Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4532–4539
Marseille, 20-25 June 2022

Incorporating LIWC in Neural Networks to Improve Human Trait and Behavior Analysis in Low Resource Scenarios

Isıl Doğa Yakut Kılıç [1], **Shimei Pan**[1]
[1]University of Maryland, Baltimore County
{yakut1, shimei}@umbc.edu

Abstract

Psycholinguistic knowledge resources have been widely used in constructing features for text-based human trait and behavior analysis. Recently, deep neural network (NN)-based text analysis methods have gained dominance due to their high prediction performance. However, NN-based methods may not perform well in low resource scenarios where the ground truth data is limited (e.g., only a few hundred labeled training instances are available). In this research, we investigate diverse methods to incorporate Linguistic Inquiry and Word Count (LIWC), a widely-used psycholinguistic lexicon, in NN models to improve human trait and behavior analysis in low resource scenarios. We evaluate the proposed methods in two tasks: predicting delay discounting and predicting drug use based on social media posts. The results demonstrate that our methods perform significantly better than baselines that use only LIWC or only NN-based feature learning methods. They also performed significantly better than published results on the same dataset.

Keywords: Low-resource Learning Tasks, Domain Knowledge Incorporation, Deep Neural Networks

1. Introduction

Psycholinguistics research has shown that language use provides clues into one's emotional and cognitive states (Pennebaker et al., 2003; Pennebaker and King, 1999). Text-based human trait and behavior analysis aims at finding patterns in user-generated texts to help analyze the characteristics and behavior of the author such as their beliefs, attitudes, emotions, personality, and online/real world behaviors. Previously, psycholinguistic lexicons such as Linguistic Inquiry and Word Count (LIWC) have been widely used in predicting human traits and behavior from text (Riff et al., 2019; Robinson et al., 2013; Yarkoni, 2010). Recently, fueled by big data and big machines, deep neural network (NN)-based models have gained near dominance due to their high prediction performance (Majumder et al., 2017a; Ding et al., 2017). However, NN-based methods may not work well when the training data is limited (e.g., with only a few hundred labeled training examples). Such a low resource scenario is common in human trait and behavior analysis because sophisticated psychometric evaluations are often required to obtain the ground truth (Zafarani and Liu, 2015). As a result, it is expensive to acquire the grount truth at a large scale.

When the ground truth data is limited, the power of deep learning can also be limited. Since high-level abstractions related to human emotions and cognitions may not be easily inferred from a small number of examples, in such a low resource scenario, psycholinguistic knowledge may complement the information extracted automatically by NNs. In this research, we investigate diverse methods to incorporate Linguistic Inquiry and Word Count (LIWC), a widely used psycholinguistic lexicon, in neural network(NN)-based human trait and behavior analysis. Many state of the art NN-based models employ a self-supervised user embedding process to take advantage of a large amount of unsupervised text (Pan and Ding, 2019). In this research, we adopt a similar architecture that first learns a concise and application-agnostic user representation/embeddings from unsupervised text (e.g., unlabeled raw social media text). Then we use supervised machine learning to predict the target user trait and behavior (e.g., drug use) from the learned user representation/embedding. Figure 1 shows the reference architecture employed in this study.

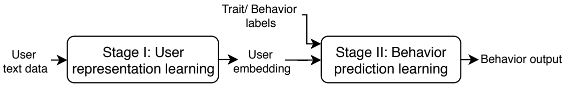

Trait/ Behavior labels
User text data
Stage I: User representation learning
User embedding
Stage II: Behavior prediction learning
Behavior output

Figure 1: The Reference System Architecture

The main contributions of this research include:

- We propose a diverse set of new methods for incorporating LIWC, a general-purpose psycholinguistic lexicon in NN-based human trait and behavior analysis. Our methods significantly outperform traditional and base neural methods in low resource scenarios.

- Our investigation also sheds new light on the impact of various design choices such as LIWC injection stage (e.g., during self-supervised user embedding or behavior prediction), goal or purpose of injection (e.g., to augment input or regularize output) and the fusion or parameter sharing strategy (e.g., early fusion or later fusion, parameter freeze or no parameter freeze) on system performance in low resource settings.

- We evaluate the effectiveness of our methods in two complex human trait and behavior analysis tasks: predicting delay discounting and predicting drug use from social media posts.

2. Related Work

In this section, we first survey the work on text-based human trait and behavior analysis with a specific focus on combining LIWC with NNs. We then extend the scope to include general methods on incorporating lexicons in NN-based models.

Early work on text-based human trait and behavior modeling focuses on using basic text features such as ngrams (Saleem et al., 2012; Iacobelli et al., 2011), psycholingustics features dervied from LIWC or sentiment lexcons (Saleem et al., 2012; Golbeck et al., 2011; Mairesse et al., 2007) and latent text features derived by LDA, SVD and factor analysis (Schwartz et al., 2013; Kulkarni et al., 2018). Recently, there is a surge of interests in nerual network-based models. For example, (Majumder et al., 2017b) employs Convolutional Neural Networks (CNN) to predict big5 personality from text. In addition, (Ding et al., 2017) employs neural network-based document embedding and multiview fusion to predict substance use from social media posts and likes. There is limited existing work on combining LIWC with NNs for human behavior analysis (Trotzek et al., 2018; Majumder et al., 2017b; Yang et al., 2021). Most of them simply concatenate LIWC features with other text features (e.g., word embeddings). No systematical investigation has been conducted to identify the most effective methods of integrating LIWC in NN models, under low resource scenarios.

Previous work on incorporating lexicons to neural network-based models can be grouped into two categories: (1) adding lexical knowledge to improve task agnostic- representation learning (e.g., to enhance word embedding) (2) adding lexical knowledge to improve task prediction. For (1), since some of recent word embedding models such as BERT are very expensive to retrain under low recourse scenarios, most existing work on incorporating lexicons into word embedding is based on word2vec (Mikolov et al., 2013). Among them, some change the objective of word embedding training to include lexical knowledge (Bian et al., 2014; Yu and Dredze, 2014). Others adapt word embeddings in a post-processing step via rretrofitting (Faruqui et al., 2014; Jo and Choi, 2018; Bian et al., 2014). Unlike these methods which try to improve the representation of a word with additional lexical knowledge, we focus on incorporating LIWC in learning a user embedding. Most work on incorporating lexical knowledge to improve the final task is for sentiment analysis (Shin et al., 2016; Teng et al., 2016; Nguyen and Nguyen, 2017). Lexical knowledge was added through either concatenation to word vectors or weighing word vectors by their corresponding sentiment scores.

3. Design Choices

In this section, we first provide an overview of LIWC, followed by a few design considerations.

LIWC

Previous psycholinguistics research has shown that language use provides clues into one's emotional and cognitive states (Pennebaker et al., 2003; Pennebaker and King, 1999). To capture the relationship between word usage and the author's cognitive and mental states, psycholinguists developed a text analysis tool called LIWC (Pennebaker and King, 1999; Pennebaker et al., 2015) which assigns words into psychologically or linguistically meaningful categories including psychological constructs (e.g., cognition, biological processes), personal concern (e.g., work, home, leisure), linguistic properties (e.g., pronoun, punctuations and word complexity) among various others. Given a text document, LIWC outputs a score for each of the LIWC categories. The resulting vector of 71 real numbers can then be used as a representation of the characteristics of the author.

Choices

In this study, we systematically explore methods listed in Table 1 based on different design choices.

Learning stage - Text-based user behavior prediction can be divided into two stages: (1) application-agnostic user embedding or representation learning and (2) predicting the target behavior using a supervised approach. We consider three options for incorporating LIWC: in user embedding learning (*embd*), during target behavior prediction (*behv*) and in both stages simultaneously.

Purpose - We further divide the above choices based on whether LIWC information is used to augment the input (e.g., adapting the text used in embedding) or regularize the embedding learning process (e.g., add LIWC score prediction as a auxiliary task in multi-task learning to regularize user embedding *Embd-Obj* or target behavior prediction *Behv-Obj*).

Fusion - We explore multiple design options when LIWC is fused with NNs. For incorporation LIWC in user embeddings, we explore two options: masking or replacing words (*Embd-Repl*) and annotating words (*Embd-Anno*). For incorporating LIWC during behavior prediction, we explore three options: LIWC and user embedding feature are concatenated in input layer (*Behv-Early*), concatenation of user embedding and LIWC occurs in later layers in the network (parallel networks concatenated later) (*Behv-Late-NoFreeze*), training of the parallel networks are isolated by freezing one when training the other (*Behv-Late-Freeze*).

4. Proposed Methods

In the following, we provide details on the proposed methods. We group them based on which stage LIWC is injected: during user embedding, target behavior prediction or both.

Methods	Design choices		
	Stage	**Purpose**	**Fusion Methods**
Embd-Repl	user embedding	augment input	replace words with LIWC categories
Embd-Anno	user embedding	augment input and regularize output	annotate words with LIWC categories and change embedding objective
Embd-Obj	user embedding	regualize output	change the embedding objective
Behv-Early	behavior prediction	augment input	early fusion
Behv-Late-Nofreeze	behavior prediction	augment input	late fusion
Behv-Late-Freeze	behavior prediction	augment input	late fusion with freeze to miminze parameter sharing
Behv-Obj	behavior prediction	regularize output	change behavior prediction objective

Table 1: Mapping between design choices and resulting methodology

4.1. Incorporating LIWC in user embedding

We have explored three different methods to incorporate LIWC in user embedding: (1) LIWC-based word replacement (*Embd-Repl*), (2) LIWC-based word annotation (*Embd-Anno*), (3) LIWC-based regualization objective (*Embd-Obj*).

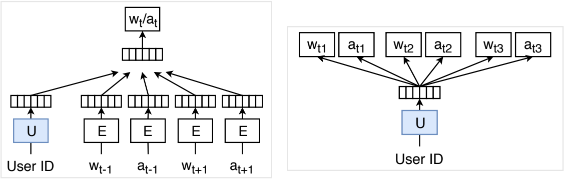
4.1.1. *Embd-Repl*

This method simply replaces word tokens in text with their LIWC categories. If a word belongs to multiple LIWC categories, then the replacement is a concatenation of the categories. For example, ”Adore” is mapped to a combination of two LIWC categories ”Affect&PositiveEmotion”. If a word is out of the LIWC lexicon, then ”None” is used as the replacement. Since we replace a specific word with its LIWC categories, this may help our system to generalize and recognize abstract patterns related to human emotions and cognitions. Moreover, user embeddings learned from this input not only capture the distributions of LIWC categories in a text, but also their sequential dependency.

4.1.2. *Embd-Anno*

In the previous method, after LIWC-based replacement, we lose the information of the original words. Here instead of replacement, we augment the input text with LIWC categories. For each word in a text file, we add its LIWC categories as word annotations.

To learn a user embedding that takes advantage of the LIWC annotations, we developed two new algorithms: LIWC-enhanced Distributed Memory (L-DM) and LIWC-enhanced Distributed Bag of Word (L-DBOW), Their architectures are illustrated in Figures 2a and 2b. For L-DM, we modify the objective of the algorithm to predict the words w_t and their annotation a_t given a document vector U and some context words (e.g.,w_{t-1}) and their annotations (e.g., a_{t-1}) . For L-DBOW, we modify the objective of the algorithm to predict the words w_{ti} and their annotations a_{ti}, given a document vector U. More formally, the objective function of L-DM is to maximize the average log probability of:



(a) (b)

Figure 2: Embd-Anno Architectures. (a) Embd-Anno with L-DM and (b) Embd-Anno with L-BOW

$$\frac{1}{N}\sum_{i=0}^{N}\frac{1}{T}\sum_{t=1}^{T}\sum_{j=-C}^{j=C}\log p(w_t/a_t|w_{t+j},a_{t+j},u_i) \quad (1)$$

and for L-DBOW the objective is to maximize the average log probability of:

$$\frac{1}{N}\sum_{i=0}^{N}\frac{1}{T}\sum_{t=1}^{T}\sum_{j=-C}^{j=C}\log p(w_{t+j},a_{t+j}|u_i) \quad (2)$$

where u_i is a user embedding, N is the number of documents, T is the length of a document, w_t and a_t are a target word and its corresponding annotation, and C is the window size.

4.1.3. *Embd-Obj*

Unlike the previous two methods where LIWC information is injected at the word level, for this method, we use document-level normalized word counts of LIWC categories as an additional training objective to regularize the NN-based user embedding process. Specifically, after learning user embeddings, we further use the user embedding matrix to predict LIWC scores and the original embeddings simultaneously. We measure the similarity between vectors using mean squared error. The optimization objective is to minimize the following function:

$$\frac{1}{N}\sum_{i=0}^{N}\alpha(\hat{u_i}-u_i)^2+(1-\alpha)(l_i-h_{LIWC}(u_i))^2 \quad (3)$$

where $\hat{u_i}$ is a pre-trained user embedding, u_i is a modified user embedding, h is a non-linear transformation and l_i is calculated LIWC scores. α is the weight to control the influence of the original embedding vs predicted LIWC scores.

4.2. Incorporating LIWC in target prediction

We also explore various methods that incorporate LIWC in the target prediction stage. Here, we are evaluating four different models: (1) early concatenation (*Behv-Early*), (2) late fusion with no freeze (*Behv-Late-NoFreeze*) (3) late fusion with freeze (*Behv-Late-Freeze*) and (4) LIWC that regularizes target behavior prediction via modified optimization objective (*Behv-Obj*). For each of these models, to isolate the effect of injecting LIWC at the target prediction stage, we restrict user representations to pre-trained user embeddings without LIWC enhancement.

4.2.1. *Behv-Early*

The early fusion model (*Behv-Early*) combines the user embedding features with LIWC features at the input layer. The concatenated features are then used as the input to train a target prediction model (e.g., predicting drug use). This is the simplest model to combine user embedding and LIWC features. The same set of network parameters are used to control both the mapping between user embedding to the target behavior and the LIWC scores to the target behavior.

4.2.2. *Behv-Late-NoFreeze*

Two parallel sub-NNs are used in *Behv-Late-NoFreeze*, one takes user embeddings as the input and the other takes LIWC scores as the input. The last layer of each sub-NN will be fused together right before the last classification layer. Comparing with *Behv-Early*, this model uses two independent sets of model parameters, one to mainly capture the relations between the target behavior and user embeddings, the other to capture that and LIWC scores. But the separation of influence between the parameters of these two sub-NNs is not clean or fused due to the fact that the backpropagation is controlled by the parameters of both networks. There is some cross sub-NN influence between the two sets of sub-NN parameters.

4.2.3. *Behv-Late-Freeze*

Similar to *Behv-Late-NoFreeze*, two parallel sub-NNs are used in this model. Unlike *Behv-Late-NoFreeze*, here each NN is trained to predict the target independently. The two NNs are then frozen first before they are fused for the final prediction. Comparing *Behv-Late-Freeze* with *Behv-Late-NoFreeze*, due to parameter freeze, the first set of parameters in the first sub-NN mainly captures the relations between user embeddings and the target behavior. The parameters in the second NN capture the relations between the target behavior and LIWC scores. There is no cross sub-NN influence in this model.

4.2.4. *Behv-Obj*

In this model (*Behv-Obj*), we formulate this as a multi-task learning problem. In addition to predicting the target behavior, the network also predicts LIWC scores. Since the prediction tasks share the same set of parameters, some of the knowledge learned during LIWC prediction may transfer and benefit the prediction of the target behavior. The objective is then to minimize the binary cross-entropy loss of the behavior classification as well as LIWC prediction:

$$\frac{1}{N}\sum_{i=0}^{N}\alpha(-(y_i log(h(u_i))+ \\ (1-y_i)log(1-h(u_i)))) \\ +(1-\alpha)(l_i - h_{LIWC}(u_i))^2 \quad (4)$$

where y_i is the ground truth label for a user, u_i is the user embedding, h and h_{LIWC} are non-linear transformations and l_i is calculated LIWC scores. The α parameter are weight values that control the influence of the original embedding vs predicted LIWC score.

4.3. Incorporating LIWC in both stages

We explore one additional method to observe the effect of combining LIWC in both stages simultaneously: user embedding learning and target behavior prediction. For this purpose, we integrate the best performing method within each stage: Embd-Anno and Behv-Late-Freeze.

5. Experiment Settings

We evaluate the methods in two downstream tasks: (1) predicting Delay Discounting (DD) and (2) predicting Drug Use (DU) based on one's social media posts.

Delay discounting, a behavioral measure of impulsivity, is often used to quantify the human tendency to choose a smaller, sooner reward (e.g., $1 today) over a larger, later reward ($2 tomorrow). Previous research has shown that steep DD is often linked problematic behaviors such as alcohol and drug abuse, pathological gambling and credit card default (Alessi and Petry, 2003; Kirby et al., 1999). However, it is quite difficult to acquire the ground truth delay discounting score of a person as it requires sophisticatd psychometric evaluation via lengthy questionnairs (Stillwell and Tunney, 2012). To facilitate comparision, we adopt the same DD threshold used in (Ding et al., 2019) to classify users into either (1) "HIGHlY IMPULSIVE" or (2) "OTHER". Moreover, substance abuse is a serious public health problem impacting many communities in our society. In our second task, we predict problematic drug use behavior from social media posts. The system is designed to be used as a scalable screening tool for substance use disorders. The DU ground truth was collected via a self-reported survey. Based on the frequency of a person's drug use, we categorize a user as either a (1) "DAILY DRUG USER" or (2) "OTHER".

User representation methods	AUC scores DU	DD
AV-W2V	0.537	0.528
AV-DM	0.600	0.592
All-DM	0.616	0.603
All-DBOW	0.644	0.643
AV-LSTM	0.593	0.577
LIWC scores	0.756	0.602
Ding et al. 2019	N.A.	0.641

Table 2: Performances of baseline models on drug use (DU) and delayed discounting (DD) prediction.

Both the DD and the substance use (DU) datasets are collected from Facebook users as a part of the myPersonality projects (Kosinski et al., 2015). The data were gathered with an explicit opt-in consent for reuse for research purposes. To protect privacy, the data were anonymized before the owners shared the data with us. In total, our datasets contain 859 and 817 users with the ground truth DD and DU annotations respectively. In addition, We also have access to a large annotated Facebook datasets with the social media posts of 100k users. This unannotated data is used to train word embeddigns.

Evalaution Metrics

Given that our training dataset size is small ($n < 1k$) and is imbalanced (about 10% positive cases), we randomly split it 80/10/10 into a training, a development and a test dataset. We repeat the process ten time and report the average results. We use weighted ROC AUC as the evaluation metric to address dataset size and class imbalance.

Baselines

We include a total of seven baselines. Among them (1)AV-W2V uses the average of word embeddings as the user representation, where word embeddings are learned using word2vec(Mikolov et al., 2013) (2) AV-DM uses the average of document embeddings as the representation of a user where each post is a document and document embeddings are learned using the Distributed Memory model (DM) in Doc2Vec (Le and Mikolov, 2014) (AV-DM), (3 and 4): ALL-DM and ALL-DBOW: train one document embedding with the Distribute Memory (DM) or the Distributed Bag of Word (DBOW) model in Doc2Vec (Le and Mikolov, 2014) for each person where all of a user's posts is a single document, (5)AV-LSTM: learn the representation of a sentence using LSTM (Hochreiter and Schmidhuber, 1997) and then average all the sentences by a person as the user presentation. (6)LIWC: we use the LIWC scores derived from all the texts authored by a person as the user representation. In addition, we also include the only known results on the same dataset on DD prediction by (Ding et al., 2019) which use 10-fold CV unlike our 10x random 80/10/10 split.

Parameter Tuning

For word2vec, DM and DBOW, we tune two hyperparameters: embedding dimensions and windows size. We varied the embedding dimension values from 50, 100, 300, to 500 and window size from 3, 5, to 8. We use grid search to find the best model parameters.

For *Embd-Anno*, we double the context window size to accomodate additional LIWC annotations. For *Embd-Obj*, we fine turn the tradeoff parameter α in Equation 3 empirially based on the development dataset.

All the Behv-* models employ different neural network architectures. For the first 3 models (Behv-Early, Behv-Late-Nofreeze and Behv-Late-Freez), the neural networks have multiple fully connected layers. The number of layers and the number of neurons in each layer were empirically determined via grid search. For Behv-Obj, the tradeoff parameter α in the objective function is empirically determined based on the development dataset.

For our final behavior classification, we trained a shallow neural network with a single fully connected layer between the input and output layers. The number of neurons in the hidden layer was determined using grid search with values ranging from 16 to 120. The activation function for the mid-layer was ReLu (Glorot et al., 2011). The output layer contained a single unit with a sigmoid activation function. All models used AdaGrad, a version of the gradient descent algorithm where the learning rate is adaptive. For AdaGrad, the initial learning rate hyperparameter was empirically determined using grid search with values between 10^-1 and 10^-5.

6. Experiments Results

In this section, first, we present the baseline model performance and move on to the performance of the proposed methods, grouped based on the stage where LIWC is incorporated.

6.1. Performance of Baseline Models

Table 2 shows the performance of the baselines. Among the baselines we implemented, LIWC performed the best for drug use (DU) prediction (.756 AUC) and third best for DD (.602 AUC). For DU, it outperformed all the NN basic models by a significant margin. Among the basic NN models we tested, All-DM and All-DBOW model performed the best; for DU prediction .644 and .616 AUC respectively and for DD prediction .643 and .603 respectively. This result demonstrates that (1) LIWC is effective, in low resource scenarios (2) NN-based models perform poorly with limited training data. These results are unsurprising. In the following, we investigate whether the knowledge learned by NNs complements LWIC and thus still can be useful in low resource scenarios.

6.2. Performance of Embd-* Models

The performance of Embd-* methods are listed in Table 3. We report the resutls on LIWC-based perfor-

Base embedding	Method	DU AUC	DU Pval	DD AUC	DD Pval
All-DBOW	-	0.644	-	0.643	-
All-DM	-	0.616	-	0.603	-
All-DBOW	Embd-Repl	0.820	6.25e-14	0.806	6.58e-12
All-DBOW	Embd-Anno	**0.879**	1.52e-13	**0.833**	4.84e-13
All-DBOW	Embd-Obj	0.687	3.23e-05	0.617	2.04e-03
All-DM	Embd-Repl	0.794	1.28e-13	0.784	2.94e-13
All-DM	Embd-Anno	0.843	1.81e-14	0.819	2.55e-14
All-DM	Embd-Obj	0.642	0.021	0.592	0.290
All-DBOW	Behv-Early	0.744	5.14e-08	0.682	6.99e-04
All-DBOW	Behv-Late-NoFreeze	0.731	3.36e-07	0.701	8.65e-07
All-DBOW	Behv-Late-Freeze	0.765	6.70e-12	0.721	4.66e-06
All-DBOW	Behv-Obj	0.679	3.58e-03	0.609	1.77e-03
All-DM	Behv-Early	0.740	5.33e-11	0.706	4.21e-09
All-DM	Behv-Late-NoFreeze	0.726	3.43e-10	0.699	5.25e-08
All-DM	Behv-Late-Freeze	**0.812**	6.70e-14	**0.783**	7.89e-13
All-DM	Behv-Obj	0.656	0.011	0.626	0.040
All-DBOW	Embd-Behv	**0.912**	2.10e-15	**0.880**	4.17e-13
All-DM	Embd-Behv	**0.889**	2.22e-17	**0.846**	5.11e-15

Table 3: Prediction Performance of different Embd-* and Behv-* methods to enhance the best base embedding methods ALL-DBOW and ALL-DM. Embd-Behv is a combination method that incorporates LIWC into both learning stages: user embedding learning with LIWC annotated words and parallel networks for these enhanced user embeddings with manually calculated LIWC scores with freezing parameter training.

mance enhancement to the two best base embedding methods: All-DM and All-DBOW. The results show that the performance does not improve when LIWC features are used as an additional optimization objective to regularize the user embedding training (Embd-Obj model) (with the exception of using DBOW for DU task). In contrast, models focusing on masking or enriching input text with LIWC (Embd-Repl and Embd-Anno) outperform all the other models. These methods either replace word tokens by their corresponding LIWC categories (Embd-Repl), or append LIWC categories as additional annotations for each word token (Embd-Anno). For drug use prediction, these methods outperform the baselines by an average ≈ .20 with a max of ≈ .23 in AUC ROC. For delayed discounting prediction, the average increase is ≈ .17 with a max of ≈ .21

6.3. Performance of Behv-* Models

The performance of different Behv-* methods are shown in Table 3. It shows that, all the models incorporating LIWC improve over the base models that use only user embeddings (except Behv-Obj using DBOW for DD task). Particularly, using LIWC to augment embedding features in early or late fusion (rather than Behv-Obj method) yields better results with an increase between ≈ .09 and ≈ .20 for DU prediction and ≈ .04 and ≈ .18 for DD prediction. This incidates that LIWC scores provide the neural network with additional information not being captured using vanilla user embeddings. Secondly, for DU prediction, all the methods except Behv-Late-Freeze fail to outperform the baseline of using LIWC scores alone. The Behv-Late-Freeze model outperformed all the other Behv-* methods for both tasks. It also outperformed the embedding only baselines with an average increase of ≈ .16 and ≈ .13 for DU and DD tasks respectively Comparing with the LIWC only model, the average increase of Behv-Late-Freeze is ≈ .03 and ≈ .15 for DU and DD respectively. This method creates two parallel networks and isolates the parameter training for each type of input (user embedding and LIWC scores respectively). In other words, it minimizes the interactions between the two networks, which seems to benefit the modeling process.

6.4. Performance of the Combined Models

The results of incorporting LIWC into both learning stages is listed in Table 3. On DU task, the results of this method outperform all other methods by an average of ≈ .14 increase compared to LIWC only, ≈ .27 increase compared to vanilla user embeddings and an average ≈ .10 increase compared to the best methods for each stage. For DD task, the results of this method outperform all other methods by an average of ≈ .26 increase compared to LIWC only, ≈ .24 increase compared to vanilla user embeddings and an average ≈ .07 increase compared to the best methods for each stage. This method was constructed using the best methods in each stage. Finally, the results are also significantly better than the the only published results on the same dataset for DD prediction. Our best model, Embd-Behv with All-DBOW achieved a 0.88 ROC AUC comparing to a 0.641 ROC AUC reported in (Ding et al., 2019).

The results indicate that LIWC contributes differently in each of the learning stage. When LIWC is incorporated in the embedding stage, it can help learn more accurate embeddings for human trait and behavior-related concepts and relations. In addition, When it is incproated during the behavior prediction stage, it may provide new features to help discriminate between target classes. As a result, when LIWC is injected in both stages, the system can benefit from not only a better representation of the concepts and relations that are important to trait and behavior analysis but also new features with discrimitive power to help classification.

As a summary, here are some of the main findings from our analysis: (1) injecting LIWC during user embedding (Embd-Anno and Embd-Repl) works better than injecting during behavior prediction (Behv-Early, Behv-Late-NoFreeze, Behv-Late-Freeze). This indicates that user embeddings based on text alone might be too generic for behavior analysis and could greatly benefit from psycholingustic knowledge captured in LIWC; (2) when injecting LIWC during behavior prediction (Behv-*), the best performing model (Behv-Late-Freeze) minimizes the interactions between the two sub-NNs. This suggests when in low resource scenarios, we may not have enough training data to train one set of parameters to account for the mappings of two different types of input to the target. Training each sub-NN separately seems to be the best strategy; (3) injecting LIWC in both stages greatly improves model performance (Embd-Behv). This suggests that the interaction of LIWC and text-based user embeddings captured in the Embd-* models provides additional power over the original LIWC features employed in the Behv-* models for behavior analysis; (4) injecting LIWC through input augmentation rather than reguliza-tion in the form of multi-task learning is consistently better in both stages (Table 3). Dataset size is likely the reason where an additional knowledge source is more valuable to the modeling process as a new input than through knowledge transfer via neural network parameter sharing using multi-task learning; (5) We also experimented with different input granularities: AV-W2V learns user representations from word embeddings, AV-DM learns user embeddings from post embeddings and All-DM learns user embeddings from all the texts authored by the same user (Table 2). Our results indicate that All-DM or All-DBOW outperform the other user embedding methods. This suggests learning a user representation from all the text is better than aggregating embeddings learned from smaller text units (a word or post).

7. Conclusion

In this work, we systematically explore novel methods that incorporate a psychcolingustic dictionary LIWC in neural network-based human trait or behavior analysis. We have investigated different algorithm design choices and assessed their impact on system performance under low resource scenarios. Our best model features (1) simultaneouly injecting LIWC in both self-supervised user embedding and target behavior prediction; (2) incorporating LIWC information as input augmentation (e.g. as annotations to input text in user embedding) instead of output regulaization; (3) employing late fusion with freeze to minimize cross network parameter sharing and influence. The proposed method significantly outperformed all the baseline models in two tasks. Our best method achieved a 0.912 ROC AUC versus the best NN-only baseline of 0.644 and the LIWC only model of 0.756 on DU prediction and achieved 0.880 ROC AUC versus the best NN-only baseline of 0.643 and the LIWC only model of 0.602 on delayed discounting prediction task. It is also significantly better than a recent published result on the same dataset for DD prediction (0.641). This paper presents a step forward and further research can break down the most useful LIWC categories for other behavior prediction tasks for comprehensive understanding.

8. Bibliographical References

Alessi, S. and Petry, N. (2003). Pathological gambling severity is associated with impulsivity in a delay discounting procedure. *Behavioural Processes*, 64(3):345–354.

Bian, J., Gao, B., and Liu, T.-Y. (2014). Knowledge-powered deep learning for word embedding. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 132–148. Springer.

Ding, T., Bickel, W. K., and Pan, S. (2017). Multi-view unsupervised user feature embedding for social media-based substance use prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2275–2284.

Ding, T., Bickel, W. K., and Pan, S. (2019). Predicting delay discounting from heterogeneous social media data. *Social Network Analysis and Mining*, 9(1):27.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

Golbeck, J., Robles, C., and Turner, K. (2011). Predicting personality with social media. In Desney S. Tan, et al., editors, *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Extended Abstracts Volume, Vancouver, BC, Canada, May 7-12, 2011*, pages 253–262. ACM.

Hochreiter, S. and Schmidhuber, J. (1997). Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479.

Iacobelli, F., Gill, A. J., Nowson, S., and Oberlander, J. (2011). Large scale personality classification of bloggers. In *international conference on affective computing and intelligent interaction*, pages 568–577. Springer.

Jo, H. and Choi, S. J. (2018). Extrofitting: Enriching word representation and its vector space with semantic lexicons. *arXiv preprint arXiv:1804.07946*.

Kirby, K. N., Petry, N. M., and Bickel, W. K. (1999). Heroin addicts have higher discount rates for delayed rewards than non-drug-using controls. *Journal of Experimental psychology: general*, 128(1):78.

Kosinski, M., Matz, S. C., Gosling, S. D., Popov, V., and Stillwell, D. (2015). Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines. *American Psychologist*, 70(6):543.

Kulkarni, V., Kern, M. L., Stillwell, D., Kosinski, M., Matz, S., Ungar, L., Skiena, S., and Schwartz, H. A. (2018). Latent human traits in the language of social media: An open-vocabulary approach. *PloS one*, 13(11).

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Mairesse, F., Walker, M. A., Mehl, M. R., and Moore, R. K. (2007). Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research*, 30:457–500.

Majumder, N., Poria, S., Gelbukh, A., and Cambria, E. (2017a). Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32:74–79, 03.

Majumder, N., Poria, S., Gelbukh, A., and Cambria, E. (2017b). Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nguyen, H.-T. and Nguyen, M.-L. (2017). Sentence modeling with deep neural architecture using lexicon and character attention mechanism for sentiment classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 536–544.

Pan, S. and Ding, T. (2019). Social media-based user embedding: A literature review. In *IJCAI*.

Pennebaker, J. W. and King, L. A. (1999). Linguistic styles: Language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296.

Pennebaker, J. W., Mehl, M. R., and Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577.

Pennebaker, J. W., Boyd, R. L., Jordan, K., and Blackburn, K. (2015). The development and psychometric properties of liwc2015. Technical report, University of Texas at Austin.

Riff, D., Lacy, S., Fico, F., and Watson, B. (2019). *Analyzing media messages: Using quantitative content analysis in research*. Routledge.

Robinson, R. L., Navea, R., and Ickes, W. (2013). Predicting final course performance from students' written self-introductions: A liwc analysis. *Journal of Language and Social Psychology*, 32(4):469–479.

Saleem, S., Prasad, R., Vitaladevuni, S., Pacula, M., Crystal, M., Marx, B., Sloan, D., Vasterling, J., and Speroff, T. (2012). Automatic detection of psychological distress indicators and severity assessment from online forum posts. In *Proceedings of COLING 2012*, pages 2375–2388, December.

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, M. E., et al. (2013). Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791.

Shin, B., Lee, T., and Choi, J. D. (2016). Lexicon integrated cnn models with attention for sentiment analysis. *arXiv preprint arXiv:1610.06272*.

Stillwell, D. J. and Tunney, R. J. (2012). Effects of measurement methods on the relationship between smoking and delay reward discounting. *Addiction*, 107(5):1003–1012.

Teng, Z., Vo, D. T., and Zhang, Y. (2016). Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1629–1638.

Trotzek, M., Koitka, S., and Friedrich, C. (2018). Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences. *IEEE Transactions on Knowledge and Data Engineering*, 32:588–601, 04.

Yang, T., Yang, F., Ouyang, H., and Quan, X. (2021). Psycholinguistic tripartite graph network for personality detection. *arXiv preprint arXiv:2106.04963*.

Yarkoni, T. (2010). Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers. *Journal of research in personality*, 44:363–373, 06.

Yu, M. and Dredze, M. (2014). Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 545–550.

Zafarani, R. and Liu, H. (2015). Evaluation without ground truth in social media research. *Communications of the ACM*, 58(6):54–60.