadi, S. Zoghbi, M.-F. Moens, and M. De Cock, "Recognising personality traits using facebook status updates," in Proceedings of the workshop on computational personality recognition (WCPR13) at the 7th international AAAI conference on weblogs and social media (ICWSM13). AAAI, 2013.

978-1-5386-2965-9/17/$31.00 © 2017 IEEE

[11] M. T. Tomlinson, D. Hinote, and D. B. Bracewell, "Predicting conscientiousness through semantic analysis of facebook posts," Proceedings of WCPR, 2013.
[12] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," Mahway: Lawrence Erlbaum Associates, vol. 71, no. 2001, p. 2001, 2001.
[13] D. Markovikj, S. Gievska, M. Kosinski, and D. Stillwell, "Mining facebook data for predictive personality modeling," in Proceedings of the 7th international AAAI conference on Weblogs and Social Media (ICWSM 2013), Boston, MA, USA, 2013, pp. 2326.
[14] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," IEEE Intelligent Systems, vol. 32, no. 2, pp. 7479, 2017.
[15] J. W. Pennebaker and L. A. King, "Linguistic styles: language use as an individual difference." Journal of personality and social psychology, vol. 77, no. 6, p. 1296, 1999.
[16] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010, pp. 384394.
[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.
[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[19] P Bojanowski, E Grave, A Joulin, et al. Enriching word vectors with subword information[J]. arXiv preprint arXiv:1607.04606, 2016.
[20] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 2015, pp. 2342 2350.

978-1-5386-2965-9/17/$31.00 © 2017 IEEE