Comparing Modeling Techniques to Predict Current and Likely Next Tasks for Cognitive Workers

Sylvain Bruni, Lisa Lucia, Patrick Cummings, and Benjamin Ford

APTIMA® Human-Centered Engineering®

Problem

Reactive digital assistants fall short in high intensity environments

The pervasiveness of digital assistants in everyday life is slowly becoming a natural expectation of users of technology at work.[1] However, commercial offerings are typically reactive and intrinsically create cognitive issues, including delays in getting a correct or exploitable answer, shortsightedness of an automated reaction solely based on limited inputs, and narrowness in performance optimization.[2]

These shortcomings constitute significant limitations in the adoption of digital assistants in dynamic, time-pressured, uncertain environments (e.g., military domains or hospital intensive care units).

Approach

Proactive technology can anticipate needs

We defined proactivity as the ability to minimize cognitive overhead through the anticipation of the user's needs. We proposed to enable proactivity through the following trifecta:

- Principle 1: Sensing the tasks and workflow of the users
- Principle 2: Understanding their goals and intent
- Principle 3: Providing support in anticipation of an explicit request

Earlier efforts have demonstrated the potential for this approach.[3-10] An in-depth understanding of tasks is required to sustain Principle 1. We posited that task definitions should be multi-faceted and include (1) user activities; (2) references associated with each task; and (3) topic(s) of interest.

Method

Test six cognitive task classifiers

We conducted a data collection on a preliminary administrative use case of a cognitive worker performing travel planning activities using a checklist tool.[11] We created a topic model with two inputs: a workflow and training data. Each log line consisted of five items: (1) Time; (2) Program Type; (3) Document Name; (4) Action Type; and (5) Action Value. We used one-hot encoding and Latent Dirichlet Allocation (LDA) to vectorize each item. We built our task prediction to be classifier-agnostic, and therefore tested it with a set of different models.

The following models were all trained via skLearn:[12] K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Extra Trees Classi-fier (ERT), and Logistic Regression Cross-Validation (LogisticRegressionCV). For comparison purposes, we added two baseline models: Uniform Random and Majority Class.

The evaluation was conducted in Python[13], using NumPy[14], Pandas[15], skLearn[12], and Matplotlib[16] libraries.

Results

Achieving 75% accuracy with limited training

The KNN, ERT, and Logistic Regression CV models performed similarly on average, and better overall than the baseline models. As expected, the baseline models performed poorly. KNN's average accuracy across all participants was 71.4%, while was slightly lower on average than ERT (75.3%) or Logistic Regression (75.7%) (Figure 1).

Focus on KNN. The confusion matrix (Figure 2) shows that while KNN performed very well on some tasks (achieving an accuracy of up to 84.8%), other tasks were not classified well. For example, the Review Travel Form task, where KNN achieved an accuracy of 26.2%, was more commonly misclassified as the Fill Out Travel Form task (62.5% of the time). Also, while Review Travel Form is commonly misclassified as Fill Out Travel Form (62.5% of the time), the reverse is not true (reverse misclassification at 15.5%).

This asymmetry may suggest that there is an asymmetry in task similarity. Because some users expressed privacy-related concerns regarding activity logging, we repeated our analyses while purposely omitting some inputs. Disabling keylogging yielded a significant drop in average accuracy from 71.4% to 33% (Figure 3).

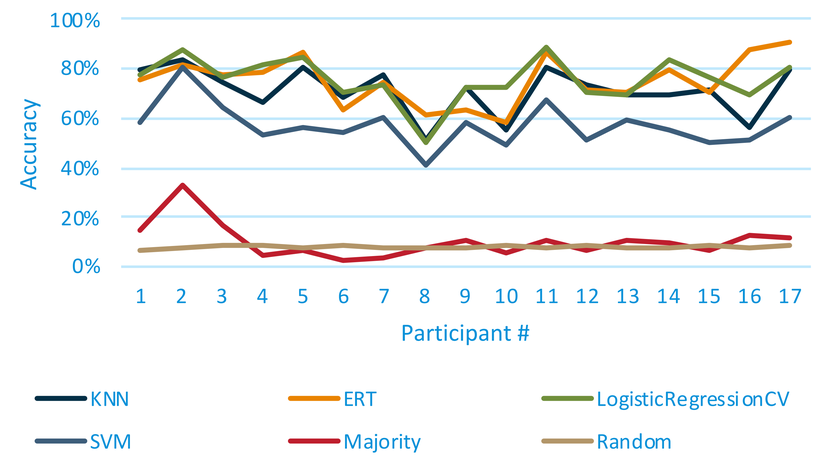

Figure 1: Test Prediction Accuracy per Participant

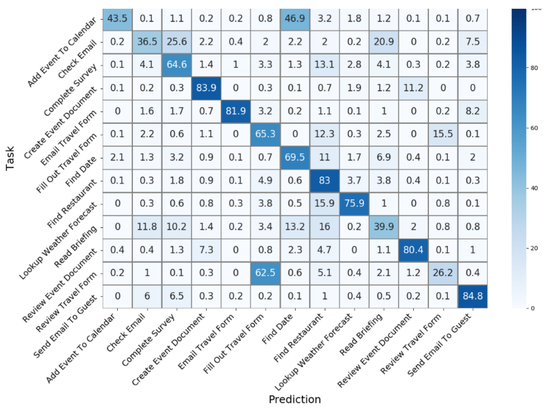

Figure 2: Confusion matrix for the cross-validation results of the full KNN model

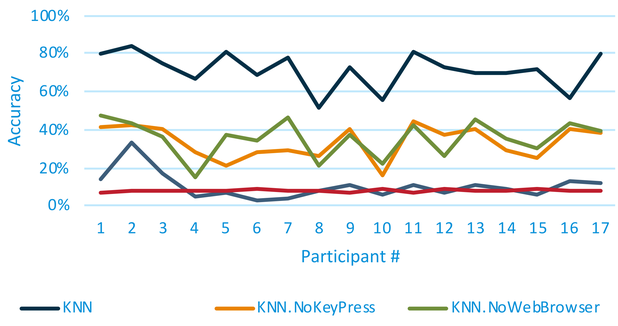

Figure 3: Test Prediction Accuracy per Participant

Next Steps

Adding Depth to Features

In future work, we plan to investigate the benefits of a feature extraction step.

Similar to how we used LDA for topic modeling, we could also experiment with temporally-explicit features such as "User's Previous Action." Another variation could reorganize training with sequences of N sample, to change the classification task to "predict the user's task based on this sequence of actions they took."

We will also research the impact of assistants on engagement versus boredom[17], stress[18,19], and fatigue.[20]

References

Please see accompanying handout.

Acknowledgements

This poster reflects work performed under subcontract to Boston Fusion Corporation on an Office of Naval Research contract sponsored by Dr. Jeff Morrison, whom the authors wish to thank.