Combining Natural Language and Machine Learning for Predicting Survey Responses of Social Constructs in a Dyad

Bruno Abreu Calfa [1], Peggy Wu [1], Mohammadamin Sanaei [2], Stephen Gilbert [2], Andrew Radlbeck [1] and Brett Israelsen [1]

[1] *Raytheon Technologies Research Center, 411 Silver Lane, East Hartford, Connecticut, USA*
[2] *Iowa State University, 527 Bissel Rd., Ames, Iowa, USA*

Abstract

Measuring social constructs such as engagement, rapport, and trust often rely heavily on surveys and behavioral observations. This paper describes a method to use features identified by psychology-based language analysis, combined with machine learning, to predict participant survey responses in a training context based on 120 dyad transcripts. The method analyzed data collected from subjects performing a circuit board training task within the project called SCOTTIE, Systematic Communication Objectives and Telecommunications Technology Investigations and Evaluations. In this study, the collected data showed low utterance count and a lack of correlation between features and survey responses, suggesting that the context in which the interactions occurred may limit opportunities for interlocutors to manifest social behaviors verbally, which in turn affected the ability to use language analysis to predict subject perceptions of the interaction. However, the methodology appears sound.

Keywords

Social Computing, Virtual Reality, Natural Language Analysis, Training

1. Introduction

One modality in which humans exhibit social behaviors is language. Linguistic categories [1] have been used in diverse applications from measuring emotional expression [2], to evaluating team dynamics through discourse [3], and identifying correlations between written student self-introductions with course performance [4]. This paper describes the use of Natural Language Processing tools to examine transcripts between trainer-student pairs in a project called Systematic Communication Objectives and Telecommunications Technology Investigations and Evaluations (SCOTTIE). SCOTTIE's goal is to investigate the impact of the interaction media on the effectiveness of achieving communication objectives. The definition of communication objectives is described in [5]. Briefly, the communication objectives of interest include co-presence, engagement, virtual embodiment, rapport, perceived usability, trust, and mental workload.

2. Method

The study protocol involved a scenario where a trained confederate staff member provided scripted instructions on a circuit board repair task to subjects. Subjects were assigned to one of three conditions. Trainer-subject interactions were either conducted through teleconference software (i.e. Zoom), in a bespoke virtual reality based environment (also called extended reality or XR), or Face-to-Face (F2F) in-person visits with a shared computer. All conditions used the same circuit board simulator testbed, where the trainer-subject pair used screen share, controlled their own avatars in the virtual environment, or shared a physical screen, for the Zoom, XR and F2F conditions respectively. The testbed and virtual environment, called Circuit World, is software created by the study staff as described in [6]. At

AAAI 2023 Spring Symposia, Socially Responsible AI for Well-being, March 27–29, 2022, USA
EMAIL: Peggy.Wu@rtx.com
ORCID: 0000-0003-0185-596X

the start of the trial, a research assistant explained the purpose of the study and obtained informed consent. The researcher then administered pre-trial surveys. Upon survey completion and other introductory materials, the trainer entered the session. The trainer provided subjects with approximately 15 minutes of instruction on how to repair a specific circuit board and invited subjects to ask questions. The trainer then left the session, and the researcher initiated the test portion of the session, cuing the testbed for the subject to repair a virtual circuit and complete a multiple-choice quiz based on knowledge conveyed during training. Subjects then completed a post-session survey which contained questions regarding their perception of the trainer and the effectiveness of the communication framed as the aforementioned communication objectives. Only the transcript between the trainer and the subject was used in the analysis. The protocol was approved by Iowa State University's Internal Review Board (IRB). Participants were recruited through Prolific, social media, and email advertisements.

3. Feature Extraction from Transcript Data

Each trainer-subject dyad transcript was generated using Zoom's auto transcription feature and stored as VTT text files following the Web Video Text Tracks format. Transcript files were parsed to extract utterances by trainers and subjects. In the F2F condition, some manual transcript correction was needed due to the lack of speaker diarization. The total number of transcripts was 120, with 33, 50, and 37 transcripts from the F2F, Zoom, and XR conditions respectively. Regarding the utterances in each condition, as expected, the number of utterances by trainers was reasonably consistent across different conditions since confederate trainers were following a script. There were larger variations on the number of utterances by subjects.

The extraction of numerical features from utterances in each transcript file was carried out using the following natural language processing (NLP) methodologies:

Lexical and Semantic Similarity Analysis using Word Embedding. The word embedding approach is based on term-frequency times inverse-document-frequency (tf-idf) [7], which is a term weighting scheme calculated as $tf\text{-}idf(t,d) = tf(t,d) \cdot idf(t)$, where t is a term and d is a document (i.e., utterance), tf(t,d) is a matrix of counts of each uttered term in a document, and $idf(t) = \log\big[(1+n)/\big(1+df(t)\big)\big] + 1$, where n is the number of documents and df(t) is the number of documents in the document set that contain term t. After embedding the utterances by a trainer and a subject into their respective tf-idf arrays, the cosine similarity score [8] is calculated as $k(x,y) = \frac{xy^\top}{\|x\|\|y\|}$, where x and y are tf-idf arrays and $\|x\|$ is the Euclidean norm.

Linguistic Style Matching (LSM). LSM is a technique in behavioral analytics to assess the stylistic similarities in language use across groups and individuals [9]. The procedure measures the degree of similarity between two individual's patterns of function word usage. Function words consist of pronouns, articles, conjunctions, prepositions, auxiliary verbs.

Linguistic Inquiry and Word Count (LIWC). LIWC is a word-counting software that uses a dictionary containing words that belong to over 80 linguistic, psychological, and topical categories indicating various social, cognitive, and affective processes [10]. In this work, the authors used the LIWC application programming interface (API) offered by Receptiviti (https://www.receptiviti.com/liwc).

Valence-Aware Dictionary for Sentiment Reasoning (VADER). VADER is a rule-based, computational sentiment analysis method that aims to measure the sentiments, evaluations, attitudes, and emotions of a speaker/writer [11]. The result of this analysis is a compound polarity score for each utterance calculated as $x/\sqrt{x^2+\alpha}$, where x is the sum of sentiment scores of all the words in the utterance and α is a normalization parameter whose value is typically set to 15.

The dataset of NLP extracted features contained 197 columns, including LIWC features, number of utterances by the subject, LSM comparison between subject and trainer, VADER compound polarity score for the subject, and cosine similarity score between trainer and subject.

4. Building Predictive Models for Communication Constructs

The overall data analytics workflow to correlate communication constructs with trainer and subject utterances is shown in Figure 1. It comprised two pipelines: feature extraction (see

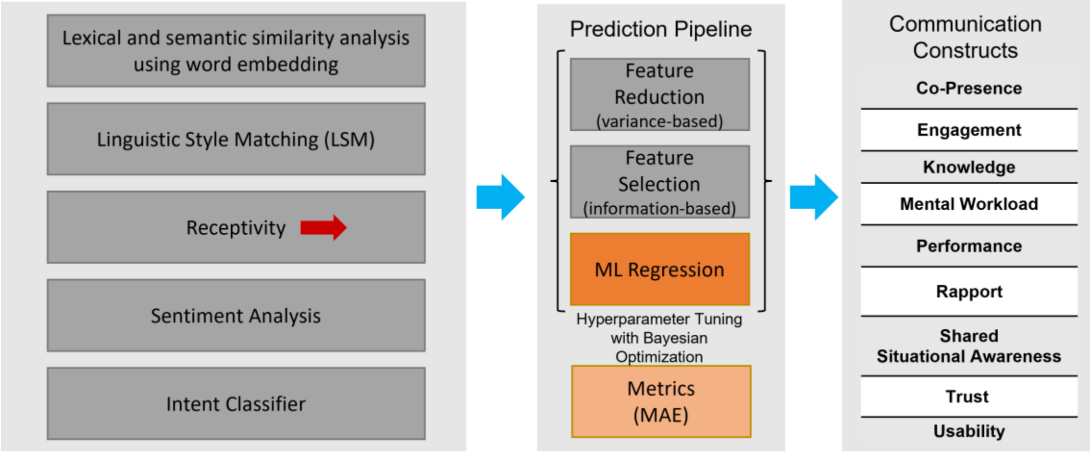
Lexical and semantic similarity analysis using word embedding
Linguistic Style Matching (LSM)
Receptivity
Sentiment Analysis
Intent Classifier
Prediction Pipeline
Feature Reduction (variance-based)
Feature Selection (information-based)
ML Regression
Hyperparameter Tuning with Bayesian Optimization
Metrics (MAE)
Communication Constructs
Co-Presence
Engagement
Knowledge
Mental Workload
Performance
Rapport
Shared Situational Awareness
Trust
Usability

Figure 1: Overall workflow for correlating communication constructs with subject and trainer utterances.

section above) and prediction. The prediction pipeline used statistical and Machine Learning (ML) techniques for data preprocessing and regression analysis coupled with hyperparameter tuning using Bayesian optimization.

The target variables for the regression analysis corresponded to survey responses related to each communication construct. There were 9 high-level constructs that were further divided into 12 targets based on the survey instruments used: co-presence, engagement, virtual embodiment, rapport, usability perceived ease, trust in the trainer (general), trust (in ability), trust (in integrity), trust (in benevolence), mental workload (general), mental workload (in operating the training system), and mental workload (related to communication). Each target variable had its own prediction pipeline. The dataset of NLP extracted features was first merged with the survey responses dataset (containing target variables) on the subject identifier. Both feature data and target values in the resulting dataset were scaled between 0 and 1 (min-max scaling), and then used in the prediction pipeline.

The first step in the prediction pipeline was to apply a variance-based feature reduction procedure to the dataset, which removed all low-variance features according to a variance threshold value. The next step was to use a feature selection approach to keep only the features that had the n highest scores; the criterion was based on mutual information between features and target, using non-parametric estimation methods based on entropy estimation from k-nearest neighbors distances [12]. The final step was to fit a supervised learning regression model to predict the target from the remaining features.

The variance threshold, the value of n (number of features with highest scores to be kept), and the hyperparameters of a regression model were tuned simultaneously and systematically using a Bayesian optimization framework [13]. The optimization approach required the definition of a search space describing the hyperparameters to be tuned and their ranges, as well as an objective function to guide the search. In this work, the mean absolute error (MAE) is used as the optimization criterion and calculated as $MAE = \frac{1}{N}\sum_i |y_i - \hat{y}_i|$, where N was the number of records or rows of the final dataset, y_i was the true target valued of record i, and $\hat{y}_i$ was the predicted target value of record i by the prediction pipeline.

Table 1 Candidate Regression Models and their search space of hyperparameters to be tuned by the Bayesian Optimization Framework

Regression Algorithm	Hyperparameter Space
Extreme Gradient Boosting [14]	n_estimators:{1000,1001,..., 10000} reg_alpha: [0.001,2.0] reg_lambda: [0.001,2.0]
Random Forest [15]	n_estimators:{1000,1001,..., 10000} min_samples_split:[0.001,0.2] min_samples_leaf: [0.001,0.2]
Gradient Boosting [16]	n_estimators:{1000,1001,..., 10000} learning_rate:[0.001,0.3] min_samples_split: [0.001,0.2]

Extremely Randomized Trees [17]	n_estimators:{1000,1001,..., 10000} min_samples_split:[0.001,0.2] min_samples_leaf: [0.001,0.2]

Multilayer Perceptron [18]	hidden_layer_sizes:{2,3,...,200}
	activation:{logistic,tanh,relu}
	learning_rate_init: [0.001,0.3]
	alpha: [0.001,3.0]

Support Vector Machine [19]	C: [0.01,4.0] kernel: {linear,poly,rbf,sigmoid}

K-Nearest Neighbors [20]	n_neighbors:{3,4,...,20}
	weights: {uniform,distance}
	leaf_size: {2,3,...,10}

Stochastic Gradient Descent [21]	{squared_error,huber,epsilon_insensitive, squared_epsilon_insensitive}
	penalty:{l2,l1,elasticnet}
	alpha:[0.0001,0.1]
	learning_rate:{constant,optimal,invscaling,adaptive}

CatBoost [22]	iterations:{1000,1001,...,10000} learning_rate:[0.001,0.3] depth:{2,3,...,10} l2_leaf_reg:[0.001,2.0] random_strength:[0.001,2.0] bagging_temperature: [0.0,10.0]

Table 1 shows the regression algorithms and models considered for all targets, as well as the respective tunable hyperparameters and their search spaces. In addition, the range for the variance threshold parameter is [0.01,0.05] and the range for hyperparameter n (number of features with highest scores to be kept) was {3,4,…,30}. The implementation of Random Forest (XGB) was provided by https://github.com/dmlc/xgboost/, CatBoost (CATB) from https://github.com/catboost/catboost, and the remaining models/algorithms from Scikit-Learn (https://scikit-learn.org). Table 2 describes the algorithm to obtain the best pipeline for each target. Bayesian optimization implementation was provided by Scikit-Optimize (https://github.com/scikit-optimize/scikit-optimize), more specifically, the cross-validated search procedure over the hyperparameter search space.

5. Feature Importance with Shapley Values

A prediction for a target variable can be explained by assuming that each feature value of the instance (i.e., survey record) is a "player" in a game where the prediction is the payout. Shapley values – a method from coalitional game theory – tell us how to fairly distribute the "payout" among the players [23]. Intuitively, Shapley values are computed by carefully perturbing input features and observing how changes to the input features impact the final model prediction. The Shapley value of a given feature is then calculated as the average marginal contribution to the overall model score.

Table 2 Algorithm for obtaining the best prediction pipeline for each target (communication construct)

Inputs: scaled dataset containing NLP feature data and targets, test set size as a percentage (30%), list of ML algorithmsand their search space, number of folds for cross-validation in Bayesian optimization (3), maximum number of Bayesian optimization iterations (300), number of starting points for Bayesian optimization (50)

Output:best pipeline for each target (i.e., pipeline with the lowest MAE on the test data)

```
For each target:
    For each ML algorithm:
        • Construct pipeline consisting of variance thresholding, high-scoring feature selection, and ML algorithm
        • Split dataset into training + validation and test sets
        • Tune hyperparameters of pipelineon training + validation set using cross-validation Bayesian optimization
        • Evaluate tuned pipelineontest set
        • Compute MAEon test set
    Save best pipeline (lowest MAE on test set) for target
```

Mathematically, the Shapley value of player (i.e., feature) i is calculated as

$$\phi_i(v) = \sum_{S\subseteq N\setminus\{i\}} \frac{|S|!\,(N-|S|-1)!}{N!}\big(v(S\cup\{i\}) - v(S)\big)$$

where S is a coalition/subset of players, N is the number of players, and $v(\cdot)$ is a value function that maps a subset of players to a real-valued payout of the game. In other words, the Shapley value was calculated by computing a weighted average payout gain that player i provided when included in all coalitions that excluded i.

6. Result

Figure 2 shows the mean absolute error (MAE) of the best prediction pipeline on the test set for each target. Note that the target values are scaled between 0 and 1; therefore, on average, the absolute error across all 12 targets varies between 11% and 22%, which does not exhibit a strong correlation between NLP features extracted from transcripts and overall communication construct survey response scores. The results also show that no single ML algorithm outperformed all others for all targets.

7. Discussion

It is possible to interpret the relatively high MAE as a lack of correlation between linguistic features and survey responses. However, upon further investigation of the data, the lack of correlation may indeed be due to the nature of the interaction within the designed scenario. A manual examination of the interaction videos and transcripts revealed that trainer-subject pairs typically greeted each other with one sentence and proceeded to the task of training without further socializing. While the trainer spoke from the script, and was therefore extremely consistent across conditions, most utterances from the subject were one- or two-word sentences such as "yes", "no", "uh-huh", or "I understand." In examining the video recordings, throughout the approximately 15 minutes of training, subjects appeared to be attentive and potentially cognitively loaded with the task of listening and absorbing the instruction. Questions from the subject were often clarifying questions or asking the trainer to repeat. After the training session, the trainer and subject did not have the opportunity for any unplanned incidental conversation. It appears that the subject simply did not have the opportunity or cognitive resources to exhibit verbal behaviors hinting at their level of engagement, rapport, trust, or sense of co-presence with the trainer, or perceived workload. Interactions that are lower in cognitive workload or are richer in social exchange may provide more opportunities for linguistic markers to manifest. In addition to the lack linguistic manifestations, the authors previously reported significant differences in objective task performance but no significant differences in social constructs such as those reported above [24] between conditions. One possible interpretation is that social interactions occur during "off-duty" time gaps between sessions, whether in-person, over video, or in virtual environments. When training sessions are highly controlled and time constrained such as our design, participants are "on-duty" and not exhibiting social behaviors. If social behaviors are desired, such as in newly formed teams, one recommendation may be to build in time gaps to afford such incidental or informal interactions to occur regardless of the communication medium.

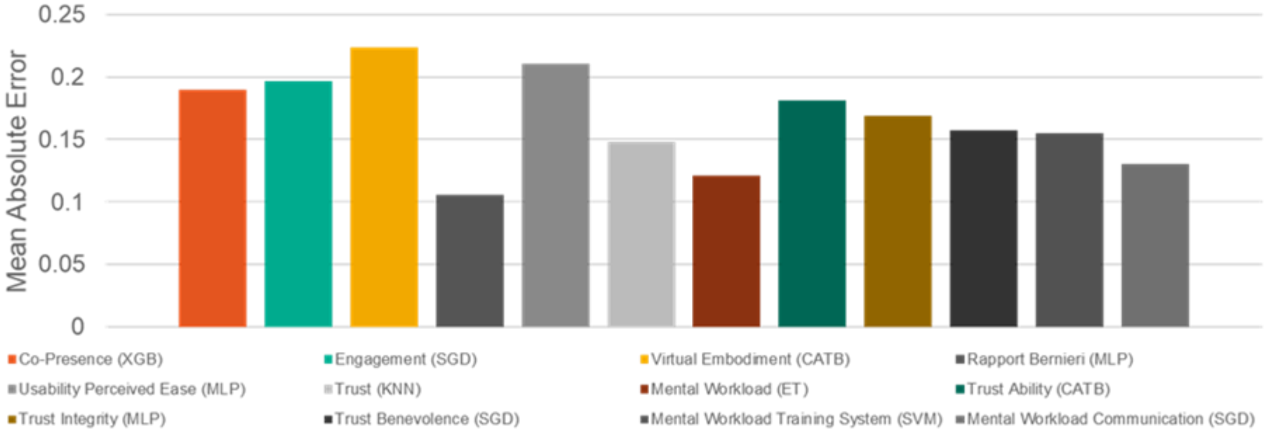

Figure 2: Mean absolute error of the best prediction pipeline for each target (ML algorithm abbreviation in parentheses).

8. Acknowledgements

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001097. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. The authors would also like to thank Jack Geddes and Jennifer Glista from Receptiviti.com for their guidance and support.

9. References

[1] Y. R. Tausczik and J. W. Pennebaker, “The psychological meaning of words: LIWC and computerized text analysis methods,” Journal of language and social psychology, vol. 29, no. 1, pp. 24–54, 2010.

[2] J. H. Kahn, R. M. Tobin, A. E. Massey, and J. A. Anderson, “Measuring emotional expression with the Linguistic Inquiry and Word Count,” The American journal of psychology, vol. 120, no. 2, pp. 263–286, 2007.

[3] C. Miller, J. Rye, P. Wu, S. Schmer-Galunder, and T. Ott, “Team psychosocial assessment via discourse analysis: Power and comfort/routine,” in International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, 2014, pp. 309–316.

[4] R. L. Robinson, R. Navea, and W. Ickes, “Predicting final course performance from students’ written self-introductions: A LIWC analysis,” Journal of Language and Social Psychology, vol. 32, no. 4, pp. 469–479, 2013.

[5] R. E. Dianiska et al., “Do You Need to Travel? Mapping Face-to-Face Communication Objectives to Technology Affordances,” Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 64, no. 1, pp. 1069–1073, Dec. 2020, doi: 10.1177/1071181320641256.

[6] J. Rozell et al., “Circuit world: a multiplayer VE for researching engineering learning,” in 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 2021, pp. 773–773.

[7] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Addison Wesley, 2011.

[8] C. D. Manning, Introduction to information retrieval. Syngress Publishing, 2008.

[9] M. E. Ireland, R. B. Slatcher, P. W. Eastwick, L. E. Scissors, E. J. Finkel, and J. W. Pennebaker, “Language style matching predicts relationship initiation and stability,” Psychological science, vol. 22, no. 1, pp. 39–44, 2011.

[10] J. Pennebaker and M. Francis, Linguistic Inquiry and Word Count. Lawrence Erlbaum Associates, Inc., 1999.

[11] C. Hutto and E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” in Proceedings of the international AAAI conference on web and social media, 2014, vol. 8, no. 1, pp. 216–225.

[12] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” Physical review E, vol. 69, no. 6, p. 066138, 2004.

[13] E. Brochu, V. M. Cora, and N. De Freitas, “A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning,” arXiv preprint arXiv:1012.2599, 2010.

[14] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[15] L. Breiman, “Random forests,” Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[16] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” Annals of statistics, pp. 1189–1232, 2001.

[17] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” Machine learning, vol. 63, no. 1, pp. 3–42, 2006.

[18] G. E. Hinton, “Connectionist learning procedures,” in Machine learning, Elsevier, 1990, pp. 555–610.

[19] C.-C. Chang, “‘ LIBSVM: a library for support vector machines,’ ACM Transactions on Intelligent Systems and Technology, 2: 27: 1–27: 27, 2011,” http://www. csie. ntu. edu. tw/∼ cjlin/libsvm, vol. 2, 2011.

[20] J. Goldberger, G. E. Hinton, S. Roweis, and R. R. Salakhutdinov, “Neighbourhood Components Analysis,” in Advances in Neural Information Processing Systems, 2004, vol. 17. Accessed: Aug. 30, 2022. [Online]. Available: https://papers.nips.cc/paper/2004/hash/42fe880812925e520249e808937738d2-Abstract.html

[21] L. Bottou and O. Bousquet, “The tradeoffs of large scale learning,” Advances in neural information processing systems, vol. 20, 2007.

[22] A. V. Dorogush, V. Ershov, and A. Gulin, “CatBoost: gradient boosting with categorical features support,” arXiv preprint arXiv:1810.11363, 2018.

[23] M. N. Coutanche, and L. S. Hallion., “Machine Learning for Clinical Psychology and Clinical Neuroscience” Cambridge University Press.(2019).

[24] R. Rocca, and T. Yarkoni, “Putting Psychology to the Test: Rethinking Model Evaluation Through Benchmarking and Prediction,” Advances in Methods and Practices in Psychological Science, vol. 4, no. 3, 2021.

[25] P. G. Fennel et al., “Predicting and explaining behavioral data with structured feature space decomposition,” EPJ Data Science, vol. 8, no. 23, 2019.

[26] J. Mell et al., “An expert-model and machine learning hybrid approach to predicting human-agent negotiation outcomes in varied data,” Journal on Multimodal User Interfaces, vol. 15, pp. 215–227, 2021.

[27] D. D. Bourgin et al., “Cognitive Model Priors for Predicting Human Decisions,” In Proceedings of the 36th International Conference on Machine Learning, 2019.

[28] L. SHAPLEY, “A Value for n-person Games. Contributions to the Theory of Games II, Kuhn, H., Tucker, A.” Princeton University Press.(1969):“Utility Comparisons and the Theory of …, 1953.