Autoregressive Affective Language Forecasting: A Self-Supervised Task

Matthew Matero
Stony Brook University
mmatero@cs.stonybrook.edu

H. Andrew Schwartz
Stony Brook University
has@cs.stonybrook.edu

Abstract

Human natural language is mentioned at a specific point in time while human emotions change over time. While much work has established a strong link between language use and emotional states, few have attempted to model emotional language in time. Here, we introduce the task of *affective language forecasting* – predicting future change in language based on past changes of language, a task with real-world applications such as treating mental health or forecasting trends in consumer confidence. We establish some of the fundamental autoregressive characteristics of the task (necessary history size, static versus dynamic length, varying time-step resolutions) and then build on popular sequence models for *words* to instead model sequences of *language-based emotion in time*. Over a novel Twitter dataset of 1,900 users and weekly + daily scores for 6 emotions and 2 additional linguistic attributes, we find a novel dual-sequence GRU model with decayed hidden states achieves best results ($r = .66$). We make our anonymized dataset as well as task setup and evaluation code available for others to build on.

1 Introduction

With the growth of social media, natural language processing has increasingly turned toward problems in the social scientific domains. Language has been used to predict personality disorders, political ideology, and mental health (Preotiuc-Pietro et al., 2016a; Iyyer et al., 2014; Coppersmith et al., 2014). However, such predictions are typically made cross-documents or -users rather than cross-time (forecasting the future).

Predictions across time can enable another level of applications for NLP in the social sciences. For example, knowing how someone's mood may change next week can be a valuable insight into preemptively preparing mental health assistance. We may be able to better predict a substance abuse relapse or suicide attempt, or provide individual insights into the precipitators of such events.[1] Applications also exist in other domains, such as correlating mood and personal buying habits, or predicting the mood of performers (e.g. professional athletes) as insight into how well they will perform next.

One of the challenges for doing temporal predictions based on language is the availability of time-series data. Most user-level datasets only provide one measurement, typically of a demographic or psychological trait (something that doesn't change frequently). On the other hand, document-level data (such as that of most sentiment analysis) may provide information across time, but they are typically divorced from the individual. Alternatively, instead of relying on annotations or self-reported data, one can simply predict the future language-based measurement (i.e. as output from another model).

Here, we propose a method of forecasting any *future* dimension of language, based on their *current and past* language. Formally, we define *affective language forecasting* as finding a function of past language (f) which estimates the output of a function of future language (a):

$$f(X_{t-1}^d, X_{t-2}^d, ..., X_{t-n}^d) + \epsilon = a(X_t^d) \tag{1}$$

[1] Deaths from suicide are the second leading cause of death for young adults and overdoses are on the rise, not far behind (Curtin et al., 2016; Rudd et al., 2016)

Proceedings of the 28th International Conference on Computational Linguistics, pages 2913–2923
Barcelona, Spain (Online), December 8-13, 2020

where X represents a d-dimensional feature vector derived from language use at a particular time-step and a is conceptually a mapping of future language to the characteristic of interest. ϵ represents an error to be minimized. We show work towards this problem by forecasting multiple different dimensions of affective language. Affective language can be defined as any language that is related to mood or emotions, such as measuring affective valance(positive/negative mood) or something more specific such as the emotion of surprise. This work covers the forecasting of: Affective valence, Intensity, Anger, Disgust, Joy, Fear, Sadness, and Surprise.

Inclusion of the time dimension to an NLP problem can increase its modeling complexity significantly. We lay a modest foundation for handling these complexities, such as, the amount of history needed at the user level and what formatting of data needs to be done to facilitate modeling accurate predictions(i.e. time-step resolution). We also address the issue of gathering a large time-series dataset that tracks dynamic information at the user level.

Our contributions include:

- The definition of the task of *self-supervised affective forecasting* of lexicon frequency and the release of our novel dataset to promote community participation[2]
- Analysis of the characteristics of language that exist over the temporal dimension, notably related to the amount of history necessary to generate accurate forecasts of future linguistic trends
- Evaluation of modern deep learning techniques and their application or modification to the domain of time-series language forecasting
- Introduction of a new validation paradigm for time-series forecasting, making a distinction between an out-of-time forecast and users who are out-of-sample

2 Related Work

Time-series forecasting has been applied to many fields and applications. Traditionally, linear autoregressive(AR) models have been used in areas such as weather (Hillmer and Tiao, 1982) and stock forecasting (Pai and Lin, 2005). Similarly, AutoRegressive Integrated Moving Average(ARIMA) models are also popular choices in practice for imparting *stationarity*[3] (Franses, 1991) to the time-series, specifically from the integration(I) step.

However, in recent years model architectures such as LSTM (Hochreiter and Schmidhuber, 1997) or GRU cells (Chung et al., 2014) have shown promise with their ability to model long-term dependencies. The recurrent nature of these models allow them to adapt to a time-series forecasting domain. Where input features are often lagged values of the predicted outcome, while still offering a more flexible architecture and non-linear modeling capabilities.

Such architectures have been used extensively in the medical domain, by leveraging electronic health records, to forecast a patient's future physical health. For example, the work of (Choi et al., 2016) designs a stacked GRU architecture to forecast the diagnosis of a patient at their next doctor's visit, while jointly learning to predict the duration until the next visit(an example of a poison process).

In the language domain the only similar work is that of (Halder et al., 2017). In which, their goal is to predict an online forum participants "emotional progression" measured by their *negative emotion index*, a simple metric that defines the polarity of a user's word count$\left(\frac{num(neg_words)-num(pos_words)}{num(total_words)}\right)$. We differentiate our work by focusing on forecasting established psychological definitions of affect and by using only linguistic features to forecast future linguistic trends.

3 Data and Time-Series

We derived our data set from a collection of publicly posted tweets previously used for demographic prediction (Volkova et al., 2013; Sap et al., 2014).

As one of the first studies of language forecasting of any kind, we sought to primarily focus on a well-represented time resolution – weekly aggregates of tweets. This helped us focus on the fundamentals

[2] Data available at https://github.com/MatthewMatero/AffectiveLanguageForecasting

[3] Ensuring that our time-series maintains the same, approximate, statistical properties over time. Such as mean and variance

of the methods rather than missing data issues.[4] Only tweets without URLs (i.e. no retweets) were considered and users were filtered based on having 3+ original tweets per week, resulting in a dataset containing 850,000 tweets spanning 1,900 users. Our dataset contains users who posted between the years of 2009 and 2014 and had 20 contiguous weeks of data.

Additionally, we treat the problem as that of predicticting future *change* in affective language as opposed to modeling the *static* scores. By focusing on the change, the data is essentially detrended as if mean centered (i.e. increasing *stationarity* in our time-series) while the static scores can still be derived. To determine the change, a difference is taken between neighboring time steps for all features. After which, 19 data points remain in our sequence which is further split into 14 weeks for training and 4 weeks for testing per user.

3.1 Linguistic Dimensions

Affect and Intensity: As our primary emotional dimension, we focus on affective valence. Valence, in conjunction with *intensity* or arousal, are often considered fundamental coarse-grained constructs for modeling emotional states (Russell, 1980; Posner et al., 2005). Valence can be thought of as the extent of positive or negative emotion. We use the pre-trained model valence and intensity of (Preoţiuc-Pietro et al., 2016b).

Six Emotions: To model the language of finer-grained emotions beyond just positive/negative valence, we include the basic 6 emotions (Ekman, 1999). These consist of the following: Anger, Disgust, Joy, Fear, Sadness, and Surprise. We extract these emotions from text using a human curated lexicon, as defined by the NRC (Mohammad and Kiritchenko, 2015). The lexicon leverages the amount of words that display each emotion per message to measure each emotion.

Embeddings: Language is also directly modeled as covariates to our psychological attributes in the form of word embeddings. 2 different sets of representations are learned from our data in an effort to evaluate which works better for temporal tasks. First, we learn 50-dimensional word2vec embeddings over a large corpus of social media data (15 million tweets). Second, we extract BERT representations (Devlin et al., 2019) for our tweets using bert-large and concatenate outputs from the last 4 layers. After extraction of BERT features, a non-negative matrix factorization (Févotte and Idier, 2011) is learned and applied using a subset of 50,000 tweets stratified across users. This matrix factorization brings the BERT features down to 50-dimensions to match word2vec. Embeddings are averaged across all words from a given week from a user. Models that leverage these additional context features are referred to as *multivariate*, as opposed to pure autoregressive(*univariate*) models.

3.2 Daily data

To consider the effect of temporal resolution and sparsity of data, a setup over days rather than weeks was also devised. This poses two interesting problems: (1) most users do not tweet in consecutive days leading to some missing data and (2) daily emotions are more prone to large fluctuations compared to longer stretches of time. The setup of the data is the same as the weekly counterpart, except that if a user does not have data for a given day it is marked empty for the models that can handle missing data (see GRU-D) or as no change, otherwise. These days are also excluded from testing data (so only days that had tweets are tested against). For daily resolution steps only a single linguistic dimension is explored: Affective valence.

3.3 Self-supervision

A key feature of affective language forecasting is its ability to be trained with self-supervision. Thus, we propose using the inherent structure of the language itself as the forecasting target. Much like how a language model will use the next observed word as its target we will use the next "observed" attribute of language, as determined by a pre-trained model (Preoţiuc-Pietro et al., 2016b).

[4] We also explored the, more sparse, daily level as a final experiment.

Importantly, the ratings used by Preotiuc-Pietro et al had high inter-rater agreement ($icc = .77$) and the models themselves were quite accurate in terms of producing continuous-valued scores($r = .65$ and $r = 0.80$).

When not leveraging a pre-trained model, a hand curated lexicon is used to extract the underlying emotional target. As described for the 6 NRC emotions.

3.4 Train-Test Setup: Time and Users

In training a k-fold validation(k=5) is also performed, where for each fold the following sub-splits are generated: train users(weeks 1-14), test users 1 (weeks 15-19 of train users) referred to as *out-of-sample time*, test users 2 (weeks 15-19 of users held-out from training/validation) referred to as *out-of-sample users*, and validation users (weeks 15-19 of users held-out from training, different than test users 2).

In more detail, for train users only their first 14 weeks are used to train our models. The remaining weeks of these users form our main test set(in-sample user, out-of-sample time(**OOS Time**)). Validation users are out-of-sample users where only weeks 15-19 of their time-series is used to tune hyperparameters. Lastly, our second test set is comprised of weeks 15-19 from users excluded from both training and validation(out-sample user, out-of-sample time (**OOS User**)). This is because in practice, there could be applications where users will not be available for training but only for testing.

4 Methods

We seek to model the affective score of future language (Y_t; e.g. the positive/negative valence of the language) as a function of past affective language ($Y_{t\text{-}1}, Y_{t\text{-}2}, ...$) as well as other dimensions of past language ($X_{t\text{-}1}, X_{t\text{-}2}, ...$; e.g. vector embeddings):

$$Y_t = f([Y_{t\text{-}1}; X_{t\text{-}1}], [Y_{t\text{-}2}; X_{t\text{-}2}], ..., [Y_{t\text{-}k}; X_{t\text{-}k}]) + \epsilon \tag{2}$$

where t represents the next time-span (e.g. next week, or tomorrow) and k represents a history length (order) of data to look backwards. All observations are over users.

This self-supervised objective, assumes affect scores at a given time point can be determined based only on the language at that time (i.e. $Y_t = a(X_t)$ where a is a model or mapping from language at time t to affect score). For example, a simple affect score may come from a lexicon of positive words which always produces the same affect score given the same set of word counts. On the more sophisticated side, the score may come from a complex predictive model based on the writings of the particular user at a particular time.

4.1 Dual Sequence, Decayed, GRU Network

Our primary approach is motivated by adapting powerful autoregressive sequence models for words and phrases (recurrent neural networks) to our task of forecasting affective language across weeks or days.

$$\hat{Y}_t = f(g(h_{t\text{-}1}, Y_{t\text{-}1}, X_{t\text{-}1})) \tag{3}$$

where Y and X are the given affect scores and vector embeddings and h_{t-1} is the hidden state from the previous time step:

$$h_s = g(h_{s\text{-}1}, Y_{s\text{-}1}, X_{s\text{-}1}) \tag{4}$$

f is the aggregation and activation function for the output layer and g is the composition and activation function for the recurrent layer.

Alternatively, one could model the progression as an aggregation over 2 separate, but related, sequences.

$$\hat{Y}_t = f(g_1(h^{(1)}_{t\text{-}1}, Y_{t\text{-}1}), g_2(h^{(2)}_{t\text{-}1}, X_{t\text{-}1})) \tag{5}$$

$$h^{(1)}_s = g_1(h_{s\text{-}1}, Y_{s\text{-}1}) \tag{6}$$

$$h^{(2)}_s = g_2(h_{s\text{-}1}, X_{s\text{-}1}) \tag{7}$$

where each g_n is their own recurrent model learning from separate sequences of the affect score and other language dimensions. f now aggregates their learned representations and generates a prediction. A simple f would be to average the 2 separate predictions from each sequence.

GRU Setup: We started with a standard GRU based RNN, due to their similar performance to LSTMs while requiring fewer computations, but change a few initial configurations to better reflect the characteristics of the autoregressive models rather than the typical sequential language tasks where GRU is mostly applied. A shared learnable initial hidden state parameter was added to the network, allowing the network to focus less on hidden state calculations at early time steps through the sequence. Additionally, all of the reset gate biases were initialized to -1, favoring retention of hidden states.

An attention layer is also included to allow the GRU to weight time-steps. The attention calculation is shown in equations 8, 9, and 10. Notice that in this context attention acts as a way of highlighting which time-steps may contribute more to mood fluctuations and the score itself is based on the hidden states of the other weeks. The first 2 equations handle the calculation of our attention weights for each time-step and the third is a weighted sum of the hidden states.

$$\hat{H} = tanh(H_{1:t}) \tag{8}$$

$$A_t = Softmax(W * \hat{H} + B) \tag{9}$$

$$attn = \sum A_t * H_t \tag{10}$$

Decayed GRU: GRU-D (Che et al., 2018) has two important differences over a traditional GRU in time-series tasks. First, it can account for missing data that is common across many time-series datasets. Secondly, it applies a (learned) decay parameter across hidden-states to better emulate the passing of time. An overview of these decays are below: 11 shows the formula for the learnable decay parameters where δ_t is a time interval vector which encodes the time gap between observations, 12 shows the parameter applied to the input layer where m_t^d is a binary masking vector for variable d at time t, $\tilde{x}^d$ is the mean, and $x_{t'}^d$ is the last observed value of x^d and 13 is the decay applied to hidden states.

$$\gamma_t = \exp\left(-max(0, W_\gamma \delta_t + b_\gamma)\right) \tag{11}$$

$$\hat{x}_t^d = m_t^d x_t^d + (1 - m_t^d)(\gamma_{xt}^d x_{t'}^d + (1 - \gamma_{xt}^d)\tilde{x}^d) \tag{12}$$

$$\hat{h}_{t-1} = \gamma_{h_t} \odot h_{t-1} \tag{13}$$

Dual-Sequence: In the models thus far, the GRU does not explicitly handle the input dimension representing previous outcomes ($Y_{t\text{-}1}$) any different than any dimension of the input embedding. Thus, we also developed another architecture that utilizes 2 parallel GRU layers. In this configuration, one GRU layer will perform a pure(univariate) autoregression where its only input will be the affect score(affective valence, intensity, etc) and the other GRU will take in any context variables(embeddings). This approach enables separate attention layers, possibly highlighting different time-steps depending on the type of data, while still training simultaneously. Structure of our proposed model is shown in figure 1, 2 GRUs with their own attention layers and their final prediction being a (learned) weighted average of their outputs.

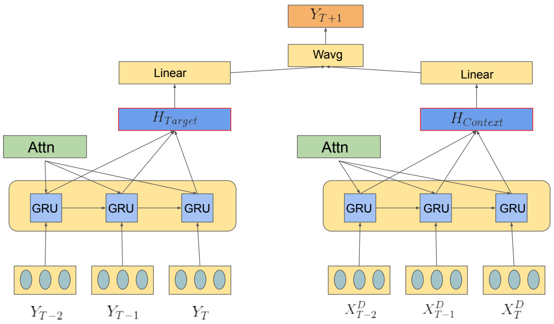


Figure 1: Architecture of parallel dual GRU system. Each GRU has its own attention weights, the red border denotes tanh activation across hidden output. Wavg is a weighted average where weights are learned through training with an initial configuration of .6/.4 target/context respectively

Dynamic Model Order: Normally when training an autoregressive model a restriction is set to apply a fixed number of lag observations, commonly referred to as the *order* of regression denoted as *p*. The order *p* is the amount of language history added to the model to make a prediction. Structure of an autoregressive model can be seen in figure 2.

Due to the dynamic unrolling nature of RNNs, there is no restriction to training with a set fixed p. Here, we introduce dynamic order training: generating training examples of various length sequences for each user with a selected minimum order and then add 1 more observation for each new history length. Resulting in a training set where the first example is taken from each fixed order model(use weeks 1-3 to predict 4, 1-4 to predict 5, until reaching the maximum 1-14 to predict 15). Both fixed and dynamic p models are explored in the context of GRUs.

For models that can directly handle missing data(GRU-D) we mark dynamic p sequences shorter than the maximum length(14) as missing. When using a fixed p we run GRU-D in a slightly different configuration, namely **GRU-DS**, which only accounts for the internal state decay over time and doesn't directly model the missing data on the input layer.

4.2 Alternative Approaches

Deep Averaging Network: One way to model this problem is to ignore the sequential information entirely but model the non-linearity's between the average change in dimensions over a time-span. However, this approach is naive and is not encoding the most important aspect: time. A deep averaging network(DAN) (Iyyer et al., 2015) with 2 dense layers and sigmoid activation between them is used. During univariate training there are only 2 hidden units at each layer, while there are 35 when doing multivariate.

Support Vector & Gradient Boosting Regression: These 2 models are popular approaches for many machine learning tasks (Yoshimi and Kotani, 2019; Liu et al., 2019). The SVR chosen uses a linear kernel and both models are implementations from the sklearn python package (Pedregosa et al., 2011).

Autoregressive Ridge (AR-Ridge): A L2-regularized(ridge) linear regression is also considered. In the field of time-series analysis a traditional linear regression is often preferred for these types of tasks (Lydia et al., 2016; Balakrishna et al., 2018).

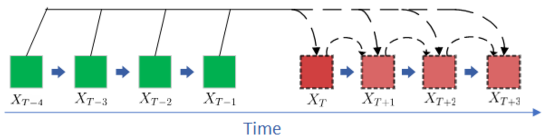


Figure 2: Visualization of an autoregressive prediction with an order p=4. The first prediction is made using only the green(previously observed) data. For each proceeding prediction the lag moves forward 1 time-step, but still maintains 4 previous observations as input.

Baselines: Simple Moving Average and Constants: As a new task we propose a collection of heuristic baseline models. For example, to predict the future one may reuse values from the past(i.e. someone's change in mood may be identical to their change last week).

We employ standard "zero rule" baselines, with some modifications to time-series. These are intended to capture easy to implement or naive approaches for prediction (Andrzejak et al., 2008; Miglionico, 2019). The following 2 baselines are considered: (1) predict the last time step's change (**last(1)**) and (2) average of all available changes over a 14 time-step history (**mean(14)**).

An additional third baseline could be used where one would predict 0 change (**no change**), However, for our evaluation we stick with the previously mentioned 2 baselines due to using *Pearson correlation (r)* as our main evaluation metric. Where predicting 0 change would have no correlation with the actual time-series values. Correlation is chosen as it is resistant to the scale of the target variable. Thus allowing direct comparison of difficulty in forecasting across all of our affective dimensions

Transformer Networks: A popular approach to many tasks in Natural Language Processing is to *fine-tune* a pre-trained language model (i.e. BERT) rather than train a smaller model from scratch. However, these language models are pre-trained over individual messages and would require alternative architectures for aggregating over multiple messages and capturing time. Since we are introducing the problem we focus on a more fundamental setup and suggest such architectures would be an interesting direction for future work. For example, one could explore various architectures for hierarchical models on top of standard pre-trained transformer language models.

Model (order)	r
Baselines	
Last (1)	.47
Mean (14)	.49
DAN (14)	.49
GRU (14)	.62
GRU-DS (14)	**.64**
GRU-D (14)	**.64**

Table 1: Performance of proposed GRU models: GRU with attention, and GRU with memory decay (GRU-D, GRU-DS) compared with baselines and DAN. All models use the max, order of 14, fixed history (p) and run with the pure auto-regressive (univariate) input. Bold results are found significant with $p < .05$ using a paired t-test.

Model (order)	**OOS Time**(r)	**OOS Users**(r)
GRU (11+)	.63	.63
GRU-D (11+)	.63	.64
GRU-DS (9)	.64	.64

Table 2: Comparison of our various GRU model performance when using in-sample users, out-sample time and out-sample users, out-sample time. Uses pure auto-regressive setting (only history of previous outcomes).

Ethics Statement: This research has been approved (deemed exempt status) by an academic institutional review board.

5 Evaluation

The first set of results showcases the chosen zero rule baselines compared to all deep learning models using a fixed order of 14 (the max). We show the results for the test set consisting of **OOS Time** users for target variable affective valence. The zero rule baselines perform quite well, showing that even out-predicting the past can be challenging. Note that the deep averaging network is operating on the same information as the **mean(14)** baseline, so we expect it to perform similarly.

Additionally in practice there are many users who would be forecasted but would possibly not be part of the training data of any particular model. To emulate this we examine results for a set of held-out users for which our model has neither trained nor been tuned on. We compare the results across select models from the previous test set and this **OOS User** test set in table 2. There seems to be no loss in predictive power when evaluating against this held out set; showing the robustness of the proposed models when fitting to a real application scenario.

In figure 3, we show the effect of order p on our autoregressive models with respect to correlation in a pure autoregressive setup. As mentioned, a benefit of using recurrent neural models for time-series modeling is the ability to implement dynamic unrolling on variable length sequences. To explore the effectiveness of our dynamic order technique for time-series, we directly compare the GRU models using variable p against training with a fixed p.

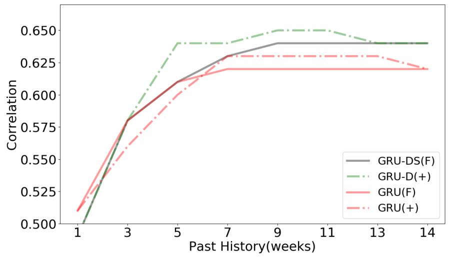


Figure 3: Comparison of GRUs for fixed and variable order pure autoregressive approaches as more history is added to the training examples for target variable affective valence. GRU-D always leverages 14 weeks of data (Where anything older than the set order is marked as missing and leverages the mean instead). + denotes dynamic order and F denotes fixed order.

We find that applying the dynamic order to a model such as GRU-D, where data can be marked as missing, gives clear benefits when compared against smaller fixed orders. However, as the order size increases this loses its effectiveness. With respect to the amount of p required to create a robust model, we typically find that most models perform best when utilizing 9 weeks of p before saturating.

We believe this is due to the shrinking of the training set. For each extra time-step of history, there is a loss of 1 training example per user. In other words, each increase of 1 time-step of p reduces training

Model (order)	**Uni**(r)	**W2V**(r)
Joint Sequence		
GRU (11+)	.63	.63
GRU-DS (13)	.64	.64
GRU-D (9+)	.65	.65
Dual Sequence		
GRU (11+)	-	.63
GRU-DS (13)	-	**.66**
GRU-D (9+)	-	**.66**

Table 3: Pure auto-regressive versus contextual auto-regressive / Joint versus Dual: Comparison of univariate (pure autoregressive) versus multivariate (contextual – includes features beyond the target variable for the past) across the best models. Bold results show significance with $p < .05$.

Model (order)	**W2V**(r)	**BERT**(r)
Dual Sequence		
GRU (11+)	.63	**.62**
GRU-DS (13)	**.66**	.61
GRU-D (9+)	**.66**	.61

Table 4: Comparison between the different types of language representations, word2vec and bert across their best GRU configurations. We believe BERT does not directly capture what our self-supervision technique highlights for emotions. Bold results show significance with $p < .05$.

examples by approximately 1,900.

In table 3, we compare results across models when word embedding features are also included. From our results we can see that utilizing the dual-GRU setup where one GRU handles a univariate autogression and the other handles context variables is generally more favorable than utilizing only a single(joint) GRU. However, generally there seems to be difficulty in extracting useful temporal information from language representations in most cases.

We also highlight the differences in performance between using word2vec and BERT representations in table 4. Typically, word2vec performs better than BERT for this task. We believe this is due to 2 reasons: (1) the objective of word2vec is more closely related to the way the pre-trained models were trained(BOW) and how lexicons judge emotion (word occurrence, not context) and (2) BERT may be too high of a level of semantics when trying to forecast the underlying temporal characteristics of language.

Affective valence prediction is also examined using off-the-shelf models such as SVR, GBR, and (ridge) autoregression. Here, we show the best performing variants of each(their optimal p) and compare to our best GRU model(dual GRU-DS p=13), shown in table 5.

A robustness check is performed to analyze sensitivity to seasonal trends. Figure 4a shows the squared error averaged across all years of data separated by month for affective valence. Notably, there is not a large increase in error during the winter months where one may expect emotions to be abnormal due to the many holidays that occur, but rather error seems more dependent on number of users present in that testing month, suggesting that this task scales well with respect to dataset size at the user level.

Robustness is also checked within user, where we visualize our model's forecasted weekly changes for a person's affective valence, shown in figure 4b. Our model achieves great success at not only producing accurate predictions but also capturing general trends in mood. However, there is evidence of the model struggling when a user continually has large changes in mood week after week.

Results for affective attributes listed in section 3.1 are shown in table 6. Similar trends to affective

Model (order)	r
SVR (13)	.64
GBR (5)	.58
AR-Ridge (9)	.64
GRU-DS (13)	**.66**

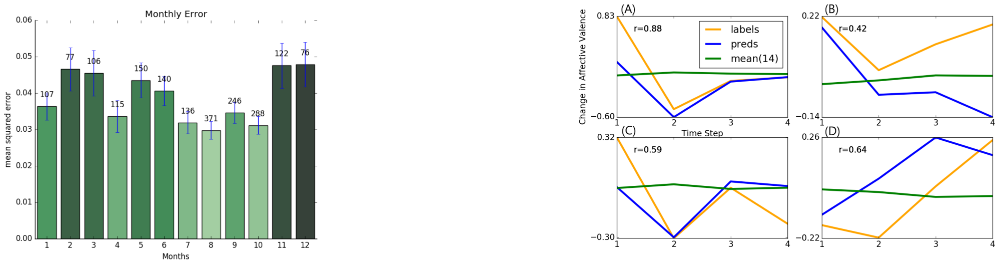
Table 5: Performance among traditional ML models given their optimal p and our best performing GRU model. The traditional models perform quite well offering simplistic use, however the improvement with deep learning implies there is more room to grow given larger data and model flexibility. Bold results show significance with $p < .05$

Language Variable	r
Intensity(Arousal)	.63
Anger	.64
Disgust	.63
Fear	.65
Joy	.63
Sadness	.66
Surprise	.64

Table 6: Results for the best model (dual GRU-DS) for all other linguistic emotion dimensions for **OOS Time** testing. In this configuration we used multivariate input(word2vec) and an order of 13 weeks.

Model (order)	r
Last (1)	.22
SVR (3)	.36
AR-Ridge (3)	.37
GRU (3+)	.34
GRU-DS (3)	.37
GRU-D (3+)	**.38**

Table 7: Pure autoregressive **OOS Time** results across models at **daily resolution**. All models use a observation order of 3, which was found ideal, except the zero rule baseline which is 1 by definition. Bold results show significance with $p < .05$.



(a) Error analysis across time

(b) Error analysis within user

Figure 4: (A) Average squared error per month across our OOS Time set; shown with standard error bars and the amount of users in that month. Higher error is related to having fewer users that fell into that test month rather than influence from any seasonal event(i.e. winter holidays or summer vacations). (B) Dual GRU-DS performance when evaluating against a user's change in affective valence across weeks, compared with baseline mean(14) and the ground truth labels. We find that our predictions are typically accurate. However, even when they fail the general trend is still present

valence are seen across these attributes, implying there is an underlying structure of affective language in the temporal domain.

Finally, performance on the daily resolution dataset for affective valence is presented. A brief selection of results is shown in table 7. In this setup, we only utilize a pure autoregressive approach rather than multivariate, due to the abundance of missing data. GRU-D performs the best on this task, which we believe is due to the nature of GRU-D and its ability to model missing data directly.

6 Conclusion

We presented a new subtask in NLP, *self-supervised affective forecasting*. The models proposed in this work cover a wide range from traditional machine learning to deep learning techniques. While traditional models perform well, GRUs with decayed hidden states provided more accurate results and offer a more flexible architecture. We find this most helpful when experimenting with extra context features, such as word embeddings, splitting the inputs into 2 separate forecasts for a dual sequence GRU offers the best predictive power.

We also show that our modeling choices are robust to both seasonal variation and multiple evaluation paradigms: **OOS Time** and **OOS User**. Lastly, we found a strong relationship between the order of our models and the accuracy of future predictions with a performance leveling out roughly at a history of 9 weeks for pure autoregressive and 13 weeks for dual GRU-DS.

Affective forecasting can be leveraged for many domains, such as social sciences (mental health), business (consumer buying habits, targeted ads), and athletics (game performance). We encourage others to pick up this challenging task by releasing our novel dataset and task setup code.

References

Artur Andrzejak, Luis Silva, and DE Informctica. 2008. Robust and adaptive modeling of software aging.

N Balakrishna, HL Koul, M Ossiander, and L Sakhanenko. 2018. Fitting a pth order parametric generalized linear autoregressive multiplicative error model. *Sankhya B*, pages 1–20.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085.

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pages 301–318.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Glen Coppersmith, Mark Dredze, and Craig Harman. 2014. Quantifying mental health signals in twitter. In *Proceedings of the workshop on computational linguistics and clinical psychology: From linguistic signal to clinical reality*, pages 51–60.

Sally C Curtin, Margaret Warner, and Holly Hedegaard. 2016. Increase in suicide in the united states, 1999–2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019: The Annual Meeting of the North American Association for Computational Linguistics*.

Paul Ekman. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Cédric Févotte and Jérôme Idier. 2011. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural computation*, 23(9):2421–2456.

Philip Hans Franses. 1991. Seasonality, non-stationarity and the forecasting of monthly time series. *International Journal of forecasting*, 7(2):199–208.

Kishaloy Halder, Lahari Poddar, and Min-Yen Kan. 2017. Modeling temporal progression of emotional status in mental health forum: A recurrent neural net approach. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 127–135.

Steven C Hillmer and George C Tiao. 1982. An arima-model-based approach to seasonal adjustment. *Journal of the American Statistical Association*, 77(377):63–70.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. 2014. Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1113–1122.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691.

Ran Liu, Joseph L Greenstein, Sridevi V Sarma, and Raimond L Winslow. 2019. Natural language processing of clinical notes for improved early prediction of septic shock in the icu. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6103–6108. IEEE.

M Lydia, S Suresh Kumar, A Immanuel Selvakumar, and G Edwin Prem Kumar. 2016. Linear and non-linear autoregressive models for short-term wind speed forecasting. *Energy Conversion and Management*, 112:115–124.

Marco Miglionico. 2019. *A Deep Learning Framework for Air Pollution Forecasting and Interpolation*. Ph.D. thesis.

Saif M Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Ping-Feng Pai and Chih-Sheng Lin. 2005. A hybrid arima and support vector machines model in stock price forecasting. *Omega*, 33(6):497–505.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Jonathan Posner, James A Russell, and Bradley S Peterson. 2005. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and psychopathology*, 17(3):715–734.

Daniel Preotiuc-Pietro, Jordan Carpenter, Salvatore Giorgi, and Lyle Ungar. 2016a. Studying the dark triad of personality through twitter behavior. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 761–770. ACM.

Daniel Preoţiuc-Pietro, H Andrew Schwartz, Gregory Park, Johannes Eichstaedt, Margaret Kern, Lyle Ungar, and Elisabeth Shulman. 2016b. Modelling valence and arousal in facebook posts. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 9–15.

Rose A Rudd, Noah Aleshire, Jon E Zibbell, and R Matthew Gladden. 2016. Increases in drug and opioid overdose deaths—united states, 2000–2014. *American Journal of Transplantation*, 16(4):1323–1327.

James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.

Maarten Sap, Gregory Park, Johannes Eichstaedt, Margaret Kern, David Stillwell, Michal Kosinski, Lyle Ungar, and Hansen Andrew Schwartz. 2014. Developing age and gender predictive lexica over social media. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1146–1151.

Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1815–1827.

Takehiko Yoshimi and Katsunori Kotani. 2019. Measurement of pronunciation difficulty using support vector regression. In *EdMedia+ Innovate Learning*, pages 711–715. Association for the Advancement of Computing in Education (AACE).