Applied Intelligence (2018) 48:4232–4246
https://doi.org/10.1007/s10489-018-1212-4

Deep learning-based personality recognition from text posts of online social networks

Di Xue[1] · Lifa Wu[1] · Zheng Hong[1] · Shize Guo[2] · Liang Gao[2] · Zhiyong Wu[1] · Xiaofeng Zhong[3] · Jianshan Sun[4]

Published online: 5 June 2018

Abstract
Personality is an important psychological construct accounting for individual differences in people. Computational personality recognition from online social networks is gaining increased research attention in recent years. However, the majority of existing methodologies mainly focused on human-designed shallow statistical features and didn't make full use of the rich semantic information in user-generated texts, while those texts are exactly the most direct way for people to translate their internal thoughts and emotions into a form that others can understand. This paper proposes a deep learning-based approach for personality recognition from text posts of online social network users. We first utilize a hierarchical deep neural network composed of our newly designed AttRCNN structure and a variant of the Inception structure to learn the deep semantic features of each user's text posts. Then we concatenate the deep semantic features with the statistical linguistic features obtained directly from the text posts, and feed them into traditional regression algorithms to predict the real-valued Big Five personality scores. Experimental results show that the deep semantic feature vectors learned from our proposed neural network are more effective than the other four kinds of non-trivial baseline features; the approach that utilizes the concatenation of our deep semantic features and the statistical linguistic features as the input of the gradient boosting regression algorithm achieves the lowest average prediction error among all the approaches tested by us.

Keywords Personality recognition · Deep learning · Online social networks · Big Five personality

1 Introduction

Personality is a psychological construct aimed at explaining various human behaviors in terms of a few stable and measurable individual characteristics [1]. It not only reflects an individual's consistent patterns of behavior, thought and interpersonal communication [2], but also influences important life aspects [3], including happiness, motivations, preferences, emotion, mental and physical health [4]. The dominant paradigm for formal description of personality in psychology [1] is the Big Five, also known as the Five Factor Model, which consists of five basic traits: openness to experience (O), conscientiousness (C), extraversion (E), agreeableness (A), and neuroticism (N) [5]. The study of personality is foundational to psychology, and personality recognition (PR) [6] can benefit many other applications, such as social network analysis [7], recommendation systems [8], deception detection [9], authorship attribution [10], sentiment analysis/opinion mining [11] and so on [12]. However, the traditional methods of personality assessment through questionnaire investigation or expert interview are costly and less practical in cyber space [13].

Along with the popularization of social media in recent years, automatic personality recognition from online social network (OSN) is gaining increased research attention because of its potentials in many computational applications [14]. There are rich self-disclosed personal information and emotional contents on social media, which have been proved to be highly correlated with user's personality traits [15–22]. Numerous studies have been done to explore optimal feature space and machine learning algorithms for recognizing individual's personality [23–28]. However, the achievements of existing methodologies are not satisfactory. For one thing, the majority of existing

✉ Zheng Hong
hz5215@163.com

1 Army Engineering University, Nanjing 210007, China

2 Institute of North Electronic Equipment, Beijing 100083, China

3 Electronic Engineering Institute, Hefei 230037, China

4 Hefei University of Technology, Hefei 230009, China

approaches focused on human-designed shallow linguistic features (e.g. dictionary-based statistical features, n-grams, topics) extracted from the user-posted texts, or basic statistical features obtained from self-disclosed personal information on users' profiles. These practices did not seem to make full use of the rich user-generated text information on social networks, while those words and texts are exactly the most direct and reliable way for people to translate their internal thoughts and emotions into a form that others can understand. Therefore, it may contribute a lot to PR task if taking the contextual information and word orders into account to capture meaningful syntactic and semantic features when modeling user's text posts. For another, most previous approaches solved the PR problem as a classification task, which simply split subjects into two or three classes. This kind of outputs are not meaningful from a psychological point of view and is not useful enough for practical purposes, because it can hardly provide convincing arguments when emphasizing comparisons among individuals, which is exactly what humans loves to do [4]. Therefore, PR models that output real-valued scores for personality traits would be more suitable and psychologically meaningful.

Besides, in recent years, deep learning based neural networks [29] and distributed representation [30, 31] have been demonstrated to be powerful in sentence/document modeling and achieved remarkable performance in natural language processing (NLP) applications, such as text-based sentiment analysis [32], opinion mining [33], etc. It is worth noting that these NLP applications seem to be similar to our personality recognition task, since they both involve mining user attributes from texts, and feature representation of texts could be their common challenge. Given this, to improve the performance of personality recognition approaches, the most intuitive and straightforward idea is to introduce the powerful text modeling techniques that have been successfully applied in NLP domains into the field of personality recognition.

Considering the above-mentioned limitations of existing PR approaches and the potentials of deep learning and distributed representation, we propose a hierarchical deep neural network-based method to predict the big five personality scores of OSN users from their text posts. Specifically, we design an AttRCNN structure by introducing the attention mechanism [34, 35] and batch normalization (BN) technique [36] to the recurrent-conventional neural network (RCNN) [37] to perform the vectorization of a single text post, and combine the AttRCNN structure with a variant of the convolutional neural network (CNN) based Inception structure [36, 59, 60] through a hierarchical architecture to learn deep semantic representations of the aggregation of each user's text posts. Then we concatenate these deep semantic representations with pre-extracted statistical linguistic features vectors to construct the final feature space, and adopt the gradient boosting regression (GBR) [38] algorithm to predict the Big Five personality scores for users. Our study is carried out based on the dataset from MyPersonality Project [12], which contains more than 11 million Facebook users' profile data and Big Five personality scores tested via online psychometric tests. Experimental results demonstrate that the deep semantic features learned from our neural network are more effective than the other four kinds of non-trivial baseline features, and our recognition approach surpasses all the others with the lowest prediction errors.

In summary, the contributions of our work are as follows:

(1) We design a new AttRCNN structure of neural network to learn the distributed semantic representation of OSN user's single text post.
(2) By combining the AttRCNN structure with a variant of the CNN-based Inception structure we propose a new hierarchical deep neural network named AttRCNN-CNNs to learn deep semantic representations of the aggregation of each OSN user's text posts.
(3) Based on the distributed semantic representations of user's text posts learned from the deep neural networks, we propose a personality recognition methodology, successfully applying the deep learning techniques on text corpora of OSN users for personality tasks.

The rest of the paper is organized as follows. We discuss related work in Section 2. The details of our proposed neural network architecture and personality recognition approach are described in Section 3. Section 4 presents the experimental evaluation, followed by the conclusion and future work in Section 5.

2 Related work

2.1 Computational personality recognition

Along with the explosive popularity of social media, various studies have been carried out for personality recognition from OSNs [23–28]. In the year of 2011, Golbeck et al. [23] extracted 77 features from 167 Facebook users' egocentric network, personal information, language usage, preferences and activities, and adopted M5' Rules and Gaussian Processes to predict their Big Five scores. A similar approach was also applied over 279 Twitter users by Golbeck et al. [24] in the same year. Quercia et al. [25] analyzed the relationship between personality and different types of Twitter users, and applied M5 algorithm to predict 335 users' Big Five traits simply based on three publicly available counts: follower, following and listed counts (i.e. the

number of individuals that include the user in their reading list). Alam et al. [27] followed bag-of-words approach and used tokens (unigrams) as feature input of different classification methods, namely Sequential Minimal Optimization (SMO) for SVM, Bayesian Logistic Regression (BLR) and Multinomial Naïve Bayes (MNB) sparse modeling, to predict Big Five traits based on the *MyPersonality*[1] corpus collected from Facebook by Celli et al. [12]. Skowron et al. [28] carried out PR research based on text, image, and users' meta data collected from two popular social networking sites, i.e., Twitter and Instagram, and found that such joint analysis could improve the prediction accuracy.

In addition to the utilization of English corpora, personality recognition research has also been carried out in Chinese language environments [39–43]. Bai et al. [39] analyzed the demographic information, usage statics and emotional states of 209 users on RenRen, a Chinese social networking platform, and applied C4.5 decision trees to classify users into three groups of low, middle or high scores. Li et al. [42] carried out PR experiments over 547 Chinese active users of Sina Weibo. They extracted not only static features from users' profiles, privacy setting, self-expression and interpersonal behaviors but also dynamic features consisting of micro-blogging updates, @ mentions, use of apps and recordable browsing. Peng et al. [43] used a Chinese text segmentation tool named Jieba as the tokenizer to process the texts of 222 Chinese Facebook users, and adopted SVM to classify their personality traits. They reported that text segmentation and utilization of side information such as the number of friends could contribute to the performance improvement [43].

Overall, most existing approaches to personality recognition adopted classification methods to solve the PR problem, and previous efforts on feature space exploration mainly concentrated on statistics of users' online activities or profile information and human-designed shallow features of texts. They did not seem to make full use of the rich user-generated text information on OSNs. In this paper, we mainly focus on text information to predict user's Big Five personality traits.

As for personality recognition that concentrated on written texts, there have been various studies, too. Argamon et al. [44] took word categories and relative frequency of function words as the input of Support Vector Machines (SVM) to discriminate between students at the opposite extremes of Extraversion and Neuroticism. Mairesse et al. [45] studied the effectiveness of different sets of textual features extracted from psychologically oriented text analysis tools (e.g. LIWC[2] [46]) or psycholinguistic dictionary (e.g. MRC [47]). In [48] and [49], the frequencies of N-grams (i.e. N-long word sequences) were extracted as input features of Naïve Bayes classifiers and SVM to classify high and low scoring bloggers for Big Five traits. Recently, Majumder et al. [50] adopted the Convolution Neural Networks (CNNs) to model document and extract deep semantic features to recognize personality from texts. The accuracy of this approach outperformed the baselines for all Big Five personality traits, making their work the state of the art.

Besides, to provide corpora and tools for standard evaluation of PR approaches, *Workshop on Computational Personality Recognition (Shared Task)* was organized in 2013 [12] and 2014 [5], and another shared task of personality recognition was organized under the umbrella of *Author Profiling task at PAN 2015* [51]. Unfortunately, the corpora they provided are not large enough to carry out deep learning based study, so we didn't use them in our research.

2.2 Deep neural networks

In recent years, deep neural networks [29] have achieved remarkable performance in sentence/document modeling, which is the foundational task in many natural language processing (NLP) applications such as text classification [37, 52], sentiment analysis [32, 53], etc. Among all these models, convolutional neural network (CNN) [54, 55] and recurrent neural network (RNN) [56, 57] constructed on top of pre-trained word embeddings are two mainstream architectures, which adopt different ways of understanding natural language and both have their own strengths and weaknesses in text modeling. CNNs achieve good performance in extracting n-gram features at different positions of a sequence through convolutional filters, but they are not good at capturing long-term sequential correlations. RNNs can handle sequences of arbitrary input/output lengths and capture long-term dependencies, but RNNs are biased models, in which later words are more dominant than earlier words [58].

To better model sentences/documents, various modified architectures were proposed based on the basic CNN and RNN. The most relevant structures to our work is the Gated Recurrent Units (GRU) [34], recurrent convolutional neural network (RCNN) [37] structure and the CNN-based Inception architectures [36, 59, 60]. The GRU is a variant of RNN, which uses a gating mechanism to track the state of sequences without using separate memory cells [52]. RCNN is proposed by Lai et al. [37] to deal with text classification task in 2015. They applied a bi-directional recurrent structure to capture contextual information as far as possible when learning word representations, which may introduce much less noise compared to traditional window-based CNN neural networks. As for the Inception architecture [36, 59, 60], it was proposed by Szegedy et al.

[1] http://mypersonality.org

[2] http://www.liwc.net

to keep relatively low computational budget while increasing the depth and width of the CNN networks.

3 Methodology

Language is the most common and reliable way for people to translate their internal thoughts and emotions into a form that others can understand. Words and language, then, are the very stuff of psychology and communication [15]. Texts tend to reflect various aspects of the author's personality [50], and if we could model the OSN user's text posts better, the performance of PR approaches would improve a lot. Motivated by this intuition, we propose a hierarchical deep neural network based on our newly designed AttRCNN structure and a variant of CNN-based Inception structure, from which we extract the deep semantic vector representations of the aggregation of each user's text posts. Then we concatenate them with pre-extracted global statistical features to construct the input feature space for traditional regression algorithm to carry out final prediction of each user's real-valued Big Five personality scores.

Overall, our methodology includes four phases: (1) Text post preprocessing phase is to tokenize and unify users' text posts; (2) Statistical feature extraction phase is to extract global statistical features by directly counting the frequency of target text elements in each user's text posts; (3) In deep learning-based text posts modeling phase, word embeddings are firstly trained through unsupervised learning. Then our newly designed deep neural network for text posts modeling are built utilizing the Facebook corpus; (4) Prediction phase is to predict real-valued Big Five personality scores with traditional regression algorithm based on the deep semantic features extracted from the neural network and the pre-extracted global statistical features.

3.1 Text posts preprocessing

3.1.1 Text tokenization

The target of this step is to tokenize each text-only status update into a sequence of tokens, which are separated by a space and roughly correspond to "words".

Considering that people with different personalities may have different habits of using punctuations, symbols, emoticons and capital letters, we choose to keep the original elements of each text post as much as possible and do not remove any words, letters or symbols in this step, so that we could extract relatively complete features (i.e. special linguistic statistics features) from the users' status updates. Specifically, we only add necessary spaces between different text elements (words, punctuations, emoticons, URLs, numbers, etc.) and delete unnecessary spaces within a single text element, e.g., emoticons like ^_^, (*~*), to ensure that each text element could be treated as a single complete token, e.g. ^_^, (*~*), etc., rather than a sequence of meaningless separated symbols or punctuations.

The outputs of this step are named as the **tokenized text posts**, which would not only be the processing objects of the following text unification step, but also the input of the special linguistic statistical feature extraction phase in Section 3.2.1.

3.1.2 Text unification

Users of online social networks tend to use informal language which may contain casually defined terms and punctuations, such as 'busyyyy', 'busyyyyyyyyy', '!!!', '!!!!!', etc. The number of this kind of usage may contribute to personality recognition, since people may emphasize their emotions by repeating letters or symbols. However, these kinds of raw texts may also directly affect the quality of the word embeddings trained based on them and further influence the performance of prediction models, because the same term with different number of tandem duplicated letters or punctuations would be considered as different "words" in the following training process of word embeddings. Thus, in this text unification step, we reduce the length of such tandem duplicated elements to make sure the length of such elements in a certain token is no more than 3, and further convert the text into lower case.

The outputs of this processing step are named as the **unified text posts**, which would be the input of dictionary-based linguistic feature extraction process (Section 3.2.2) and the word embeddings learning process (Section 3.3.1).

3.2 Statistical feature extraction

3.2.1 Extracting special linguistic statistics features

As mentioned above, people with different personalities may have different habits of using punctuations, symbols, emoticons and capital letters. Users of online social networks tend to use informal language which may contain casually defined terms and punctuations, and they may emphasize their emotions by using capital letters and emoticons, repeating letters or symbols in one term, and so on. Considering that the statistics of these special tokens in user-generated texts may contribute to personality recognition, we extract the following 5 special linguistic statistical features from the **tokenized text posts**: (1) rate of emoticons; (2) rate of tokens which have no less than 3 tandem duplicated letters or symbols; (3) rate of capital letters; (4) rate of capitalized words; (5) total number of text posts of each user.

3.2.2 Extracting dictionary-based linguistic features

Correlations between lexical categories of user generated texts and user's personality have been widely proved by previous psychological studies [14, 45]. In this paper, we adopt the Linguistic Inquiry and Word Count (LIWC) tool, a popular text analysis software widely used in psychology studies [61], to extract the psychology dictionary-based linguistic features. For each user, we first aggregate all his/her **unified text posts**, preprocessed by text tokenization and unification in Section 3.1, to construct a unified **document** for him/her. Then, with the aid of LIWC tool, we extract 64 features from each Facebook user's unified document, which includes features related to standard counts (e.g., word count), psychological processes (e.g., the number of anger words such as hate and annoyed in the document), relativity (e.g., the number of verbs in the future tense), personal concerns (e.g., the number of words that refer to occupation such as job and majors), and linguistic dimensions (e.g., the number of swear words). For a complete overview of the features, please refer to literature [62].

3.3 Deep learning based text posts modeling

3.3.1 Unsupervised learning of word embeddings

Word embeddings are distributed representations of words that can capture meaningful syntactic and semantic regularities [37]. The underlying idea of word embedding is the "distributional hypothesis" proposed by Zellig Harris [63], which can be summarized as "a word is characterized by the company it keeps" and "words that occur in the same contexts tend to purport similar meanings". Word embeddings are dense, low-dimensional, real-valued vectors, which can be generated using large unlabeled text data and are suitable as input for neural network models to alleviate the data sparsity problem. Previous studies have shown that word embeddings can boost the performance of deep learning methods in numerous natural language processing (NLP) tasks [64] such as text classification and sentiment analysis.

In this work, to model OSN user's text posts better, we try to extract the deep semantic features from the texts based on deep neural network, where pre-training word embeddings (i.e., word-level semantic features) should be the first step. We adopt the CBOW model [65], state-of-the-art in many NLP tasks, to pre-train word embeddings by the aid of word2vec,[3] a popular word embedding toolkit developed by Mikolov et al. which is used on corpus of 11 million Facebook users' text posts. Finally, each word in the vocabulary is represented as a real-valued vector of fixed length (i.e., E dimensions).

After obtaining the word embeddings, we construct the embedding matrix M_e, which would be used in the embedding layer of the neural network. For the unknown words that do not appear in the pre-trained word list, we assign all its E coordinates randomly with a uniform distribution in $[-0.25, 0.25]$.

3.3.2 Supervised learning of deep semantic features

To learn the deep semantic features of each user's text posts, we propose a two-level hierarchical deep neural network model, code-named AttRCNN-CNNs, whose schematic is shown in Fig. 1. For the sake of clarity, we refer to each unified text post of a Facebook user as a **sentence**, and the aggregation of each user's unified text posts as a **document**.

The input of our neural network is a 3-dimensional real-valued array from $\mathbb{R}^{N\times S\times W}$, where N is the total number of documents, S is the number of sentences in each document and W is the number of words in each sentence.

As shown in Fig. 1, we first utilize the AttRCNN-based sentence encoder to learn semantic vector representations of sentences, and then we apply the CNN-based document encoder to extract document vectors from the aggregation of previously-learned sentence vectors. The detailed structures and components of these two encoders are presented in Section 3.3.2-(1) and -(2).

As for the output of our neural network, we apply a fully connected layer of size 1 (denoted as FC (1) in Fig. 1) on top of the document encoder, and further apply the function in (3.1) as the activation function (i.e., Custom Activation in Fig. 1) to constrain the output in the [1, 5] range, so that its output could be compared with the real personality score to compute the loss value and further help tune the parameters of the model.

$$Out(x) = 1 + \frac{4}{(1 + e^{-x})} \tag{3.1}$$

(1) **Sentence Vectorization with AttRCNN**

Inspired by the popular RCNN model [37] applied for text classification task, we design a new structure named AttRCNN for sentence vectorization by introducing the attention mechanism [34, 35] and batch normalization technique [36] into RCNN model to modify the way it captures the semantics of context. The intuition underlying our modification is that not all words that occur around a certain word w_{it} contribute equally to the semantics of w_{it}'s context, thus, we introduce the attention mechanism to help find the informative contextual words and learn the left- and right-side context vectors of w_{it} better. Details are as follows.

[3] https://code.google.com/p/word2vec/

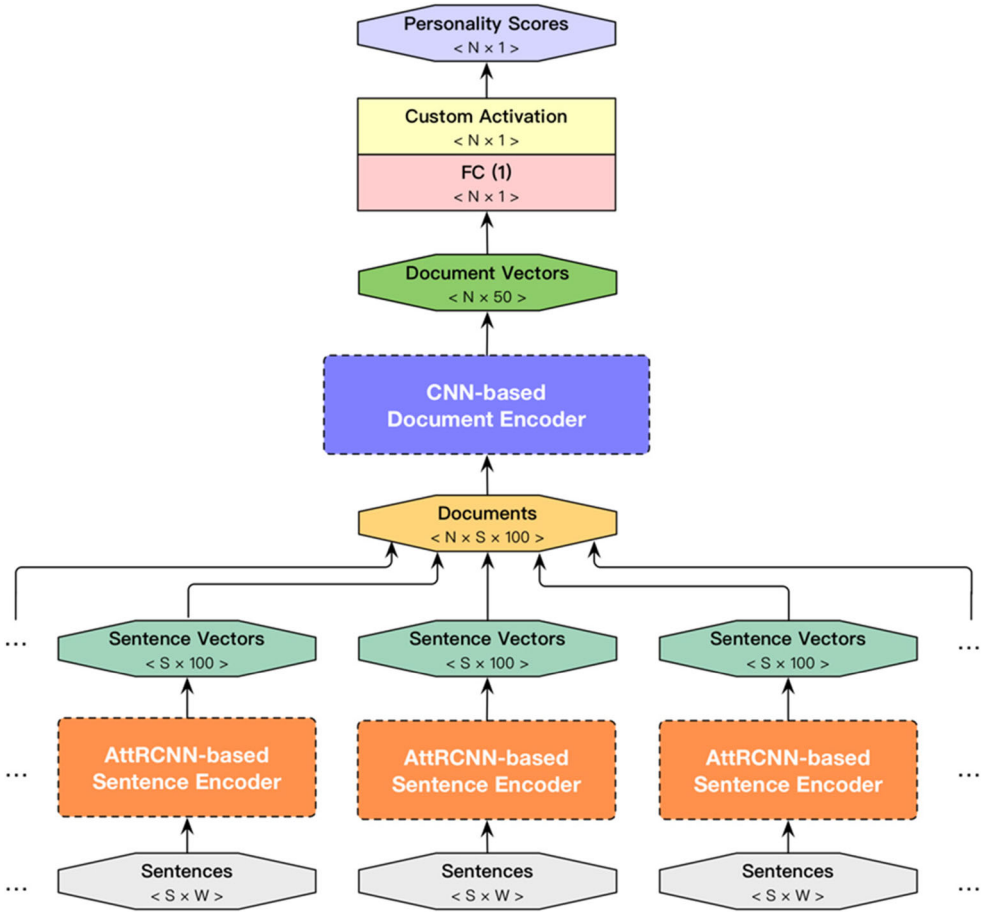

Fig. 1 Schematic of our proposed hierarchical neural network for text posts modeling. The fully-connected layer is donated as "FC (number of neurons)". The shape of each object is shown within angle brackets, and so is the output shape of each layer

S1. Given a sentence s_i with words w_{it} ($i \in [1, S], t \in [1, W]$), we first use the embedding layer to convert each word index in the sentence into a pre-trained word vector $\boldsymbol{v}_{it}$ of length E through the embedding matrix M_e.

S2. Then, as shown in Fig. 2, we apply the GRU-based block G1 and G2 to the sentence (sequence of pre-trained word vectors) to obtain all left- and right-side context vectors for each word, respectively. The operation processes in G1 and G2 are similar to each other, except the scan directions of their GRU [34] layer: G1 executes a forward scan of sentence (i.e., reads the sentence s_i from words w_{i1} to words w_{iW}), while G2 carries out a backward scan (i.e., reads s_i from w_{iW} to w_{i1}). Given this, we only take G1 as an example to describe the detailed processes of these two blocks in the following part.

S2-1. In block G1, we first apply a forward GRU of size 50 that read the sentence s_i from words w_{i1} to words w_{iW} to get annotations of words by summarizing previous contextual information. On top of the GRU, we apply batch normalization to help achieve stable distribution of activation values throughout training [36], and apply ReLU [66] function as activation function to introduce nonlinearity. At this point, we get a 50-dimensional vector named $\vec{\boldsymbol{u}}_{it}$ as the annotation of word w_{it}.

S2-2. Considering that not all words that occur around a certain word w_{it} contribute equally to the semantics of w_{it}'s context, we introduce the attention mechanism to block G1 by applying an attention layer on top of G1's ReLU activation layer as shown in Fig. 2. Note that the detailed components of this attention layer are not shown in Fig. 2, and it actually includes a fully connected sublayer and a softmax function sublayer, whose detailed working process are as follows: Given a word w_{il} that occurs to the left of the target word w_{it}, we first feed its annotation $\vec{\boldsymbol{u}}_{il}$ (obtained through the lower three layers of G1) into a fully connected sublayer to get a hidden representation $\vec{\boldsymbol{h}}_{il}$ of $\vec{\boldsymbol{u}}_{il}$. Then we calculate the similarity between $\vec{\boldsymbol{h}}_{il}$ and a word level contribution vector $\vec{\boldsymbol{c}}_w$ to measure the importance of word w_{il} and

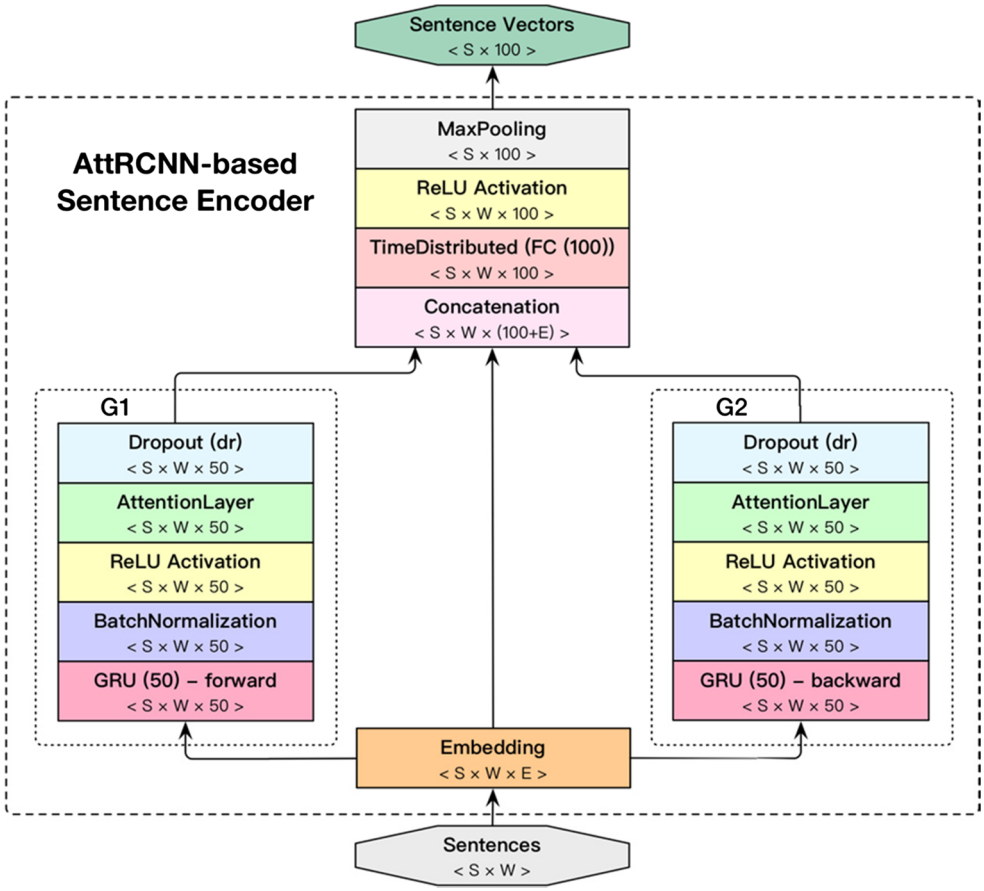

Fig. 2 Structure of the AttRCNN-based Sentence Encoder. The GRU layer and dropout layer are donated as "GRU (number of neurons)-scan direction" and "Dropout (dropout rate)", respectively. The output shape of each layer is shown within angle brackets

obtain a normalized contribution weight alpha $\vec{\boldsymbol{\alpha}}_{il}$ through a softmax function. Then, we calculate the left-side context vector $\boldsymbol{p}_{l-it}$ of word w_{it} as a weighted sum of the word annotations $\vec{\boldsymbol{u}}_{il}$ based on all the contribution weights of w_{it}'s left neighbors. Note that, the contribution vector $\vec{\boldsymbol{c}}_w$ is randomly initialized and jointly learned during the training process. In sum, the attention layer is used to extract informative words that are important to the meaning of w_{it}'s left context and aggregate the representation of those informative words to form the left-context vectors of w_{it}.

S2-3. Besides, to avoid overfitting in the training process, we further apply a dropout layer [67] on top of the attention layer.

S3. Through the GRU-based block G1 and G2, we respectively get the 50-dimensional left- and right-side context vectors for all words in each sentence. After that, we concatenate those context vectors with the pre-trained word embeddings to represent each word as $\left[\boldsymbol{p}_{l-it}, \boldsymbol{v}_{it}, \boldsymbol{p}_{r-it}\right]$, where $\boldsymbol{v}_{it}$ is the pre-trained word vectors, $\boldsymbol{p}_{l-it}$ and $\boldsymbol{p}_{r-it}$ are the left- and right-context vectors of word w_{it}, respectively. At this point, each word is represented as a real-valued vector of length $(50 + E + 50)$.

S4. We further apply a fully connected layer with ReLU activation function on top of the concatenation layer to convert each word vector of length $(50 + E + 50)$ to 100 dimensions, which is exactly the final distributed semantic vector of the word.

Through the processing process from S1 To S4, each word w_{it} of sentence s_i has been transformed by the AttRCNN encoder from the original pre-trained word embedding $\boldsymbol{v}_{it}$ to the final distributed word representation $\boldsymbol{v}'_{it}$ that contains rich semantic information of w_{it}'s left- and right-side neighbor words. Detailed transformation process is shown in Fig. 3, where $\boldsymbol{v}_{il}$ and $\boldsymbol{v}_{ir}$ are the pre-trained word embeddings of w_{it}'s left- and right-side neighbor words. Till now, each sentence is represented as a sequence of its words' distributed semantic vectors of length 100.

S5. Then we continue to apply a max-pooling layer, which uses an element-wise max function to get the max element in each dimension of word representations across all the words in one sentence, so as to capture the most important latent semantic factors.

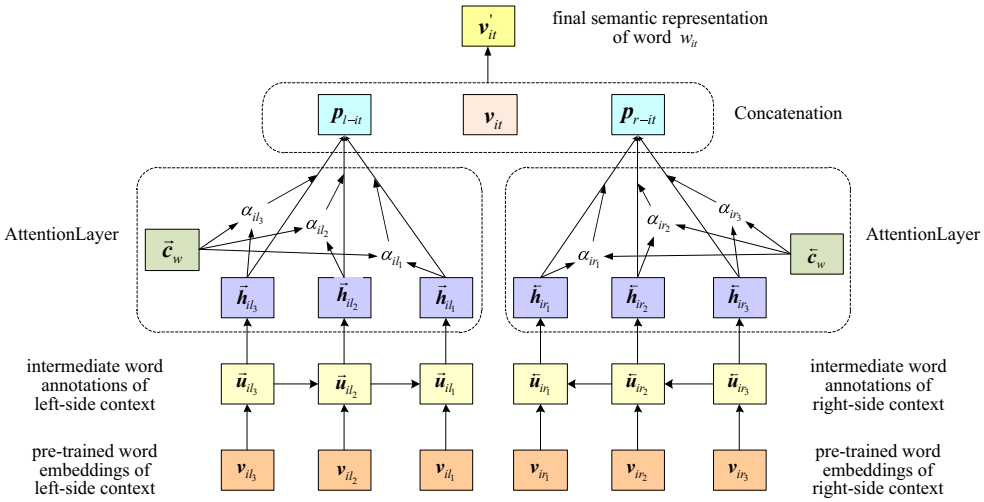

Fig. 3 The process of obtaining the distributed semantic representation of word w_{it} by AttRCNN

Specifically, suppose $\boldsymbol{v}_{s_i}$ as the final representation of sentence s_i, $\boldsymbol{v}'_{it}$ as the annotation of word w_{it} that fed into the max-pooling layer, then the k-th element of s_i's vector $\boldsymbol{v}_{s_i}$ could be calculated following equation (3.2), where $e_k(\boldsymbol{v})$ means the k-th element of vector $\boldsymbol{v}$.

$$e_k(\boldsymbol{v}_{s_i}) = \max_{t=1}^{W}(e_k(\boldsymbol{v}'_{it})) \tag{3.2}$$

Eventually, through the AttRCNN-based sentence encoder, all sentences in a document would be represented as real-valued vectors of length 100.

(2) **Document Vectorization with CNNs**

Users of online social networks tend to publish text posts on social networks to express their feelings and share their daily life experiences. It's common and sometimes inevitable that there would be some emotional continuities and semantic correlations between each user's text posts that have been posted within a relatively short period of time (e.g., one day or a few days), while the semantic dependencies would be not so strong if the time intervals between the "Sent Time" of text posts are large (e.g., several weeks or months). Given this, we think the biased RNN-based architecture that are good at capturing long-term semantic dependencies may not be optimal to be used for document vectorization in our case, while the CNNs, that can efficiently capture local features in a parallel way and then assemble global representations through layer stacking, seem to be more appropriate. Besides, it's well known that increasing the depth and width of network can improve the performance of deep learning model, but the computing budget would increase at the same time. To balance the contradiction between the performance and efficiency, Christian et al. [36, 59, 60] proposed the Inception architecture and achieved good performance in computer vision tasks.

Given the above consideration, we take a variant of the popular CNN-based Inception architecture as our document encoder to learn vector representation from the aggregation of each user's text post vectors extracted through the AttRCNN-based sentence encoder. As shown in Fig. 3, the document encoder of our model consists of four CNN-based blocks (i.e., C1 $\sim$ C4), one concatenation layer and two fully connected layers.

C1 and C4 each contain one convolutional layer that comprises 50 independent filters of size 3 and 5, respectively. C2 and C3 mainly contain two convolutional layers: The first layer comprises 100 convolutional filters of size 1 and size 5, respectively, on top of which we further introduce batch normalization mechanism and ReLU function; the second layer comprises 50 convolutional filters of size 3 and 5, respectively. The stride value of all the convolutional filters in each block is set to be 1, and in the convolution, the input is padded so that the output would be as long as the original input. These four blocks are applied in parallel to the 3-dimensional input of document from $\mathbb{R}^{N\times S\times 100}$, whose elements corresponding to the sentence vectors obtained from the AttRCNN sentence encoder, where N is the number of documents, S is the number of sentences in each document, and 100 is the length of the learned sentence vectors. Through each of these four blocks, the 100-dimensional sentence vector would be converted into 50 dimensions.

Then we concatenate these four kinds of 50-dimensional sentence vectors through a concatenation layer and flatten them to get the preliminary document annotations of length $(S \times 200)$. We further apply two stacked fully connected layers of size 100 and 50 on top of the flatten layer to convert the $(S \times 200)$-dimensional document feature vectors to 50 dimensions. Besides, batch normalization mechanism, ReLU activation function and the dropout technique are still applied here as shown in Fig. 4.

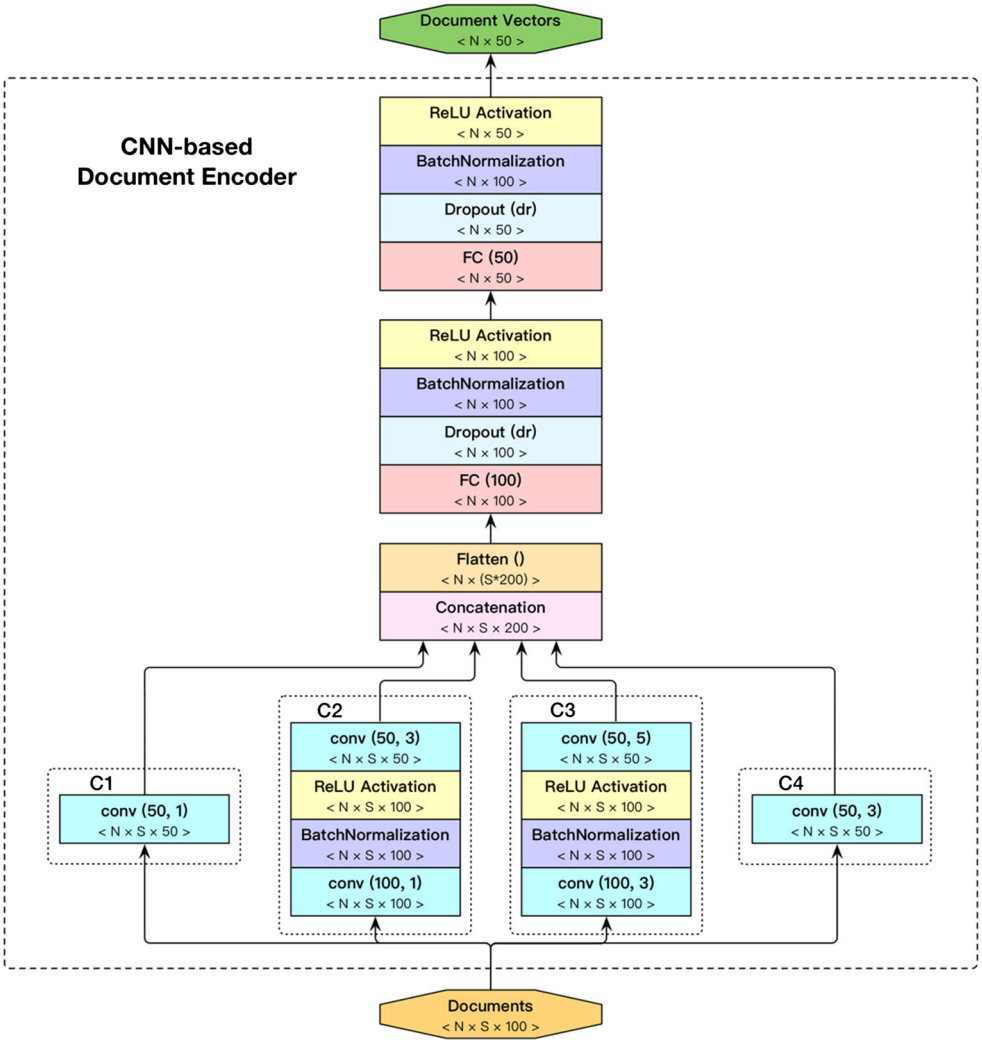

Fig. 4 The structure of the CNN-based Document Encoder. The convolutional layer, fully-connected layer and dropout layer are donated as "conv (number of filters, kernel size)", "FC (number of neurons)" and "Dropout (dropout rate)", respectively. The output shape of each layer is shown within angle brackets

Note that the 50-dimensional output obtained through all the above-mentioned layers from the well-trained model is exactly the target document vectors that we aim to use as the input of the final personality prediction algorithm.

(3) **Model Training**

We adopt mini-batch training approach with batch size B to train five different neural networks with the same architecture for five personality traits. Mean Square Error (MSE) is used as the objective function for training, which can be calculated using (3.3), where n denotes the number of unseen instances, $\left(s_{x_i}^{y_j}\right)^*$ the predicted value for trait y_j,, and $\left(s_{x_i}^{y_j}\right)$ the observed one. The Adam [71] optimizer is adopted to tune the network parameters to minimize the MSE. The training process runs for 30 iterations. We monitored the MSE value of the validation set after each epoch and employ early stopping mechanism (stop training when the monitored MSE has stopped reducing for 4 epochs) to avoid overfitting when training the model. Besides, if the monitored MSE value reduces after a certain epoch, we save current model and overwrite the previously-saved one to guarantee that saved model would be the best model we ever obtained.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\left(s_{x_i}^{y_j}\right)^* - \left(s_{x_i}^{y_j}\right)\right)^2 \tag{3.3}$$

3.4 Prediction

After obtaining the neural network model, we extract the 50-dimensional output of the document encoder as the final deep semantic vector of each document. Then we concatenate it with the dictionary-based features (64 dimensions) and special linguistic features (5 dimensions) which have been pre-extracted directly from user's text posts to construct the ultimate feature space.

In order to get real-valued scores for Big Five personality traits, we adopt regression algorithm as the final prediction algorithm in our approach. Finally, the 119-dimensional

features are fed into the popular gradient boosting regression algorithm to predict the Big Five personality scores.

4 Experiments

4.1 Dataset

To evaluate the effectiveness of the feature vectors extracted from our proposed neural network, we carried out experiments based on the dataset collected as part of the *MyPersonality* project [12], a popular Facebook application allowing users to test their personality via online psychometric tests and donate their scores and Facebook profile data to research. We concentrated on the users whose default language is English and the Big Five personality trait scores are available. The final dataset we utilized involves 115,864 Facebook users, 11,494,862 text posts and 3,055,272 unique word tokens. The average number of text posts per user is 142; the average of the maximum text post length per user is 70. The standard deviation of the text posts number per user is 162.51; the standard deviation of the maximum text posts length per user is 31.22.

According to the statistics of the dataset, we found that the posting habits of different users are quite different. Thus, when constructing the 3-dimensional input array for our neural network, we set the length of each document S as the average number of sentences across all documents, and the length of each sentence W as the average number of words across all sentences. To guarantee that all documents/sentences contain the same number of sentences/words, we padded the shorter documents/sentences with dummy sentences/words and truncate the excess part of the longer documents/sentences. The number of documents N equals to the total number of users. Eventually, we got a 3-dimensional input array of shape$(115864 \times 140 \times 70)$. Note that, we didn't set S to be 142 but rounded the number down by changing the ones digit to zero and set S to be 140 for simplicity.

4.2 Experimental setting

4.2.1 Baseline feature sets

To evaluate the effectiveness of the 50-dimensional feature vectors obtained from our proposed deep neural network, which are named as ARCC, we extracted other four kinds of non-trivial feature set to construct the baseline feature spaces in our experiments.

The first one, code-named Cnn, is the semantic feature set extracted from a CNN-based deep neural network as presented in literature [50]. This is the only work that introduced deep learning technique into personality recognition from texts before ours, and its outperformance over traditional approaches making it the state of the art. Utilizing the 3-dimensional input array with shape $(115864 \times 140 \times 70)$, we built five neural networks with the same structure of that proposed in literature [50] and extracted the document-level features from the model for the prediction of Big Five personality traits.

The second one, code-named RCC, is the deep semantic features extracted from the RCNN-CNNs architecture of hierarchical neural network, which adopts the original RCNN structure as the sentence encoder but the same CNN-based document encoder with our proposed neural network. This baseline feature set is mainly used to evaluate the effectiveness of our modification to RCNN.

The third one, code-named D2V, is another kind of document-level semantic feature vectors extracted through the unsupervised Doc2Vec algorithm [68], an extension of the popular word2vec algorithm that could learn continuous representations for larger blocks of text, such as sentences, paragraphs or entire documents. Taking the aggregation of all the unified text posts in the Facebook dataset as input, we extracted this kind of doc2vec feature vectors of length 50 using the Doc2Vec module provided by Gensim.[4]

The fourth one, code-named SL, is the 69-dimension statistical linguistic feature set extracted following the methods presented in Section 3.2. It consists of the dictionary-based features (64 dimensions) and the special linguistic features (5 dimensions) extracted from the unified text posts and the tokenized text posts, respectively.

Overall, the feature sets can be divided into two categories: ARCC, RCC and Cnn are all extracted from deep neural networks through supervised learning, while D2V and SL are both obtained from unlabeled corpus. To comprehensively evaluate these feature sets and find the optimal feature space for personality recognition, we adopted not only the above-mentioned single feature set but also their combinations as the input of the prediction algorithms. In total, 15 kinds of feature space were evaluated in our experiments, and the detailed list could be found in Table 1.

4.2.2 Regression algorithms

We experimented with four regression settings in total, including multi-layer perceptron (MLP) with one hidden layer, which is trained together with the hierarchical neural network as shown in Fig. 1, and other three commonly-used regression algorithms: support vector regression (SVR) [69], gradient boosting regression (GBR) and random forest

[4] https://radimrehurek.com/gensim/

Table 1 Average mean absolute error (MAE) obtained from personality recognition approaches with different configurations

Prediction Algorithm	Feature set	O	C	E	A	N	Average
SVR	ARCC	**0.3577***	0.4266*	0.4791*	**0.3864***	0.4900*	0.42796*
	ARCC +D2V	0.3580	0.4267	0.4793	0.3869	0.4901	0.42820
	ARCC +SL	0.3623	0.4324	0.4809	0.3922	0.4950	0.43256
	ARCC +D2V +SL	0.3648	0.4341	0.4820	0.3964	0.4957	0.43460
	RCC	0.3785	0.4703	0.4901	0.4159	0.5028	0.45152
	RCC +D2V	0.3785	0.4707	0.4910	0.4166	0.5030	0.45196
	RCC +SL	0.3832	0.4714	0.4932	0.4172	0.5054	0.45408
	RCC +D2V +SL	0.3848	0.4724	0.4944	0.4200	0.5086	0.45604
	Cnn	0.4179	0.5620	0.5999	0.4260	0.6000	0.52116
	Cnn +D2V	0.4179	0.5614	0.5999	0.4263	0.5998	0.52106
	Cnn +SL	0.4341	0.4901	0.5460	0.4585	0.5424	0.49422
	Cnn +D2V +SL	0.4385	0.4930	0.5518	0.4608	0.5470	0.49822
	D2V	0.4090	0.5533	0.5999	0.4290	0.5990	0.51804
	SL	0.4400	0.4963	0.5499	0.4674	0.5510	0.50092
	D2V +SL	0.4355	0.4929	0.5471	0.4626	0.5459	0.49680
GBR	ARCC	0.3618	0.4264	0.4813	0.3902	0.4890	0.42974
	ARCC +D2V	0.3623	0.4268	0.4816	0.3904	0.4893	0.43008
	ARCC +SL	0.3601*	**0.4251***	**0.4776***	0.3882	0.4874	**0.42768***
	ARCC +D2V +SL	0.3605	0.4252	0.4777	0.3878*	**0.4873***	0.42770
	RCC	0.3824	0.4691	0.4907	0.4212	0.5022	0.45312
	RCC +D2V	0.3828	0.4690	0.4909	0.4210	0.5022	0.45318
	RCC +SL	0.3807	0.4649	0.4871	0.4143	0.4991	0.44922
	RCC +D2V +SL	0.3806	0.4647	0.4872	0.4146	0.4987	0.44916
	Cnn	0.4430	0.5363	0.5087	0.4339	0.5170	0.48778
	Cnn +D2V	0.4468	0.5203	0.5262	0.4530	0.5252	0.49430
	Cnn +SL	0.4450	0.4906	0.5509	0.4628	0.5411	0.49808
	Cnn +D2V +SL	0.4457	0.4914	0.5507	0.4627	0.5411	0.49832
	D2V	0.4396	0.5203	0.5286	0.4560	0.5245	0.49380
	SL	0.4455	0.4900	0.5522	0.4619	0.5404	0.49800
	D2V +SL	0.4459	0.4906	0.5527	0.4624	0.5409	0.49850
RF	ARCC	0.3669	0.4330	0.4871	0.3980	0.4946	0.43592
	ARCC +D2V	0.3636	0.4295	0.4871	0.3941	0.4933	0.43352
	ARCC +SL	0.3621*	0.4291	0.4800*	0.3914	0.4904*	0.43060*
	ARCC +D2V +SL	0.3629	0.4284*	0.4803	0.3907*	0.4907	0.43060*
	RCC	0.3896	0.4720	0.4966	0.4281	0.5081	0.45888
	RCC +D2V	0.3879	0.4709	0.4951	0.4255	0.5022	0.45632
	RCC +SL	0.3836	0.4669	0.4898	0.4190	0.5000	0.45186
	RCC +D2V +SL	0.3806	0.4661	0.4903	0.4180	0.5012	0.45124
	Cnn	0.4475	0.5365	0.5087	0.4364	0.5172	0.48926
	Cnn +D2V	0.4717	0.5171	0.5748	0.4856	0.5550	0.52084
	Cnn +SL	0.4482	0.4939	0.5545	0.4669	0.5453	0.50176
	Cnn +D2V +SL	0.4493	0.4939	0.5565	0.4678	0.5476	0.50302
	D2V	0.4287	0.5182	0.5759	0.4868	0.5566	0.51324
	SL	0.4489	0.4932	0.5543	0.4686	0.5459	0.50218
	D2V +SL	0.4494	0.4939	0.5572	0.4676	0.5480	0.50322

Table 1 (continued)

Prediction Algorithm	Feature set	O	C	E	A	N	Average
MLP	ARCC	0.4445*	0.5387*	0.5851*	0.5006*	0.4892*	0.53162*
	RCC	0.5021	0.5810	0.6437	0.5312	0.6329	0.57818
	Cnn	0.5439	0.5970	0.6503	0.5556	0.6363	0.59662

O, C, E, A, N refer to the five dimensions of Big Five traits: Openness to experience, Conscientiousness, Extraversion, Agreeableness, Neuroticism, respectively. In each column, the lowest MAE among all the approaches is typeset in bold, and the lowest MAEs among approaches with the same prediction algorithm are marked by *

(RF) [70], which are trained separately using the pre-extracted feature sets or the combination of them as their input. These regression algorithms were adopted not only to test the predictive ability of different algorithms, but also to figure out whether the performance of each feature set are consistent or not when fed into different prediction algorithms.

4.2.3 Evaluation metrics

In our experiments, the predictive ability of the personality recognition approaches was evaluated by MAE (Mean Absolute Error), a frequently used measure of differences between the predicted score and the observed score tested by Big Five Inventory in APR research. It can be calculated using (4.1), where n denotes the number of unseen instances, $\left(s_{x_i}^{y_j}\right)^*$ the predicted value for trait y_j, and $\left(s_{x_i}^{y_j}\right)$ the observed one. Since MAE is a measure of error, thus, the lower, the better.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\left(s_{x_i}^{y_j}\right)^* - \left(s_{x_i}^{y_j}\right)\right| \tag{4.1}$$

4.3 Methodology

Following the methods presented in Section 3, we built five different neural networks with the same architecture of our proposed model for five personality traits. From each well-trained neural network model, we extracted the 50-dimensional outputs of the document encoder as the final deep semantic vectors of documents. The baseline feature vectors were also extracted following the above presented methods. Then, we separately fed the obtained feature sets or the combinations of them to the regression algorithms to build different prediction models for Big Five traits using scikit-learn,[5] a powerful Python module for machine learning.

We carried out model training and testing experiments with 5-fold cross-validation, and the parameter selection process was nested into the 5-fold cross-validation. In details, we split the whole dataset into 5 equal chunks randomly. Each time three chunks were used as training set, one was used as validation set and the rest one was used as test set. Each model was trained with different parameter settings on the training set, validated on the validation set, and tested on the testing set. The average MAE of each model over a 5-fold cross-validation was recorded, and the parameter setting with the best average performance was selected.

In the case of the neural network models' hyper-parameters, we tried different vector size (namely 50, 100, 150, 200) of the pre-trained word embeddings, dropout rate (varied from 0.1 to 0.9 with step size of 0.1), and batch size (namely 8, 16, 32, 64). The final selected hyper-parameters are as follows: the vector size of the pre-trained word embeddings E was 100; the dropout rate of the dropout layer dr was set to be 0.4; the batch size B was 16. The parameters of the Adam optimizer were set following the original paper [71]. The other hyper-parameters, such as the layer size, are shown in Figs. 1, 2 and 3.

As for the parameters of the regression algorithms, we tried different kernels (namely radial, linear and polynomial) for SVR, different number of estimators for GBR and RF. For each kind of these regressors, the best results were respectively achieved by the SVR learner with the radial kernel, the GBR learner with 100 estimators and the RF learner with 100 trees.

4.4 Results and analysis

The average testing results over 5-fold cross-validation achieved by different approaches with their best parameter settings are presented in Table 1.

Without regard for the MAE differences between approaches with different prediction algorithms, we may find that any personality recognition approach that took the ARCC feature vectors into its input feature set achieves lower prediction errors than the other approaches that didn't involve the ARCC features. In other words, the ARCC feature vectors extracted from our proposed AttRCNN-CNNs neural networks are the most effective ones compared to the other four kinds of baseline features,

[5] http://scikit-learn.org/stable/

including the features extracted from RCNN-CNNs neural networks that code-named RCC, the features extracted from the CNN baseline neural networks code-named Cnn, the features learned from the Doc2Vec algorithm code-named D2V and the statistical linguistic features code-named SL. Ranking after the above mentioned approaches that utilized the ARCC features, the RCC feature-involved approaches come off the second most effective features with overall average MAEs no higher than 0.45888. The outperformance of both the ARCC and the RCC feature set over the Cnn ones (which were also extracted from deep neural network models) demonstrates the advantages of both the overall hierarchical architecture and the document encoder structure of our proposed neural network for text posts modeling in PR. Furthermore, since the ARCC features surpass the RCC ones, we could conclude that our modifications, including introductions of the attention mechanism and batch normalization technique, to RCNN structure laid the foundation for the effectiveness of our methodology.

The bottom three rows in Table 1 show the performance of the end-to-end approaches that utilized a fully-connected layer on top of the document encoder to directly output the predicted personality scores. With average MAEs no lower than 0.53162, these three approaches underperformed all the 45 two-phase approaches whose performance are shown in the top portion of Table 1. It implies that deep neural networks are good at feature extraction; applying separate prediction algorithm rather than fully-connected layers trained together with neural networks could improve the performance of PR approaches. Despite the overall underperformance of the end-to-end approaches, the ARCC feature set learned from our AttRCNN-CNNs model surpass both the RCC and the Cnn ones obtained from the baseline models in all dimensions of the Big Five personality traits, demonstrating the advantages of our proposed text modeling neural network over other two networks.

5 Conclusions and future work

Computational personality recognition is an emerging research field that consists of the automatic inference of users' personality traits from publicly available information on online social platforms. In this paper, we present a two-level hierarchical neural network based on the newly designed AttRCNN structure and a variant of the CNN-based Inception structure to learn the deep semantic representations of online social network users' text posts. Experimental evaluation shows that taking these kinds of deep semantic features as input of traditional regression algorithms contribute a lot to the performance improvement of Big Five personality recognition approaches. In future work, we will utilize these kind of deep semantic features as the input of some special designed regression algorithms so as to further improve the prediction accuracy of the personality recognition approaches.

Acknowledgements This work was support by the scientific research funds of PLA (Grant No. AWS13J003). The authors would like to thank the anonymous reviewers for their careful review and constructive comments. Thanks also to David Stillwell, Michal Kosinski and the myPersonality project for their efforts on collecting the Facebook dataset.

References

1. Vinciarelli A, Mohammadi G (2014) A survey of personality computing. IEEE Trans Affect Comput 5(3):273–291. https://doi.org/10.1109/taffc.2014.2330816
2. Funder DC (2001) Personality. Annu Rev Psychol 52:197–221
3. Allport GW (1937) Personality: a psychological interpretation. Henry Holt, New York
4. Xue D, Hong Z, Guo S, Gao L, Wu L, Zheng J, Zhao N (2017) Personality recognition on social media with label distribution learning. IEEE Access 5:13478–13488
5. Celli F, Lepri B, Biel JI, Gatica-Perez D, Riccardi G, Pianesi F (2014) The workshop on computational personality recognition 2014. In: ACM conference on multimedia, Orlando, November 3–7, 2014. ACM, pp 1245–1246
6. Zhang L, Huang XL, Liu TL, Li A, Chen ZX, Zhu TS (2014) Using linguistic features to estimate suicide probability of Chinese microblog users. In: Human centered computing. Springer, pp 549–559
7. Celli F, Rossi L (2012) The role of emotional stability in Twitter conversations. In: Workshop on semantic analysis in social media, Avignon, 23–27 April 2012. Association for Computational Linguistics, pp 10–17
8. Roshchina A, Cardiff J, Rosso PA (2011) Comparative evaluation of personality estimation algorithms for the twin recommender system. In: 3th international workshop on search and mining user-generated contents, Glasgow, October 28 2011. ACM, pp 11–17
9. Enos F, Benus S, Cautin RL, Graciarena M, Hirschberg J, Shriberg E (2006) Personality factors in human deception detection: comparing human to machine performance. In: INTERSPEECH 2006 and 9th international conference on spoken language processing, Pittsburgh, Pennsylvania, 17–21 September. DUMMY PUBID
10. Luyckx K, Daelemans W (2008) Personae: a corpus for author and personality prediction from text. In: 6th international conference on language resources and evaluation, Marrakech, 28–30 May 2008, pp 2981–2987
11. Golbeck J, Hansen D (2011) Computing political preference among twitter followers. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 1105–1108
12. Celli F, Pianesi F, Stillwell D, Kosinski M (2013) Workshop on computational personality recognition: shared task. In: 7th international AAAI conference on weblogs and social media. Boston, Jul 8–11
13. Nie D, Guan ZD, Hao BB, Bai ST, Zhu TS (2014) Predicting personality on social media with semi-supervised learning. In: IEEE/WIC/ACM international joint conferences on web intelligence and intelligent agent technologies -volume 02, Warsaw, August 11–14, 2014. IEEE Computer Society, pp 158–165

14. Farnadi G, Sitaraman G, Sushmita S, Celli F, Kosinski M, Stillwell D, Davalos S, Moens MF, De Cock M (2016) Computational personality recognition in social media. User Model User-Adap Int 26(2–3):109–142. https://doi.org/10.1007/s11257-016-9171-0
15. Schwartz HA, Eichstaedt JC, Kern ML, Dziurzynski L, Ramones SM, Agrawal M, Shah A, Kosinski M, Stillwell D, Seligman ME (2013) Personality, gender, and age in the language of social media: the open-vocabulary approach. PloS ONE 8(9):e73791
16. Polonsky M (2007) Online social networks and insights into marketing communications. J Internet Commer 6(4):55–72
17. Rosen PA, Kluemper DH (2008) The impact of the Big Five personality traits on the acceptance of social networking website. Paper presented at the 14th Americas Conference on Information Systems, Toronto
18. Schrammel J, Ffel C, Tscheligi M (2009) Personality traits, usage patterns and information disclosure in online communities. In: 23rd annual conference on human computer interaction, HCI 2009, Cambridge, UK, September 01–05, 2009. BCS Learning & Development Ltd, Swindon, pp 169–174
19. Selfhout M, Burk W, Branje S, Denissen J, Aken MV, Meeus W (2010) Emerging late adolescent friendship networks and Big Five personality traits: a social network approach. J Pers 78(2):509–538
20. Li A, Yan Z, Zhu TS (2013) Self-report versus web-log: which one is better to predict personality of website users? Int J Cyber Behav Psychol Learn 3(4):44–54
21. Kosinski M, Stillwell D, Graepel T (2013) Private traits and attributes are predictable from digital records of human behavior. Proc Natl Acad Sci 110(15):5802–5805
22. Gosling SD, Augustine AA, Vazire S, Holtzman N, Gaddis S (2011) Manifestations of personality in online social networks: self-reported Facebook-related behaviors and observable profile information. Cyberpsychol Behav Soc Netw 14(9):483–488
23. Golbeck J, Robles C, Turner K (2011) Predicting personality with social media. In: CHI'11 extended abstracts on human factors in computing systems, Vancouver, May 7–12, 2011. ACM, pp 253–262
24. Golbeck J, Robles C, Edmondson M, Turner K (2011) Predicting personality from Twitter. In: 2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing, Boston, 9–11 Oct. 2011. IEEE, pp 149–156
25. Quercia D, Kosinski M, Stillwell D, Crowcroft J (2011) Our Twitter profiles, our selves: predicting personality with twitter. In: 2011 IEEE international conference on privacy, security, risk and trust and 2011 IEEE international conference on social computing, Boston, October 9–11, 2011. IEEE, pp 180–185
26. Farnadi G, Zoghbi S, Moens MF, De Cock M (2013) Recognising personality traits using Facebook status updates. In: Workshop on computational personality recognition at the 7th international AAAI conference on weblogs and social media, Boston, Massachusetts, July 8–11 2013. AAAI, pp 14–18
27. Alam F, Stepanov EA, Riccardi G (2013) Personality traits recognition on social network-Facebook. In: International conference on weblogs and social media, Cambridge, July 11 2013. AI Access Foundation, pp 6–9
28. Skowron M, Ferwerda B, Tkalčič M, Schedl M (2016) Fusing social media cues: personality prediction from Twitter and Instagram. In: 25th international conference companion on world Wide Web, April 11–15 2016, Montreal. ACM, pp 107–108
29. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
30. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
31. Sridhar VKR (2015) Unsupervised text normalization using distributed representations of words and phrases. In: VS@ HLT-NAACL, pp 8–16
32. Poria S, Cambria E, Hazarika D, Vij P (2016) A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv:161008815
33. Poria S, Cambria E, Gelbukh A (2016) Aspect extraction for opinion mining with a deep convolutional neural network. Knowl-Based Syst 108:42–49
34. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv:14090473
35. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015) Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning, pp 2048–2057
36. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp 448–456
37. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: AAAI, pp 2267–2273
38. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232
39. Bai ST, Zhu TS, Cheng L (2012) Big-five personality prediction based on user behaviors at social network sites. arXiv:12044809
40. Bai ST, Hao BB, Li A, Yuan S, Gao R, Zhu TS (2013) Predicting big five personality traits of Microblog users. In: IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, pp 501–508. https://doi.org/10.1109/1109/wi-iat.2013.70
41. Nie D, Li L, Zhu TS (2013) Conscientiousness measurement from Weibo's public information. In: Zhou Z-H, Schwenker F (eds) Partially supervised learning: second IAPR international workshop, PSL 2013, Nanjing, China, May 13–14, 2013, Revised Selected Papers. Springer, Berlin, pp 58–67, https://doi.org/10.1007/978-3-642-40705-5_6
42. Li L, Li A, Hao BB, Guan ZD, Zhu TS, Liu C (2014) Predicting active users' personality based on micro-blogging behaviors. PLoS ONE 9(1):e84997
43. Peng KH, Liou LH, Chang CS, Lee DS (2015) Predicting personality traits of Chinese users based on Facebook wall posts. In: 24th wireless and optical communication conference, Taipei, 23–24 Oct. 2015. IEEE, pp 9–14, https://doi.org/10.1109/WOCC.2015.7346106
44. Shlomo A, Moshe K, Dhawle S (2005) Pennebaker JW Lexical predictors of personality type. In: Joint annual meeting of the interface and the classification society of North America, St. Louis, 8–12 June
45. Mairesse F, Walker MA, Mehl MR, Moore RK (2007) Using linguistic cues for the automatic recognition of personality in conversation and text. J Artif Intell Res 30:457–500
46. Pennebaker JW, Booth RJ, Francis ME (2007) Linguistic inquiry and word count: LIWC [Computer software]. LIWC Net, Austin
47. Coltheart M (1981) The MRC psycholinguistic database. Q J Exp Psychol 33(4):497–505
48. Oberlander J (2006) Nowson S Whose thumb is it anyway? Classifying author personality from weblog text. In: COLING/ACL on main conference poster sessions, Sydney, Australia, July 17–18, 2006. Association for Computational Linguistics Stroudsburg, pp 627–634
49. Nowson S, Oberlander J (2007) Identifying more bloggers: towards large scale personality classification of personal weblogs. In: International conference on weblogs and social media, Boulder, Colorado, USA, March 26–28 2007. AAAI Press

50. Majumder N, Poria S, Gelbukh A, Cambria E (2017) Deep learning-based document modeling for personality detection from text. IEEE Intell Syst 32(2):74–79
51. Rangel F, Rosso P, Potthast M, Stein B, Daelemans W (2015) Overview of the 3rd author profiling task at PAN 2015. In: Conference and labs of the evaluation forum, Toulouse, September 8–11 2015. CEUR-WS.org
52. Yang Z, Yang D, Dyer C, He X, Smola AJ, Hovy EH (2016) Hierarchical attention networks for document classification. In: HLT-NAACL, pp 1480–1489
53. Glorot X, Bordes A, Bengio Y (2011) Domain adaptation for large-scale sentiment classification: a deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 513–520
54. LeCun Y, Boser BE, Denker JS, Henderson D, Howard RE, Hubbard WE, Jackel LD (1990) Handwritten digit recognition with a back-propagation network. In: Advances in neural information processing systems, pp 396–404
55. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
56. Elman JL (1990) Finding structure in time. Cognit Sci 14(2):179–211
57. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
58. Zhou C, Sun C, Liu Z, Lau F (2015) A C-LSTM neural network for text classification. arXiv:151108630
59. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
60. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-ResNet and the impact of residual connections on learning. In: AAAI, pp 4278–4284
61. Pennebaker JW, King LA (1999) Linguistic styles: language use as an individual difference. J Pers Soc Psychol 77(6): 1296
62. Tausczik YR, Pennebaker JW (2010) The psychological meaning of words: LIWC and computerized text analysis methods. J Lang Soc Psychol 29(1):24–54
63. Harris ZS (1954) Distributional structure. Word 10(2–3):146–162
64. Collobert R, Weston J (2007) Fast semantic extraction using a novel neural network architecture. In: Annual meeting-association for computational linguistics, vol 1, p 560
65. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv:13013781
66. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 315–323
67. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
68. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on international conference on machine learning, pp II–1188
69. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222
70. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
71. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. arXiv:14126980