AFRL-RH-WP-TR-2020-0136

ACTIVE SOCIAL ENGINEERING DEFENSE (ASED)

David Schroh
Uncharted Software Inc.
2 Berkeley St., Suite 600
Toronto, ON M5A 4J5

July 2020

Final Report

DISTRIBUTION A. Approved for public release; distribution unlimited.

AIR FORCE RESEARCH LABORATORY
711 HUMAN PERFORMANCE WING
AIRMAN SYSTEMS DIRECTORATE
WARFIGHTER INTERACTIONS AND READINESS DIVISION
WRIGHT-PATTERSON AFB, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

Qualified requestors may obtain copies of this report from the Defense Technical Information Center (DTIC).

AFRL-RH-WP-TR-2020-0136 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

ANDERSON.TIMOTHY.RAY.1230210728 Digitally signed by ANDERSON.TIMOTHY.RAY.1230210728 Date: 2021.03.01 06:32:34 -05'00'

TIMOTHY R. ANDERSON, DR-IV, Ph.D.
Work Unit Manager
Mission Analytics Branch
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

MURDOCK.WILLIAM.P.1048742161 Digitally signed by MURDOCK.WILLIAM.P.1048742161 Date: 2021.03.26 12:58:29 -04'00'

WILLIAM P. MURDOCK, DR-IV, Ph.D.
Chief, Mission Analytics Branch
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

CARTER.LOUISE.ANN.1230249128 Digitally signed by CARTER.LOUISE.ANN.1230249128 Date: 2021.05.10 12:51:51 -04'00'

LOUISE A. CARTER, DR-IV, Ph.D.
Chief, Warfighter Interactions and Readiness Division
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
03-07-2020	Final	September 2018 – October 2019

4. TITLE AND SUBTITLE

Active Social Engineering Defense (ASED)

5a. CONTRACT NUMBER
FA8650-18-C-7889

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
David Schroh

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
H0XA

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)
Uncharted Software Inc.
2 Berkeley St., Suite 600
Toronto, ON M5A 4J5

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
711th Human Performance Wing
Airman Systems Directorate
Wright-Patterson AFB, OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RH-WP-TR-2020-0136

12. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution A. Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
AFRL-2021-0469; Cleared 19 Feb 2021

14. ABSTRACT

Uncharted created ReCourse, a novel mixed-initiative platform to scalably coordinate, monitor and selectively moderate automated, conversational, enterprise-scale bots for defense against social engineering attacks. ReCourse combined advanced analytics with intuitive and scalable visualizations of activity to deliver threat awareness and unprecedented capability to evaluate and shape bot tactics at the global enterprise level. A human-in-the-loop system was designed to ensure ongoing adaptation to changes in adversarial tactics, and elimination of false positives, ultimately achieving dramatically improved success rates in the defense against social engineering.

15. SUBJECT TERMS
Chatbots, social engineering, information extraction, classifiers, machine learning, visualization, visual analytics

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Timothy R. Anderson, Ph.D.
Unclassified	Unclassified	Unclassified	SAR	93	19b. TELEPHONE NUMBER *(include area code)*

TABLE OF CONTENTS

LIST OF FIGURES iv

LIST OF TABLES vi

1.0 SUMMARY 1

2.0 INTRODUCTION 3

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES 4

3.1. Methods, Assumptions and Procedures 5

3.2. Program Activities 5

4.0 RESULTS AND DISCUSSION 7

4.1. Datasets and Resources 7

4.1.1 Data Normalization 8

4.1.2 Reddit Thread Dataset 8

4.1.3 Other 9

4.2. TA1 9

4.2.1 Semantic Interface for the Modeling of Ontologies (SIMON) Text Classifiers 10

4.2.2 Streaming Anomaly Detection 11

4.2.3 Malicious Message Classification 13

4.2.3.1 Malicious Message Detection 13

4.2.3.2 Speech Act Classification 13

4.2.3.3 Overview: Binary Classification Metrics 14

4.2.3.4 Traditional ML Classifiers 15

4.2.3.5 Recurrent Neural Classifiers) 16

4.2.3.6 Speech Act Detection 17

4.2.3.7 Seq2Seq and Language Model Training 18

4.2.3.8 Fusion Techniques 18

4.2.3.9 Results: Model Fusion 19

4.2.3.10 Dialogue Model Decoding Methods 20

4.2.4 Intent 21

4.2.4.1 Data 21

4.2.4.2 Annotations 22

4.2.4.3 Models 22

4.2.4.4 Evaluation 23
4.2.4.5 Results 23
4.2.5 SpamAssassin 25
4.2.6 Thug Honeyclient 26
4.2.7 Rolodex 27
4.2.8 Pigeonhole 28
4.2.9 Dyslexia 28
4.2.10 Whois 29
4.2.11 Additional Methods 30
4.2.11.1 Shapelet Classifier 30
4.2.11.2 LSTM-FCN Classifier 30
4.2.11.3 Information Diffusion Methods 31
4.3. TA2 31
4.3.1. Natural Language Style Transfer 31
4.3.1.1 Marionette 32
4.3.1.2 Ventriloquist 32
4.3.2. Automated Dialogue Modeling 33
4.3.2.1 Seq2Seq Models 34
4.3.2.2 Fused Seq2seq Models 34
4.3.3. Fingerprint 34
4.3.4 Gabby 37
4.3.5 Fezzik 39
4.3.6 Integration with Strategies for Investigating and Eliciting Information from Nuanced Attackers (SIENNA) (Bolt Beranek and Newman Inc. [BBN] Technologies) 39
4.3.7 Integration with Continuously Habituating Elicitation Strategies for Social-Engineering-Attacks (CHESS) (Hughes Research Laboratory [HRL]) 40
4.4. ReCourse 41
4.4.1. Architecture 42
4.4.1.1. Grapevine 42
4.4.1.2. Ingestion/Autoreply 43
4.4.1.3 Chute 43
4.4.1.4 Other components 43

4.4.1.4.1 Persona Management Platform (PMP) 43
4.4.1.4.2 Ibex Named Entity 43
4.4.2 Design 43
4.4.2.1 Engagements View 46
4.4.2.2 Bot Engagement Workflow 46
4.4.2.3 Conversations View 48
4.4.2.4 Experimental Sandbox 48
4.5. Program Results 48
4.5.1 Orchestration and Deploymen
4.5.1.1 Docker 48
4.5.1.2 Swarm 48
4.5.1.3 Kubernetes 49
4.5.2 Dry Run Evaluation 49
4.5.3 Program Evaluation 51
4.6 Publications 55
4.7. Open Source Repositories 56
5.0 CONCLUSION 58
6.0 RECOMMENDATIONS 59
6.1 Data Annotation 61
6.1.1 Task-oriented Dialogue 61
6.1.2 Entity-Specific Sequence Labeling 62
7.0 REFERENCES 64
8.0 LIST OF ABBREVIATIONS, ACRONYMS AND SYMBOLS 71
APPENDIX A - Investigating Language Model Fusion Methods for Open-Domain Dialogue Systems 74

LIST OF FIGURES

Figure 1. ReCourse Project Timeline and Deliverables..... 6

Figure 2. Datasets..... 7

Figure 3. Network Visualization of User Interactions in the Curated Reddit Thread Dataset 8

Figure 4. Accuracy of TA1 Methods. 9

Figure 5. Confusion Matrix For Binary Text Classifier on Held-Out Data From March 2019 ASED Dry-Run. 11

Figure 6. Metrics for Binary Text Classifier on Held-Out Data from March 2019 ASED Dry-Run. 11

Figure 7. Confusion Matrix for Streaming Anomaly Detection on Held-Out Data from March 2019 ASED Dry- Run. 12

Figure 8. Metrics for Streaming Anomaly Detection on Held-Out Data from March 2019 ASED Dry-Run 12

Figure 9. RRCF Data Structure: Points Closer to the Root are more Anomalous. 12

Figure 10. RRCF Applied to Real-world Twitter Streaming Data. 13

Figure 11. Ten-Fold Cross Validation Results for the Binary Email Classification Task. 16

Figure 12. Results for Malicious Message Classification Using RNNs Utilizing Both Single-Task and Multi-Task Training (best scores in bold). 17

Figure 13. Results for Multiclass Speech Act Classification across 17 Speech Acts in the SWDA Corpus (best scores in bold). 17

Figure 14. Performance Metrics for our Proposed Fusion Methods - NS, WS and GP. 19

Figure 15. Loss Convergence for Different Fused Dialogue Models Tested in our Conversational Modeling Experiments. 20

Figure 16. Examples from our Systems Showing Generic Phrases from Greedy Decoding (left) and Increased Diversity in Responses Utilizing Beam Search (middle) and Top-K Sampling (right) Approaches. 21

Figure 17. Comparison of BERT and HAN Models on Relevant Labels in the Speech Dialog Act 23

Figure 18. Training Curves of Final Models for Evaluated on a Mix of Noisy and Hand-Labeled Data. 24

Figure 19. Mapping of Model Classification Labels to Labels in Campaign Email Dataset. 24

Figure 20. DistilBERT (trained on noisy data) Evaluated on Campaign Emails. 25

Figure 21. SpamAssassin Classification Accuracy 26

Figure 22. Accuracy of Thug Honeypot. 27

Figure 23. Accuracy of Rolodex 27

Figure 24. Pigeonhole Results on the 'Historical' Email Dataset, Containing mostly FRIEND (non-malicious) Emails Received by Five ASED Volunteers. 28
Figure 25. Pigeonhole Results on the 'March Dry Run' Email Dataset, Containing both FRIEND and FOE Emails. 28
Figure 26. Accuracy of Dyslexia TA1 Classifer. 29
Figure 27. Accuracy of Whois TA1 Classifier. 30
Figure 28. Style Transfer Examples from Marionette. 32
Figure 29. Style Transfer Examples from Ventriloquist. 33
Figure 30. Fingerprint Site as it Appears to an Unassuming Attacker. 35
Figure 31. Fingerprint Results Surfaced in ReCourse. 35
Figure 32. Example Data Points Obtained from a Fingerprint. 37
Figure 33. Sample Conversation with Gabby. 38
Figure 34. Comparison of Gabby and Fezik against Ground Truth Responses. 39
Figure 35. SIENNA-Supplied Suggested Reply for HITL Scenario 40
Figure 36. Example Game-Theory-Based Strategies from CHESS (image from HRL). 40
Figure 37. Example CHESS Results being Surfaced in ReCourse. 41
Figure 38. ReCourse Functional Architecture. 41
Figure 39. ReCourse System Architecture. 42
Figure 40. User-Driven Nomenclature for Identities. 44
Figure 41 User-Driven Nomenclature for Plans 45
Figure 42. Visual Vocabulary for States and Actor Types 45
Figure 43. Dry Run Evaluation TA1 Results (image from JPL). 50
Figure 44. Full Evaluation TA1: Friend/Foe Identification (image from JPL). 51
Figure 45. TA1: Full Evaluation Friend/Foe Identification - Additional Context (image from JPL). 51
Figure 46. Full Evaluation TA2 Results (image from JPL). 52
Figure 47. Full Evaluation TA2 Results Continued (image from JPL). 52
Figure 48. Full Evaluation TA2 Attribution Example (image from JPL). 53
Figure 49. Full Evaluation TA3 Attributes obtained by ReCourse (image from JPL). 53
Figure 50. Dialogue Evaluation: Flags Captured by Bot Systems (image from JPL). 54
Figure 51. Dialogue Evaluation: Flags Captured by Human/Bot Systems (image from JPL). . 54
Figure 52. Dialogue Evaluation: Qualitative Patterns from TA3 (image from JPL). 55
Figure 53. Annotation Interface for Dialogue Negotiation Task (Lewis 2017). 62

Figure 54. Example of Word-Level Tagging for Proposed ASED-specific Entities using Standard BIO format Common to NER Tasks. 63

LIST OF TABLES

Table 1. Open Source Repositories..... 57

1.0 SUMMARY

For the Active Social Engineering Defense (ASED) program, Uncharted Software Inc. created ReCourse, a novel mixed-initiative platform to scalably coordinate, monitor and selectively moderate automated, conversational, enterprise-scale bots for defense against social engineering attacks. ReCourse combines advanced analytics with intuitive and scalable visualizations of activity to deliver threat awareness and unprecedented capability to evaluate and shape bot tactics at the global enterprise level. A human-in-the-loop (HITL) system ensures ongoing adaptation to changes in adversarial tactics, and elimination of false positives, ultimately leading to dramatically improved success rates in the defense against social engineering.

ReCourse is a combined technical area (TA) 1+TA 2 platform for situational awareness (SA) of the attack surfaces of the enterprise; scalable HITL bot and persona management for both detection and investigation; cross-channel monitoring and bot dialogue for detecting attacks; and automated and semi-automated cross-channel actor engagement for investigative information elicitation. ReCourse creates new, generalizable, scalable methods for inclusion of human cognition and feedback in orchestrating novel conversational agents across enterprise channels.

The state of the art in (SOTA) chatbot systems "produce short, generic responses that lack diversity [Sordoni 2015; Li 2015]. Even when longer responses are explicitly encouraged they tend to be incoherent or contradictory" [Shao 2017]. Uncharted applied a novel modeling, usability and visualization experience to automated dialogue systems and designed scalable techniques that allow orchestrators to confidently monitor and guide large networks of bots to discover actor goals and identities.

The Uncharted team included proven Defense Advanced Research Projects Agency (DARPA) collaborators in Qntfy and Yonder (aka New Knowledge aka Popily), and leveraged unique expertise in invention of HITL systems of influence for large populations

In addition to providing a complete end-to-end solution, our team collaborated with other TA1 and TA2 performers to integrate additional best-of-breed analytics into ReCourse. Lightweight, practical application programming interfaces (API) were defined for integration and interoperability. Open source, baseline detection and investigation modules were combined with the integration of other performer technologies. Uncharted also worked closely with the TA3 performers to ensure effective evaluation of an HITL approach for automated defense, using integrated capabilities to measure performance.

We built on DARPA XDATA, Memex and QCR mixed-initiative techniques including human-in- the-loop approaches for increasing combined human-system effectiveness of building classifiers; multi-modal models for persona building and entity linking; and semi-automated methods for expanding networks of identifiers for "know your customer" (KYC).

Yonder presented research components for supervised classification techniques, graph-based multi-channel models, and natural language style transfer. Of these, Yonder submitted three-time series classification techniques and a binary/multi-class text classifier for the supervised classification techniques. The dry-run and evaluation results from these methods encouraged further exploration into unsupervised techniques because of the lack of representative training data. Yonder also built two comprehensive style transfer systems with two to four styles and was the only team actively working on this problem. In addition to these components, Yonder also

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

began work on named-entity recognition, information diffusion methods, and the persona management platform described in this report. Lastly, Yonder submitted a social media dataset with multiple back and forth interactions to allow the other ASED performers to explore ask-detection techniques.

Qntfy contributed technology to the ReCourse tool from two major research and development efforts.

- Methods for passive detection of malicious messages were designed and evaluated and under TA1.
- Methods of training a task-based dialogue were developed under TA2.

Qntfy's TA1 research focused on email attack detection and speech act classification. Several models are discussed and evaluated, including a multi-task approach. Qntfy's TA2 work included sequence-to-sequence (seq2seq) dialogue model research, in combination with language models and alternative decoding strategies.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

2.0 INTRODUCTION

This document constitutes a Scientific and Technical Report for the ReCourse project. The goal of this document is to summarize objectives, technical strategy and approaches, key results and accomplishments, and lessons learned for the future.

Uncharted Software Inc. served as the prime contractor, with Qntfy and Yonder sub-contractors.

Yonder proposed several valuable components to the ASED program to detect and mitigate advanced spear-phishing threats. These included supervised classification techniques on message frequency time-data and/or content, rule-based analysis of anomalies in message targeting and graph-based models to detect anomalous cross-channel activity, and a comprehensive natural language style transfer system. Additionally, Yonder proposed a persona management platform to support the curation, coordination, and programming of autonomous and semi-autonomous cross-channel conversational agents. Furthermore, Yonder planned to use its extensive social media data collection to curate relevant datasets to share with the ASED performers, including instances of multiple back-and-forth interactions where one party is trying to get another party to perform an action, and broadcasts where the poster is trying to get others to perform specific actions.

Research conducted by Qntfy primarily supported two aspects of the ReCourse system: attack detection analytics (TA1) and attack investigation via dialogue (TA2). Initial attack detection research focused on single document analytics, predominantly using text features. These models sought to classify emails as non-/malicious and the speech acts contained therein. Intended future work on attack detection would have emphasized cross-attack analytics in order to identify commonalities between attacks and the vectors used to deliver them. Initial attack investigation research explored developing chit-chat style dialogue systems for the purposes of wasting an attacker's time. These models were intended to extend conversations identified by malicious content classification with human realistic dialogue utterances. Future work would have built upon these models in order to establish rapport with an attacker and ultimately elicit additional information. We describe this type of task as a "distant goal," where potentially a range of conversation outcomes are acceptable and require multiple dialogue turns to reach.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

ReCourse provides a combined TA1+TA2 platform for leveraging enterprise users together with networks of automated bots for detection, defense and source of social engineering attacks. The combined total of all communications (e.g. phone, email, social media) into an enterprise provides too large an attack surface for humans to monitor unassisted. ReCourse automatically characterizes, organizes and visualizes the enterprise's attack surface. Automated social engineering attack classifiers flag suspicious or known actor communications by analyzing cross-channel communication within and across the enterprise. Semi-automated bots intercept suspicious communications and embody the personas of the target user to verify identity, or solicit identifying information from actors, or tie up their resources via distraction tactics.

SA of the Attack Surfaces at Enterprise Scale. Enterprise users cannot verify every communication they receive. ReCourse bots do so automatically. Persona models and communications risk scores are used to prioritize open source scraping tools to follow links. Public cloud services are used to verify phone numbers and email addresses. The knowledge base of linked personas and blacklists is used to detect previously identified malicious content. Similarly, enterprise identity services and whitelists are used to verify the identity of the target personas. ReCourse bots share information through the searchable persona models.

ReCourse uses Natural Language Processing (NLP) and automated text and metadata extraction to collect identifiers and characteristics of potential actors and potential target personas. Scalable tile-based visual analytics (TBVA) combine with chart elements to form rich dashboards for assessing enterprise threats and attack vectors. Communication events are scored for attack risk, using classifiers built from persona and communication features. Operators can use ReCourse to understand signatures of social engineering attacks using many dimensions and monitor detection against attack types, sources and targets, and changes in behaviors over time.

HITL Persona Management. ReCourse provides an aggregate view of the communications for the personas in the enterprise, such as email, phone and social media. Enterprise Persona Management routes communications to the correct target persona and selects or generates a channel for each persona's sub-identity. As sources and communications go through automated verification and risk scoring, channels in turn get risk scores, allowing users to quickly identify high-risk communications, increasing their trust in "asks" from those channels. Risk scores are generated by classifying aggregate data across similar channels, based on common features. Persona Management provides curation and management, paired with standards for identity and authorization, to safeguard private user information.

Cross-channel Monitoring for Detection of Social Engineering Attacks. Often, detection and verification of actors is not possible through automated, out-of-band techniques. In those cases, a bot is required to interactively validate the identity of the communication source and model attack risk. ReCourse creates easy-to-use interfaces for enterprise users to choose from, curate and edit strategies and challenges that assist in active detection. A novel system for easy dialogue management allows untrained users to quickly pick challenges as easily as they would solve "captchas" and other current web techniques for soliciting input.

Semi-automated Cross-channel Engagement with Social Engineering Actors. Once an attack is identified, a coordinated system is required to pick and launch active investigation. ReCourse provides a set of novel interfaces for operator visibility into active investigations, strategies that bots are employing (direct and indirect) and the status of elicitation efforts. An

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

overview of bot performance provides understanding of investigative bots, with key performance indicators for elicitation (such as identifiers collected, resources used, information gain, risk exposure).

ReCourse summarizes communications and alerts, presents detected threats for review, and monitors and potentially guides bot responses.

Continuous HITL System Adaptation to Changes in Adversarial Tactics. We created novel mechanisms for adaptive, HITL dialogue. An interface for enterprise defenders allows introspection into the strategies and dialogues of bots and a framework for humans to participate in the same dialogue. ReCourse orchestrators can shape current strategy and bot personas, influence paths of conversation, or tune elicitations and distractions.

Searchable Knowledge Base of Shareable Personas. ReCourse aggregates known actor information into a scalable knowledge base populated over time from verification and investigation bots, as well as from blacklists, social networks and whitelists of trusted actors. The knowledge base guides risk scoring and bot behaviors and serves as a first-class resource for enterprise clients to understand the networks of actors that are contacting them.

Real-time, Enterprise-Scale Streaming Architecture. ReCourse mediates communications between potential actors and target personas, streaming communications, and providing near- and real-time analytics. A platform configured as a buffered event stream and job queue provides the backbone to stream and filter communications through the persona fabric to enterprise users; create target channels; route and manage responses and ongoing dialogues. A cloud-based virtual container pool, elastically provisioned via Docker Swarm, provides resources for active bots.

3.1. Methods, Assumptions and Procedures

Creative insights were guided by structured interviews, cognitive task analysis, task observation, and informal user feedback. Collaboration with Subject Matter Experts (SME) users in a "double helix" model of evolution was essential to the development of relevant innovative capability [Wright 2002].

Uncharted's approach involved monthly design and testing and quarterly focus group sessions with representative users, facilitated by the government team and TA3 performers.

In designing for end users, our interdisciplinary methods derived from Jacob Nielson, Ben Schneiderman, Don Norman, Colin Ware, Christopher Wickens, Stuart Card and other pioneers. Example ease-of-use principles included recognition rather than recall, simplifying task structure, error prevention and recovery, visibility of system status, consistency, user control, etc. Ease of use does not mean foregoing professional tools and powerful functions used by domain experts for difficult problems. But ease of use does mean pick up and use with minimal training.

3.2. Program Activities

The first year of the program consisted primarily of a kickoff meeting, a "dry-run" of the evaluation framework in March 2019, and then a baseline system evaluation in August 2019.

The first quarter was building an end-to-end system and coordinating with the TA3 evaluation team to design and implement the technical infrastructure for evaluation. The second part of the year was spent focusing on initial research challenges and integrating promising analytics and technologies into the ReCourse platform for the first evaluation.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

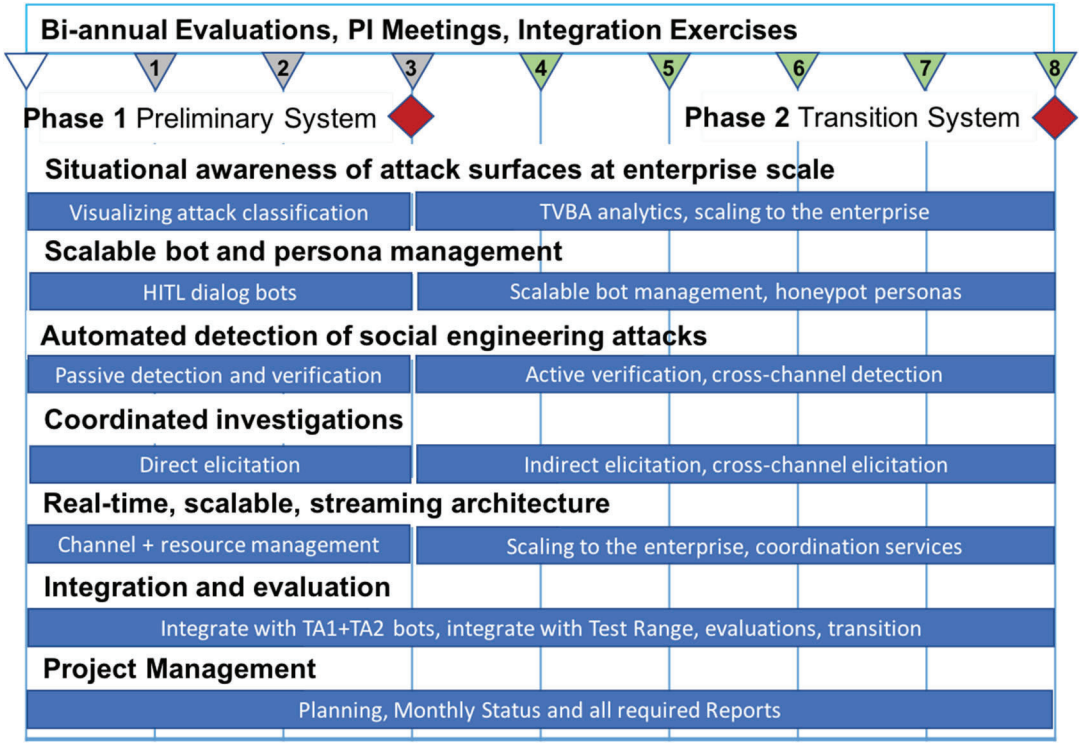


Figure 1. ReCourse Project Timeline and Deliverables

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

4.0 RESULTS AND DISCUSSION

The following is a summary of progress results and discussion by research efforts.

4.1. Datasets and Resources

Many different datasets and other resources were collected, cleaned and processed. These were used as training and test data and shared with other researchers.

Dataset	Location
JPL Historic Dataset	https://ased.io/data/jpl/historic/
JPL Dry Run Dataset	https://ased.io/data/jpl/dry-run/
TA3 Datasets	https://ased.io/data/TA3_May_Campaign/ https://ased.io/data/TA3_June_Campaign/ https://ased.io/data/TA3_July_Campaign/
JPL Abuse Dataset	https://ased.io/data/jpl/JPL_Abuse_2017/
APWG Dataset	https://ased.io/data/apwg/
Enron Email Dataset	https://www.kaggle.com/wcukierski/enron-email-dataset
Fraudulent Email Corpus	https://www.kaggle.com/rtatman/fraudulent-email-corpus
SMS Spam Collection Dataset	https://www.kaggle.com/uciml/sms-spam-collection-dataset
Reddit Coarse Discourse Dataset	https://github.com/google-research-datasets/coarse-discourse
Switchboard Dialogue Act Corpus	http://compprag.christopherpotts.net/swda.html
Twitter Customer Support	https://www.kaggle.com/thoughtvector/customer-support-on-twitter
Reddit Dataset from PolyAI	https://github.com/PolyAI-LDN/conversational-datasets
Personachat	https://github.com/facebookresearch/ParlAI/tree/master/projects/personachat
DailyDialog	https://www.aclweb.org/anthology/I17-1099

Figure 2. Datasets

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

In addition to these sources, the team made use of:

- Online blacklists of phishing URLs (updated hourly; includes REST API) at https:// www.phishtank.com/api_info.php
- Anti-Phishing Working Group: https://apwg.org/

4.1.1 Data Normalization

Email-based datasets (e.g., Enron, Fraudulent, Jet Propulsion Laboratory [JPL] datasets) were processed by ReCourse in raw electronic email [EML] format. In some cases, the dates of raw emails were modified to simulate a steady stream of incoming messages to a given ReCourse account (i.e., *X* messages received /day / account).

Dialogue-based datasets (e.g., Reddit, Personachat, DailyDialog) were normalized into dialogue turns: *message* and *response* pairs. The *message* text representing an incoming message and the dialogue *response* representing the ground truth text. For example, the raw Reddit data contains topic *threads* of replies to a given post; these were parsed into dialogue message/response pairs.

The normalized dialogue datasets were used for Bot training and evaluation. A multi-turn dialogue (conversation) was represented by a sequence of dialogue pairs.

4.1.2 Reddit Thread Dataset

We collected and curated 151 Reddit threads mentioning the words “boycott” or “protest” that include significant user interaction as measured by the mean clustering coefficient in the user sub-networks formed by users in the threads. There are 77,855 posts in this dataset, which was shared with the broader ASED program for training dialogue models and researching ask-detection techniques.

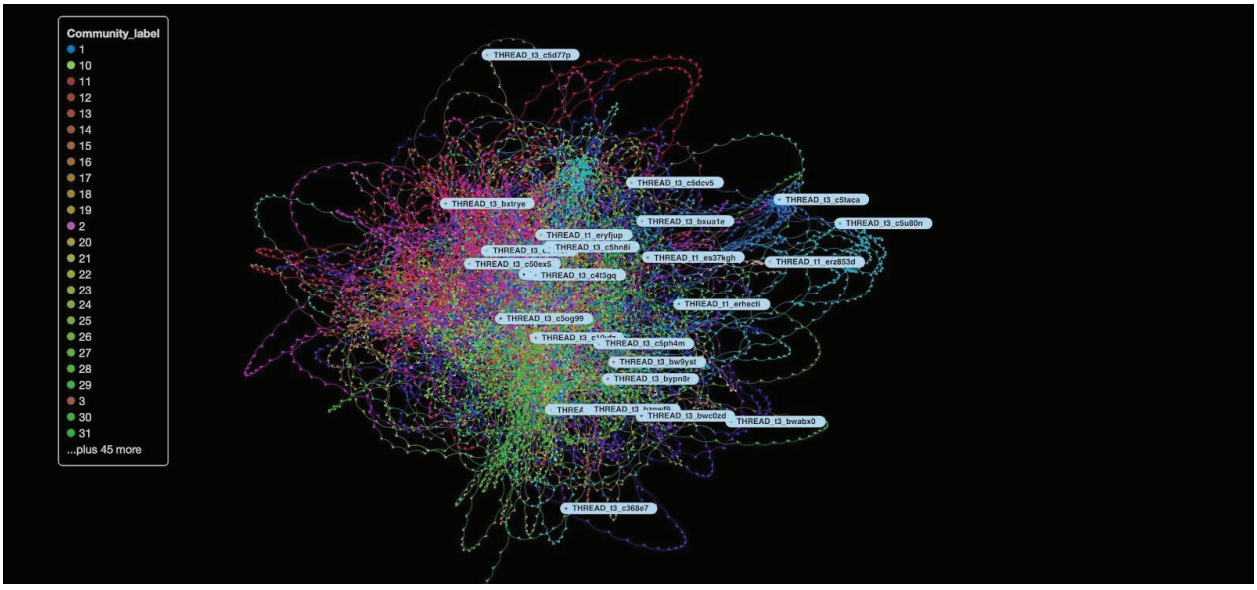


Figure 3. Network Visualization of User Interactions in the Curated Reddit Thread Dataset.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

4.1.3 Other

Qntfy initiated work on providing data infrastructure support to the program. This included the provisioning of cloud resources on Amazon Web Services and the build-out of a Kubernetes cluster. Work was initiated to deploy and configure a processing pipeline for the collection, normalization, enrichment, and storage of communications data, beginning with public social media content. Qntfy coordinated with other program performers to collect requirements for API changes and additional capabilities to meet their data collection and processing needs, thus providing a single platform for collection and processing program-wide. Qntfy also assessed the feasibility of integrating email collection, normalization, and processing, in coordination with other program performers.

4.2. TA1

Our approach to TA1 classification and reporting used a microservice architecture that was orchestrated by Grapevine. As a message comes in on the Kafka topic, Grapevine normalizes it into the ReCourse format and then uses general-purpose Remote Procedure Call (gRPC) to concurrently send it to a number of services that perform friend/foe classification, motive detection, Named-Entity Recognition (NER), time series analysis, clustering, etc.

Those results were collected, and an ensemble approach was used to then make a final determination on whether or not the message is considered an attack. That decision was sent to a Structured Threat Information Expression (STIX) endpoint and the message itself was augmented with both the decision and the results from the microservices before being sent off to the ReCourse Knowledge Base.

After the dry-run evaluation, we added a number of new TA1 services in addition to iterating on its ensemble approach. As a result, its performance on the dry-run dataset improved significantly and in the program evaluation the ReCourse system had the highest TA1 accuracy while maintaining a low false alarm rate.

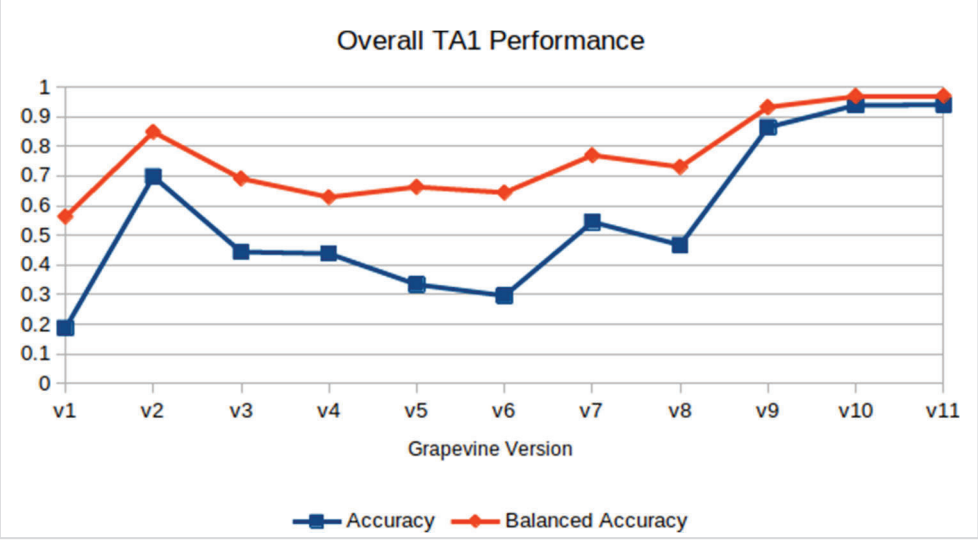


Figure 4. Accuracy of TA1 Methods.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

4.2.1 Semantic Interface for the Modeling of Ontologies (SIMON) Text Classifiers

We hypothesized that the text of incoming messages would contain useful signals for classifying spear-phishing messages. Therefore, our first line of TA1 development work focused on text classification models using a state-of-the-art Convolutional Neural n etwork (CNN) - Long Short-Term Memory (LSTM) neural network architecture [Zhou 2015]. Specifically, SIMON consists of two deep neural networks—one which encodes each individual sentence and a second which encodes the entire document.

The sentence encoder consists of three convolutional stacks (1-D convolution, dropout, max-pooling) and two LSTM layers that process the representation (one forward and one backward) before concatenation and a final dropout layer. The sentences are encoded in parallel, after which the document encoder applies an attention operation to each sequence of sentences. Then, the representation is passed to two smaller LSTM layers (again, one forward and one backward) before concatenation and dropout layer. Finally, two linear layers (separated by a dropout layer and a softmax activation function) are used for classification.

Initially, the model is trained for binary classification using a diverse set of ham and spam examples. Specifically, the ham examples come from both the Enron email corpus [Klimt 2004] and a set of National Aeronautics and Space Administration (NASA) JPL emails supplied by volunteers. On the other hand, the spam examples come from the 419 spam corpus [Radev 2008], an email abuse dataset curated by JPL, and high-quality attack examples manually created by Thomson Reuters Special Services (TRSS). The general examples (i.e., Enron and 419) are meant to regularize the classifier by expanding the data manifold, while the specific examples (JPL and TRSS datasets) approximate the evaluation data with as much fidelity as possible. As expected, there were many more examples from the Enron and 419 corpora than from the curated JPL and TRSS datasets.

Additionally, transfer learning is employed to bootstrap a multi-class classifier from the original binary (spam/ham) classifier. The sentence encoder, document encoder, and first dense layer are held fixed while the final dense layer is unfrozen. The training data for multi-class classification consists of nine classes (acquire credentials, acquire PII, annoy recipient, build trust, elicit fear, access social network, gather general information, get money, and install malware), where the samples are the same as those from the binary training regime, but with more specific (and for some samples, multi-label) annotations.

Finally, the classifier iteratively finetunes its parameters for N iterations after a batch of M benign messages have been seen during the training phase in program evaluations. Iterative fine-tuning is meant to prevent model degradation by conservatively adjusting the model parameters over time to effectively separate the most recent batches of malignant and benign examples.

Figure 5 and Figure 6 present the results of applying the SIMON binary text classifier to a held-out test set from the March 2019 ASED dry-run. The dry-run consisted of 983 total samples, 23 of which were high-quality attack messages that TRSS manually designed. Unfortunately, the classifier was unable to identify any of the attack messages from the test-set, despite presenting an accuracy of above 99% and numerous true positives on the validation set. This emphasizes the difficulty of identifying attack messages manually designed by TRSS and the potential misalignment between the attack messages in the test set and those on which the classifier was trained. However, the classifier does present a very low false positive rate, which is a desirable feature for a spear-phishing classifier.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

	True Ham	True Spam
Predicted Ham	945	23
Predicted Spam	15	0

Figure 5. Confusion Matrix For Binary Text Classifier on Held-Out Data From March 2019 ASED Dry-Run.

Metrics	
Accuracy	0.961
Precision	0.000
Recall	0.000
F1 Score	0.000

Figure 6. Metrics for Binary Text Classifier on Held-Out Data from March 2019 ASED Dry-Run.

4.2.2 Streaming Anomaly Detection

The dynamic nature of communication on social messaging platforms motivated a second line of TA1 research around streaming approaches to anomaly detection. The goal of this approach was to design components that can continuously update their representations to reflect the contemporary state of the messaging platform ecosystem.

One component generated by this research agenda was an implementation of the robust random cut forest (RRCF) streaming anomaly detection algorithm [Guha 2016]. The algorithm maintains a forest of trees data structure in which the depth of a data point is inversely proportional to its anomaly score. Thus, data points that are closer, on average, to the roots of trees are more anomalous. The anomaly score of a new data point is proportional to the average change in complexity that results from inserting the new data point into each tree in the forest.

For the purposes of TA1 classification in the ASED program, this component trains separate anomaly detection "forests" on each individual account being protected. The features at each timestep consist of time features (day of the week, hour of the day, minute of the hour, and second of the minute) and a 128-dimensional feature vector produced by the SIMON model after its first dense classification layer.

When a new message is sent to the classifier, the data structure for each unique account receiving the message processes the features and produces an anomaly score. These anomaly scores are then weighted by the number of points that each data structure processed (corresponding to the number of messages each account saw) to produce an overall anomaly score. Finally, this overall anomaly score is thresholded to generate a classification. Figure 7 and Figure 8 present the results of applying the RRCF streaming anomaly detection classifier to a held-out test set from the March 2019 ASED dry-run. In contrast to the SIMON text classifier, we note that the streaming anomaly detection component was able to correctly identify 4 out of the 23 attack messages in the test set, generating a recall of 0.174. However, this increase in recall came at a price: a significantly higher false positive rate. Overall, the streaming anomaly detection

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

component presented a slightly higher F1 score on the held-out test data than the text classifier.

However, the main takeaway is that neither TA1 component had much success at identifying manually generated TRSS attack emails. These deficiencies emphasized the need for more representative training data and encouraged deeper investigation into unsupervised and semi-supervised approaches to classification, which were upcoming on our TA1 roadmap.

	True Ham	True Spam
Predicted Ham	678	19
Predicted Spam	282	4

Figure 7. Confusion Matrix for Streaming Anomaly Detection on Held-Out Data from March 2019 ASED Dry-Run.

Metrics	
Accuracy	0.709
Precision	0.014
Recall	0.174
F1 Score	0.026

Figure 8. Metrics for Streaming Anomaly Detection on Held-Out Data from March 2019 ASED Dry-Run.

.

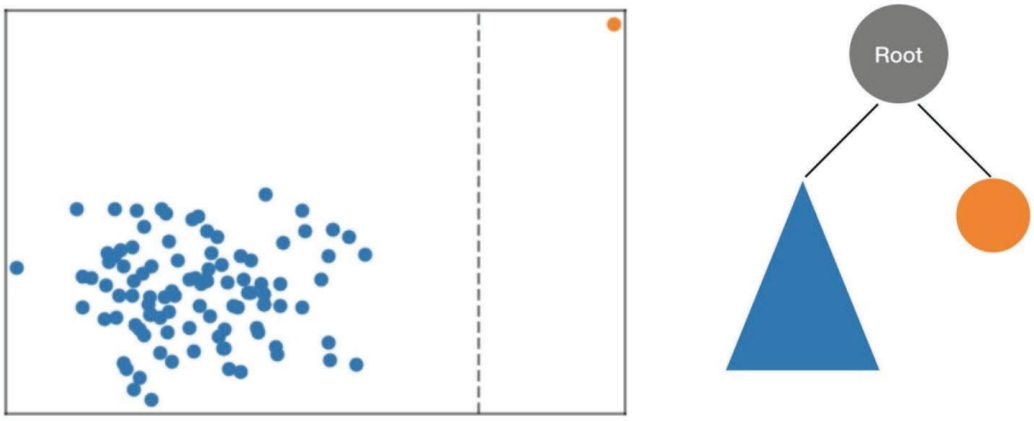


Figure 9. RRCF Data Structure: Points Closer to the Root are more Anomalous.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

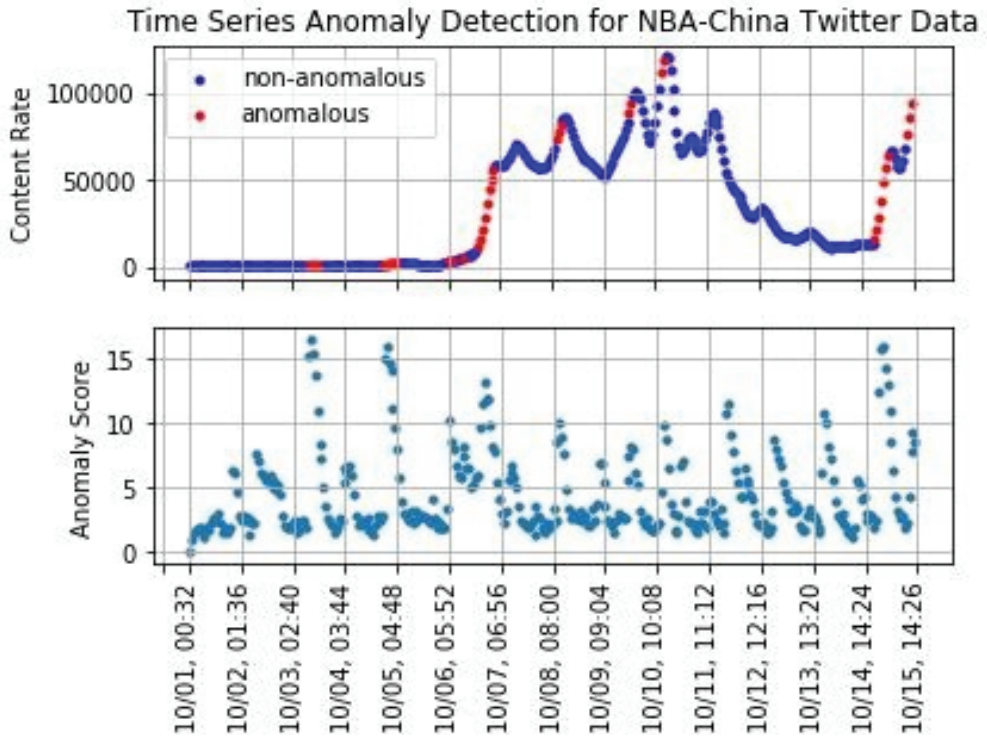


Figure 10. RRCF Applied to Real-world Twitter Streaming Data.

The top plot shows the rate of content over time. Timesteps with an anomaly score above a z-score threshold are marked as anomalous. The bottom plot shows the anomaly score generated by the RRCF algorithm at each timestep.

4.2.3 Malicious Message Classification

As part of our TA1 detection analytics, we investigated developing several classifiers as an automated way to triage incoming messages based on their perceived maliciousness. That is, since the overall goal of ASED systems was to automatically engage with malicious actors, the first step in this process was to automatically identify messages with a malicious intent. These messages could then be automatically routed to further downstream processing such as more fine-grained classification systems or automated dialogue systems to engage with the email sender.

4.2.3.1 Malicious Message Detection

For initial message classifiers, we built off of the rich set of previous work carried out in spam detection. In previous work in the NLP and machine learning (ML) literature, spam detection has often been approached as a binary classification task where the goal is to develop a classifier that assigns one of two labels to any given document - either “spam” or “ham” (e.g., messages that should be filtered from user, or those that should be surfaced to a user respectively). Our initial efforts in this area of work compared several ML classifiers trained and evaluated on the well-studied Enron email corpus [Cohen 2015]. In addition to being linked to previous work in document classification, the corpus also has the advantage of over 500,000 emails to train robust, high-capacity models.

4.2.3.2 Speech Act Classification

In addition to developing binary classifiers to detect and triage malicious incoming messages, we also experimented with multiclass classifiers to identify the perceived intent of a message. That

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

is, in addition to identifying messages that may be a threat to an end-user, we wanted additional, more fine-grained, topical information about the intent of messages. For example, malicious messages may seek to elicit very different information from a user, ranging from surface-level details such as first and last name to more damaging information such as bank account details. Past work in this area has utilized several approaches including simple Naive Bayes classifiers [Grau 2004] to recurrent neural networks that are able to more accurately capture conversational context [Bothe 2018].

As an approximation to this, we developed several multiclass classifiers trained on "speech act" labels. Speech acts, developed out of the linguistics and pragmatics community, are used to characterize the purpose or goal of a statement within a conversation. In our case, speech acts are useful in understanding the purpose of an incoming message (e.g., a malicious message might contain a general question "Where do you bank?" or a more pointed directive "Give me your account information.") Accurately identifying these intents could have positive implications not only for the automatic triage of these messages, but also for downstream dialogue modeling tasks discussed below.

To train these models, we used the Switchboard Dialogue Act Corpus (SWDA)[Potts 2011]. This dataset consists of over 200,000 utterances transcribed from two-party spoken conversations and contains over 40 dialogue/speech act labels to be used for training classifiers. While not all speech acts within the data are relevant to the program goals (e.g., spoken non-verbal speech acts such as "throat-clearing" are not relevant to the text-only domain of emails), the large amount of data allowed for experimentation with a wide range of classifiers for this task.

For the TA1 task of detecting malicious messages, we began by training binary classifiers on a large-scale email corpus consisting of messages from the Enron corpus combined with a large set of spam emails soliciting money or bank account information. This formed an initial dataset to train binary classifiers to automatically detect these malicious spam emails and separate them from benign emails. Results from these models are presented in the sections below.

4.2.3.3 Overview: Binary Classification Metrics

To evaluate these binary classifiers, we utilized several metrics from the machine learning literature. Before discussing individual metrics, it is helpful to understand some common terminology. We use the term **true positive** (TP) to refer to instances within a dataset that are *predicted* to belong to class P that are *actually members* of class P, where class P is the positive class. Likewise, we use the term **true negative (**TN**)** to refer to instances within a dataset that are *predicted* to belong to class N that are *actually members* of class N. Conversely, we use the term **false positive** (FP) to refer to instances *predicted* to belong to class P that *actually* belong to class N, and the term **false negative** (FN) to refer to instances *predicted* to belong to class N that *actually* belong to class P.

Precision (also sometimes referred to as positive predictive value) measures what proportion of a classifier's predicted positive instances are true positives. We can express precision in the simple formula TP / (TP + FP), where TP and FP refer to true positives and false positives respectively.

Recall (also sometimes referred to as sensitivity) measures what proportion of true positives in the dataset are correctly identified by the classifier. Recall can be expressed in the following formula TP / (TP + FN), where FN refers to false negatives.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

F1-Score is a metric that combines both precision and recall into a single value using a weighted mean that places equal weight on precision and recall. Given values computed for precision and recall, the F1-score is computed by $2 \times (P \times R) / (P + R)$, where P is precision and R recall.

Average Precision is a more nuanced metric that considers the classifier's confidence, typically expressed as the probability of a data-point belonging to the positive class. Conceptually, to compute average precision we take a set of data-points and rank them by the probability that each data-point belongs to the positive class assigned by the model. We then descend this ranking and at each point where recall increases, we measure precision thus producing a list of precision scores. To compute the final metric, we take the average of these precision scores. The score can be computed via k=1|D|(P(k) I(k))|D|, where |D| is the size of the dataset, P(k) is the precision measured at rank k, and I(k)is the indicator function denoting whether an instance belongs to the positive class.

Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) is another metric that takes probability or model confidence into account by measuring the tradeoff between true-positive rate vs. false-positive rate. Often this metric is presented as a curve where any point along the curve tells us what true-positive rate can be expected from a model at a given selected false-positive rate. The AUC is a single summary number representing the area under this curve. Given a set of ground-truth binary labels and their associated probability estimates from a model, one way to compute this metric is from the following formula: 1mni=1mj=1n(1 pi > pj), where m denotes the number of positive examples in the data, n denotes the number of negative examples, and 1 is the indicator function that returns 1 when the probability assigned by a model for a positive example is greater than the probability for a negative example.

4.2.3.4 Traditional ML Classifiers

We evaluated several traditional statistical learning techniques on the task of malicious message detection. In particular, we evaluated logistic regression, a linear support vector machine, several variants of the Naive Bayers classifier, a simple decision tree, and a random forest ensemble model. We display results after performing ten-fold cross validation in Figure 11. We evaluate each model along several classification metrics including precision, recall, F1-score, average precision and ROC. While each of these metrics is appropriate for evaluating binary classifiers, they each make different assumptions about model performance and thus give us several views on each classifier, allowing for a more complete understanding of overall performance.

Across all models, we observed particularly high performance in terms of the two probabilistic metrics - average precision and ROC-AUC. Interestingly, for these metrics we also saw very tight distributions of these scores across all folds of cross validation, indicating that for most models these scores are robust to sample-based noise within the data. Across all metrics, we observed fairly high performance and tight distributions over all scores for the random forest classifier. This suggests that this model's ability to ensemble individual models trained on randomly sampled feature sets and training instances makes this model's performance especially robust across all folds of cross validation. In almost all cases, we observed that binary classifiers performed well on this task.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

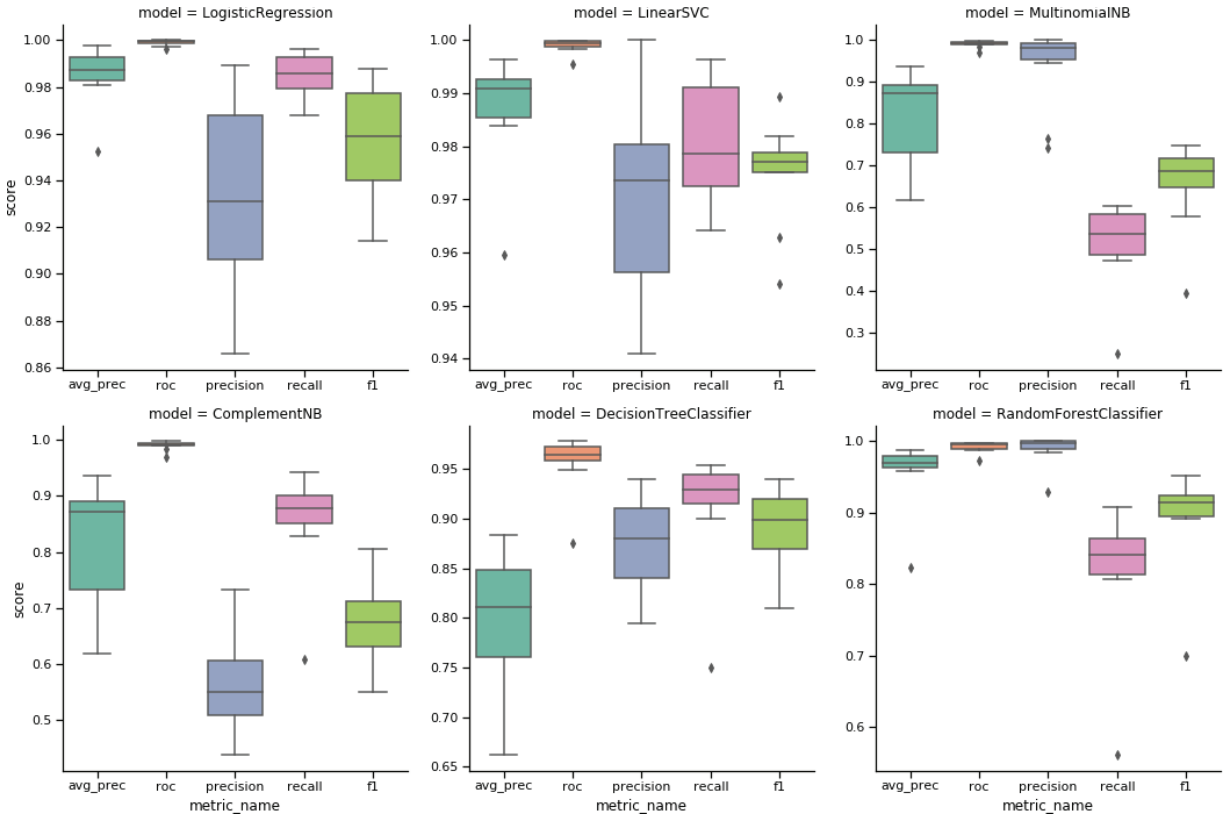


Figure 11. Ten-Fold Cross Validation Results for the Binary Email Classification Task.

4.2.3.5 Recurrent Neural Classifiers

In addition to the models described above, we also trained and evaluated several neural classifiers based on recurrent neural nets (RNN). While nearly all traditional ML models demonstrated strong performance, we were able to achieve superior performance across all metrics using RNNs as shown in the right-most column of Figure 12 below. We also explored training multi-task networks that were trained on both the binary labels (malicious vs. non-malicious) as well as multiclass labels (several types of malicious vs. non-malicious). Previous work has shown that text classifiers can be improved by training on auxiliary objectives, including general NLP tasks such as part-of-speech tagging or language modeling [Yu 2016]. In our case, we used the closely related task of multiclass email classification as an auxiliary objective in training our classifiers. Results for these models are given in the middle column of Figure 12 below, and show that even with extremely strong performance in the single-task regime, we were able to achieve even better results utilizing multi-task training on most metrics.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

	Multi-task Binary Metrics	Single-task Binary Metrics
ROC-AUC	0.999	0.999
Average Precision	**0.993**	0.989
Precision	**0.980**	0.958
Recall	0.980	**0.989**
F1	**0.980**	0.973

Figure 12. Results for Malicious Message Classification Using RNNs Utilizing Both Single-Task and Multi-Task Training (best scores in bold).

4.2.3.6 Speech Act Detection

We also explored initial work in developing classifiers for speech act detection for better understanding of author intent and further downstream conversational processing. Given the promising results in developing neural networks for detecting malicious messages, we focused on comparing different neural architectures for this task. Previous work has also shown strong performance on speech act detection using neural networks [Khanpour 2016]. Additionally, using auxiliary inputs such as speech acts, intention-based latent variables, or persona-based attributes has been shown to be useful in generating more appropriate responses from dialogue models and we took this as further motivation for this branch of research [Wen 2017] [Li 2016].

Results from these experiments are shown in Figure 13 below. We evaluated several RNN-based neural architectures and found minimal differences between the type of recurrent cell chosen (e.g., gated recurrent unit [GRU] or LSTM cells). We compared the performance of a GRU classifier against a convolutional classifier and found consistently superior performance from the CNN. This is somewhat surprising, given that GRUs are able to explicitly model sequential dependencies in text. However, being able to use a CNN in this context was advantageous due to the model's fast training and inference time compared to GRUs or other recurrent networks.

	Precision	Recall	F1-Score
CNN	**0.68**	**0.59**	**0.61**
GRU-NN	0.60	0.51	0.53

Figure 13. Results for Multiclass Speech Act Classification across 17 Speech Acts in the SWDA Corpus (best scores in bold).

For the TA2 task, we focused the majority of our research effort investigating neural network approaches to dialogue modeling. As discussed in section 4.3.2 below, we focused primarily on seq2seq networks capable of encoding an input to a fixed-length vector, and generating a response word-by-word conditioned in part on this encoded vector. Much of this early work relied on the framework OpenNMT and models built using this framework were deployed under the name "polarbot" for the ASED program. These models were all based on LSTM encoders and decoders.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

In addition to these models we explored several techniques for fusing pre-trained language models with seq2seq models in an effort to **(a)** obtain more realistic conversational output from these models and **(b)** hopefully encourage model convergence for faster training times. We detail this work below.

4.2.3.7 Seq2Seq and Language Model Training

Much of the research done in our dialogue modeling work centered around experimenting with fusing pre-trained language models with seq2seq models as described above. In particular we investigated training Transformer language models as in [Radford 2018] in an initial pre-training stage. We trained these models on a standard language modeling objective where the task is to predict a word at a current timestep, given previous words as context as in the following formula:

$$\text{PLM}(x) = t=1|\mathbf{x}|\log\text{PLM}(x_t \mid x<t)$$

Given this trained language model, we then freeze its weights and incorporate the output log-probabilities with the decoder of the seq2seq model during training on the dialogue task. To investigate any potential benefits from this fusion technique, we trained a baseline seq2seq model on the standard dialogue modeling objective:

$$\text{PDM}(y) = t=1|x|\log\text{PLM}(y_t \mid y<t, x)$$

This objective is similar to that used for language modeling, with the addition that the probability of generating a word at a given timestep is conditioned on the encoded representation x as well as previously generated words y<t.

4.2.3.8 Fusion Techniques

Given a pre-trained language model and seq2seq training objective, we then investigated fusing the output of the language model with the output of the seq2seq decoder. In particular we focused on three methods for fusing the outputs of these two models during training on the dialogue modeling objective.

The ***naive sum*** (NS) approach simply takes the element-wise sum of the log-probabilities from the language model and dialogue model prior to computing the loss and updating model weights in the dialogue model only. Final outputs from this training regime are computed according to the following formula:

$$y = \log\text{PDM}(y \mid x) + \log\text{PLM}(y)$$

where logPDM denotes log-probabilities assigned by the seq2seq dialogue model and logPLM denotes log-probabilities assigned by the pre-trained language model. We also investigated a ***weighted sum*** (WS) variant of the naive sum fusion approach in which we introduce a tuning parameter, assigned to a value between 0 and 1, which weights the contribution of the language model. During training of the dialogue model, we allow to be updated along with the dialogue model weights in an attempt to find the optimal value. Finally, we investigate a ***gated product*** (GP) fusion method in which we apply a nonlinear function (the sigmoid) to the language model probabilities and multiply these elementwise with the dialogue model probabilities as in the formula below:

$$y = \text{PDM}(y \mid x)(\text{PLM}(y))$$

where denotes the sigmoid function applied to the language model output probabilities.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

4.2.3.9 Results: Model Fusion

We ran a number of experiments training and evaluating models on the Cornell Movie Dialogue Corpus [Danescu-Niculescu-Mizil 2011] and deployed these models within program pipelines under the name Pacino. An overview of model performance is given in Figure 14 below. We evaluated all dialogue models with two sets of unsupervised metrics - one that primarily measures how well a model's responses overlap with a ground-truth utterance (Bilingual Evaluation Understudy [BLEU], METEOR, ROUGE), and another that attempts to measure semantic overlap between candidate and ground-truth utterances (SkipThought, Embedding Average, Embedding Extrema). Finally, we also evaluate in terms of perplexity, a standard measure in evaluating language models where lower scores denote better fit between the probability distribution learned by the model and the ground-truth distribution over words.

Previous work has cited issues in evaluating dialogue models solely with these overlap metrics [Liu 2016], however we hope to gain a more detailed sense of model performance by considering all metrics in concert. We see from Figure 14 that at least one fusion model is able to outperform the baseline model with respect to every metric. Interestingly, we see little difference in performance between each of the fusion methods. This may suggest that each of these methods converges to similar solutions to the dialogue modeling problem after sufficient training.

	Baseline	Naive Sum	Weighted Sum	Gated Product
Perplexity	50.4	42.4	**41.6**	42.1
BLEU-2	0.011	0.025	**0.031**	0.003
METEOR	0.018	0.032	**0.033**	0.027
ROUGE-L	0.016	**0.031**	0.029	0.018
SkipThought	0.36	**0.60**	0.58	0.59
Emb. Avg	0.80	**0.83**	0.81	**0.83**
Emb. Extrema	0.44	0.50	**0.53**	0.52

Figure 14. Performance Metrics for our Proposed Fusion Methods - NS, WS and GP.

In addition to these summary performance metrics, we plot the loss of each model over the first 100 batch updates during training in Figure 15 to gain a better sense of the learning dynamics of each training regime early in the training process. In this very early training stage, we see that each fusion technique appears to dampen severe jumps in loss compared to training the baseline model. Though it does not achieve the best perplexity score on the test set, we also see the GP training method appears to have the greatest effect in smoothing out these noisy updates during training, which may indicate an advantage in terms of training time over the other methods. Overall, these curves suggest a strong regularizing effect due to incorporating the language model in addition to the much faster convergence to a lower loss.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

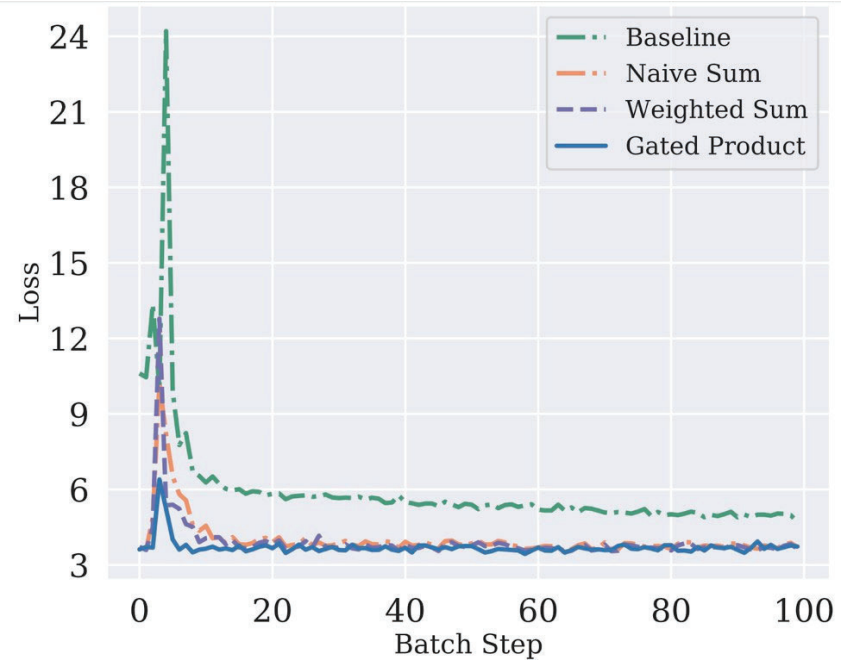


Figure 15. Loss Convergence for Different Fused Dialogue Models Tested in our Conversational Modeling Experiments.

4.2.3.10 Dialogue Model Decoding Methods

In addition to the experiments with language model fusion outlined above, we also focused part of our efforts on evaluating different decoding methods for our models. While many of the assumptions from neural machine translation (NMT) have been carried over to neural dialogue modeling, there are important differences between the two problem spaces. NMT in general can rely on training regimes that guide a model toward output that resembles a single ground-truth example sentence. That is, there is typically a one-to-one relationship between an input sentence and its translation (e.g., there is one or very few acceptable translations). In dialogue modeling this assumption does not hold. For a given input sentence, there may be several equally appropriate responses, but these systems are trained to generate outputs that resemble a single ground-truth output as in the NMT training regime.

This issue in dialogue model training often results in models that output generic or short responses such as "I don't know", regardless of the input. To combat this, previous work has investigated alternative decoding techniques to encourage greater diversity in model responses. **Beam search** attempts to address the issue of maximizing the probability over an entire generated sequence by avoiding greedily selecting the word with highest probability at each timestep. This technique constructs n hypotheses (beams) that consist of partially constructed sequences and their probabilities, and the final returned output consists of the sequence of words with the maximum probability over the entire sequence among the n beams. **Top-*k*** sampling encourages diversity in the model's output by ranking all words in the vocabulary by their assigned probability and sampling a word to be generated from the top-*k* words in that ranking. This technique avoids generic responses by introducing randomness at each decoding step while restricting the candidate words to only those that are highly probable.

Examples from the Pacino system using these three decoding techniques are given in Figure 16

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

below. Despite each row in the table corresponding to different input given to the model we see that greedy decoding outputs identical responses regardless of input, demonstrating the need to investigate these additional techniques. In contrast, we observe shorter but slightly more diverse output from the beam search method and the most diverse output from top-*k* sampling. Most importantly, we note that the output from the top-*k* sampling technique does not sacrifice readability or grammaticality in its outputs while yielding more diverse and interesting output.

Greedy Decoding	Beam Search	Top-*k* Sampling
i'm not sure.	yeah.	i'm sorry. i don't know.
i'm not sure.	i don't know.	oh, yes?
i'm not sure.	no.	i want you to do it.

Figure 16. Examples from our Systems Showing Generic Phrases from Greedy Decoding (left) and Increased Diversity in Responses Utilizing Beam Search (middle) and Top-K Sampling (right) Approaches.

4.2.4 Intent

Intent classification was an area of NLP-focused research for TA1. Knowing whether or not a message is malicious or benign often involves identifying what the sender's end goal is. Knowing the intent of an email can be used not only to classify a message as malicious or not, but also acts as an indicator of how the sender should be engaged in future steps. Additionally, not every malicious email immediately contains requests for information. Sophisticated attacks often occur over a longer period of time with multiple messages exchanged between the sender and receiver. We focused on the following classes of intent:

- Acquire credentials
- Acquire personally identifiable information
- Build trust
- Elicit fear
- Gain access to social network
- Gather general information
- Get money
- Install malware

For each intent, a binary classification model was trained and aggregated into a single multi-class classifier. Various ML models were evaluated with focus on more advanced models including BERT [Devlin 2018] and attention-based models.

4.2.4.1 Data

Switchboard Dialog Act Corpus [Jurafsky 1997]. Initial models were trained on this dataset as a preliminary step to get an idea of how various models performed on a similar-NLP tasks.

Twitter Customer Support Dataset (Kaggle) [Thought Vector 2017]. Customer support

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

conversations often contain multiple exchanges of requests for information. The dataset did not provide the labels required for our task but the types of messages in terms of formatting, language, and tone, would be similar to the types of messages we expect to receive in a social engineering attack scenario.

The Enron Email Dataset (Kaggle) [Cukierski 2016]. It was important to find a dataset with a variety of messages written in an email context. As with the Twitter dataset, this dataset did not have the types of labels we needed. However, the context of this data was useful in reproducing real-world scenarios for our models to train on.

Campaign Emails. This labeled dataset was provided by members of the program. It contained a collection of emails with classifications for certain abuse categories such as spam, malware, credential phishing, and social engineering**.**

4.2.4.2 Annotations

A weak supervision [Ratner 2017] pipeline was developed to generate "noisy" labels for unlabeled data. Multiple user-defined labeling functions were aggregated to roughly estimate labels to train a generative model. These labeling functions varied in methodology and consisted of a range of keyword classifiers, regular expressions matchers, and rule-based classifiers using parts of speech tagging, named entity recognition, and other linguistic features. The generative model was used to produce labels that were then used to train a generalizable discriminative model (i.e., BERT and Hierachical Attention Networks [HAN]). By applying weak supervision to the datasets described above, approximately 2 million noisy labels were generated for training. Additionally, a small sample of approximately 3,000 texts in the data were labeled by hand in order to evaluate the accuracy of the models after training on noisy data.

4.2.4.3 Models

DistilBERT - A transformer model based off of BERT (originally released by Google). DistilBERT is lighter and faster with similar classification performance to BERT. Many other "lite" versions of BERT were released shortly after DistilBERT. Additional performance gains may be found using those.

HAN - A hierarchical attention model implementation. This model analyzes the hierarchical structure of a document by focusing on identifying high-value sentences in the document, and high-value words in those sentences.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

4.2.4.4 Evaluation

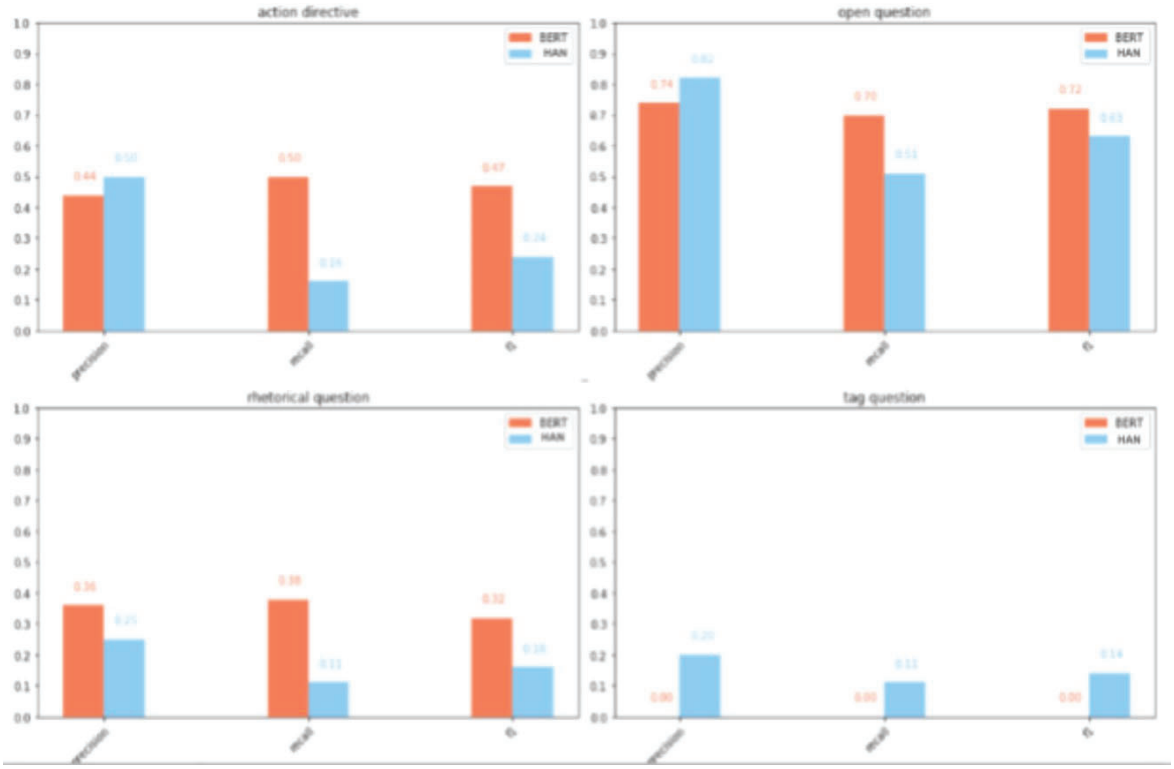


Figure 17. Comparison of BERT and HAN Models on Relevant Labels in the Speech Dialog Act.

Figure 17 shows the evaluation of BERT and the HAN on the Switchboard Dialogue Act Corpus for 4 out of the 44 tags. This test was performed in order to get a broad idea of each model's ability to understand the context and intent behind language. Though this dataset was used to evaluate a similar type of task, the data was in the context of spoken utterances and the tags were dialog acts which did not align well with the specific problem we were trying to solve. Therefore, this data was only used to evaluate the general NLP capabilities of each model.

4.2.4.5 Results

On noisy labels, both models performed very well in determining whether or not texts had the intent of acquiring information (i.e. acquire credentials, acquire personally identifiable information, gather general information, or get money classifications). Each model was trained on the set of 2 million noisy labels and tested on a hand-labeled mixture of emails and texts from the Twitter and Enron datasets. Accuracy for the HAN and BERT peaked at about 90% and 88%, respectively. It is important to note that because benign messages are much more common than attack messages, the test set was unbalanced with significantly more benign messages than attack messages

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

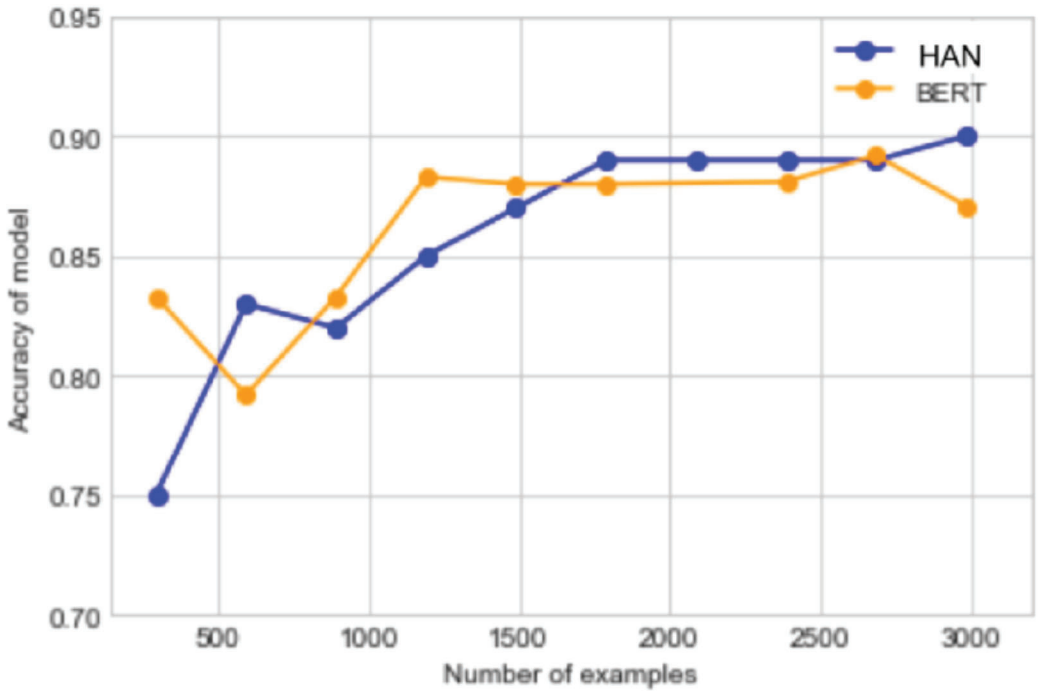


Figure 18. Training Curves of Final Models for Evaluated on a Mix of Noisy and Hand-Labeled Data.

Though the HAN performed marginally better in terms of accuracy, it ran slower than DistilBERT. Therefore, DistilBERT was the more practical solution and was chosen as the model to use for the deployed classifiers. Additional evaluations were run on the campaign emails using this model. Labels for the campaign emails did not align directly with the classes the model was trained for so a mapping of labels (Figure 19) was made.

Model Classification Label	Campaign Classification Label
Acquire Credentials	Cred Phishing
Install Malware	Malware
Gather General Info	Recon Spam
Access Social Network	Phish Training Propaganda Social Engineering

Figure 19. Mapping of Model Classification Labels to Labels in Campaign Email Dataset.

Figure 20 below shows the results of the classifier on the campaign emails. Once again, as the data was imbalanced, having an accuracy of 0.50 is likely better than random guessing. More work is required to determine the extent of the imbalance and how the models perform on a balanced dataset.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

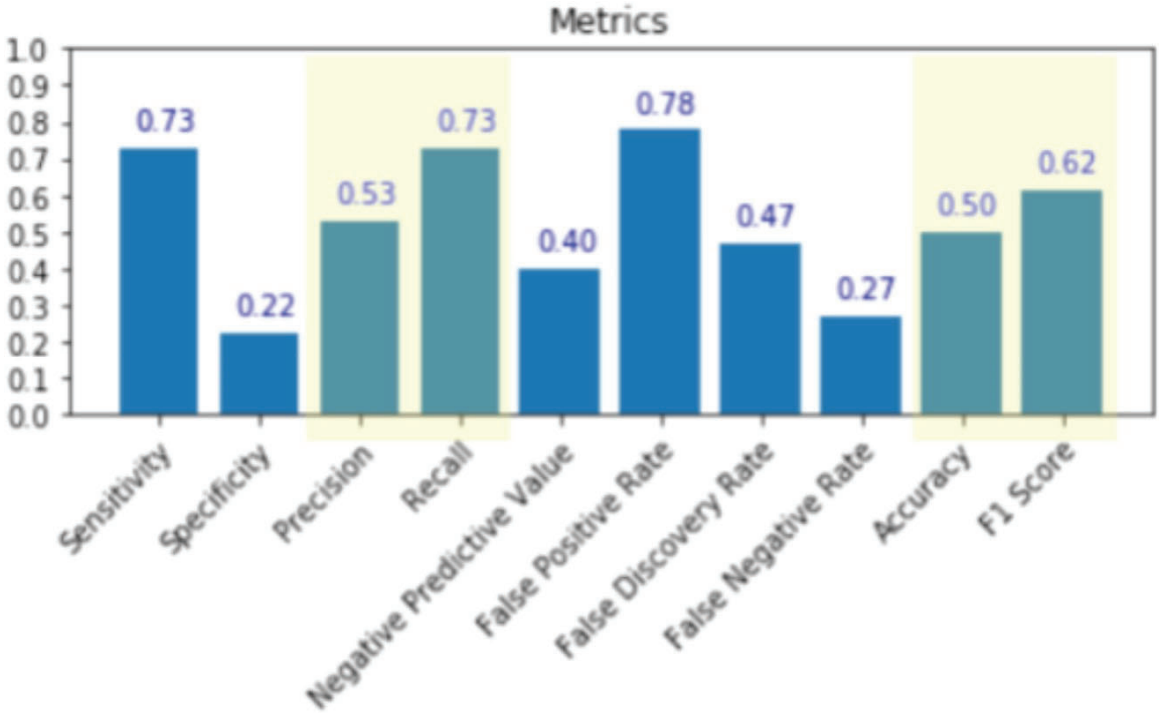


Figure 20. DistilBERT (trained on noisy data) Evaluated on Campaign Emails.

The results showed that an F1-score of 0.62 and an accuracy of 0.50 was achieved. This is significantly lower than the tests on the previous evaluation on noisy labels. The lower performance is likely attributed to a combination of differences in context and formatting of the emails as well as the misalignment of data labels.

Overall, the tests showed promising results in the application of NLP models and weak supervision on intent classification. Continued work is recommended to improve the generalizability and accuracy of the models. Future work should include focus on generating more accurate labels, as well as curating a dataset with my variation in the types and contexts of messages.

4.2.5 SpamAssassin

Initially, Recourse used an off-the-shelf open source spam classifier, SpamAssassin [Apache SpamAssassin 2018], as one of its main TA1 components. SpamAssassin performs spam detection of email messages using Bayesian classification, fuzzy checksums, and online domain blacklists of known spammers.

Initial baseline results for SpamAssassin showed promising results on ‘generic’ spam messages (e.g., the ‘Nigerian Prince’ dataset [Tatman 2017]). However, targeted spear-phishing attacks, such as those used during TA3 testing, were often missed by SpamAssassin.

In the end, SpamAssassin was deemed useful due to low false positive rate: if it classified an incoming message as “spam,” ReCourse could be somewhat confident that the message was malicious.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Dataset	Number of emails	Classification Accuracy
Nigerian Prince (SPAM)	2805	0.845
Enron emails (assumed HAM)	502,005	0.996

Figure 21. SpamAssassin Classification Accuracy

4.2.6 Thug Honeyclient

Thug [Dell'Area 2020] is an open source browser emulation 'honeypot' that follows URLs extracted from messages to detect malicious content.

Baseline results for Thug showed good performance during manual testing with known malware exploits (i.e., with known Common Vulnerabilities and Exposures [CVE] signatures [MITRE 2020]), but Thug missed many of the more targeted TA3 attacks. Some of these missed attacks could be due to the synthetic nature of the TA3 attacks -- it was not possible to use real malware payloads during live ASED testing due to the risk of infecting volunteers' computers.

To improve performance on the kinds of spear-phishing attacks employed by TA3, several heuristics-based enhancements were made to Thug, including:

- Hypertext Markup Language (HTML)-based Intent Detection -- Analysis of HTML form fields in URL links to detect if an HTML page is asking the user to enter user info/credentials (e.g., spoofed login page).
- Hypertext Transfer Protocol (HTTP)-redirection analysis -- Number of redirections, cyclical/suspicious redirection behaviour
- Payload Multipurpose Internet Mail Extensions (MIME)-type obfuscation detection (e.g., EXE file with Joint Photographic Experts Group (JPEG) file extension)
- Additional payload malware analysis using VirusTotal Representational State Transfer (REST) API
- Analysis of email attachments as well as Uniform Resource Locator (URL) links

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

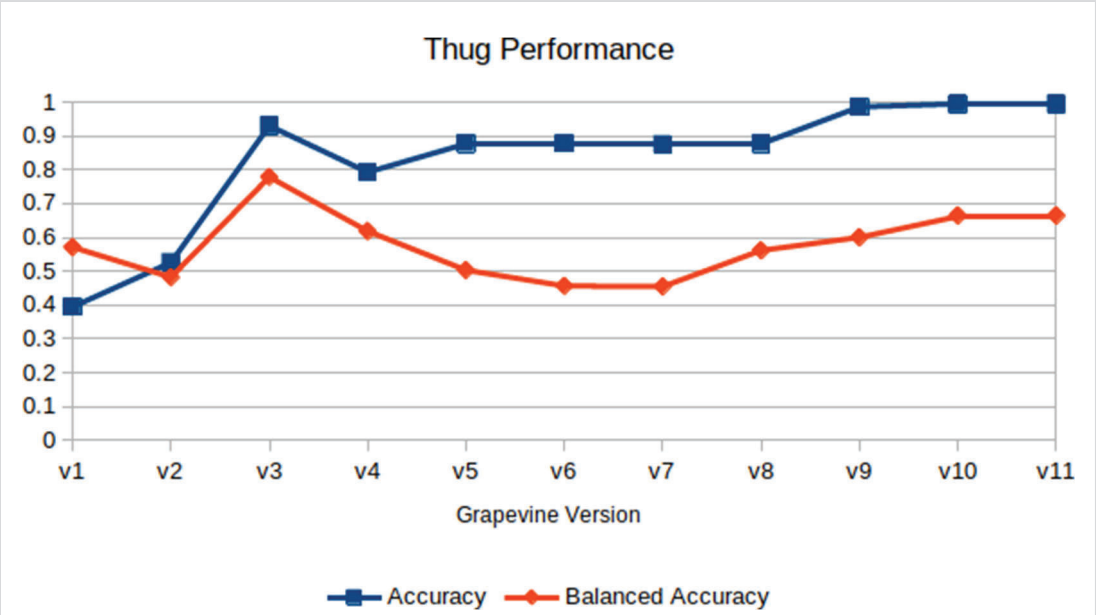


Figure 22. Accuracy of Thug Honeypot.

4.2.7 Rolodex

Rolodex uses ReCourse's knowledge base to determine a 'trust score' for a given account that evolves over time. Rolodex used multiple sources of data to calculate an account's aggregate trust score, including:

- Total number of FRIEND vs FOE messages received for a given account
- Is account internal or external to the organization
- 'Whois' information for an account's domain
- Is account easily 'spoofable'? (e.g., freemail accounts such as Gmail, Yahoo, Hotmail)

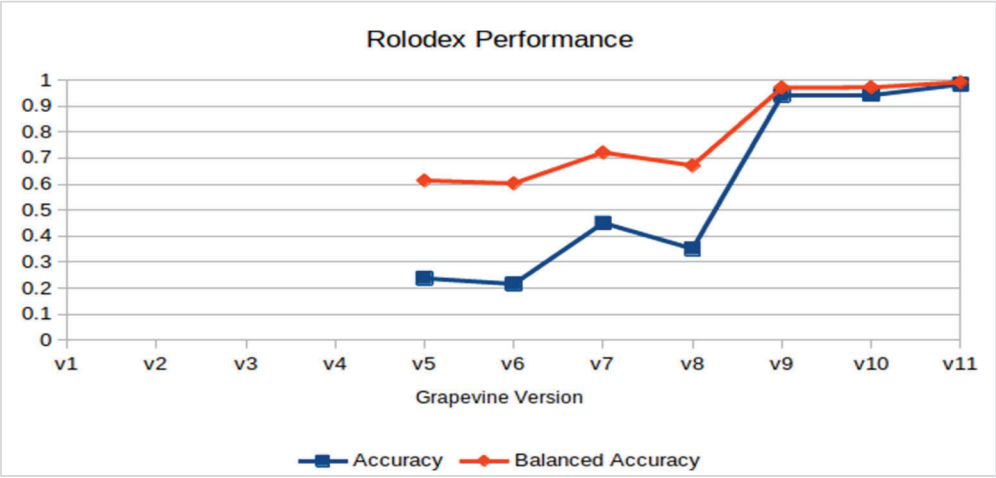


Figure 23. Accuracy of Rolodex

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

4.2.8 Pigeonhole

Pigeonhole is a cross-account stream-based analytic that groups similar messages together based on time and content. Based on the *CopyCatch* method [Beutel 2013].

For example, if different unique messages having similar content were received by different recipients within a given time window, they are grouped together. If the messages are deemed to have malicious content then they are classified as being part of the same malicious 'campaign,' possibly originating from the same Bad Actor.

In this context, Pigeonhole is not simply looking for emails copied to multiple people, but rather malicious messages having a similar 'signature' sent to several people.

Heuristics such as text similarity, URL links and attachments, were used to determine overall email message similarity.

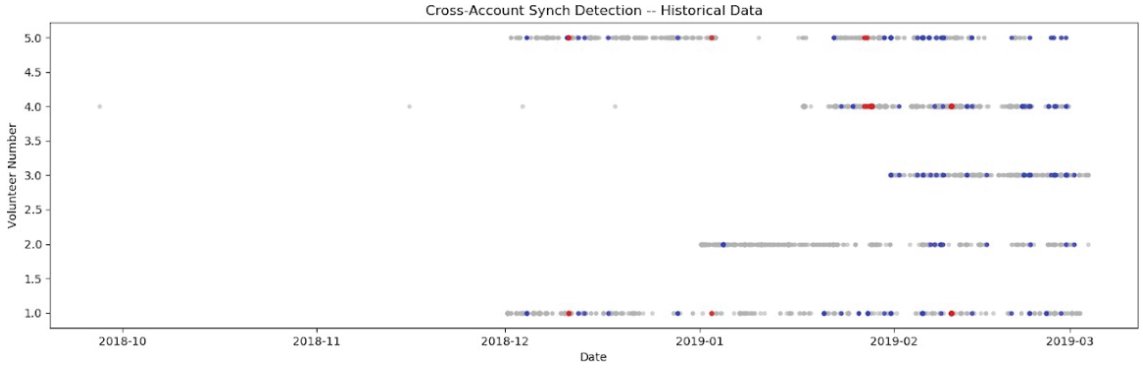


Figure 24. Pigeonhole Results on the 'Historical' Email Dataset, Containing mostly FRIEND (non-malicious) Emails Received by Five ASED Volunteers.
Red dots indicate incoming messages with suspicious time/content signatures.

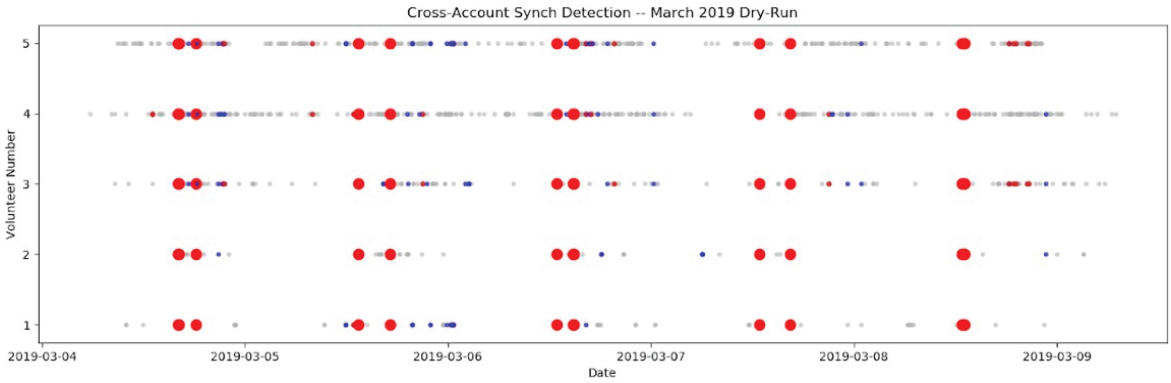


Figure 25. Pigeonhole Results on the 'March Dry Run' Email Dataset, Containing both FRIEND and FOE Emails.
More malicious cross-account campaigns were detected, as expected. The size of the dot represents the number of messages per 'campaign' group.

4.2.9 Dyslexia

Dyslexia is a TA1 classifier that leverages ReCourse's knowledge base to check for accounts or

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

domains that are intended to look legitimate at a glance (e.g., *grnail.com* vs *gmail.com)*. This attack technique can be quite successful, especially in case of impersonation attacks. Dyslexia flags accounts or domains that are above a tuneable similarity threshold, where the similarity is provided by PostgreSQL's pg_trgm extension [PostgreSQL 2020]. Dyslexia was included in the TA1 pipeline after the March dry-run evaluation.

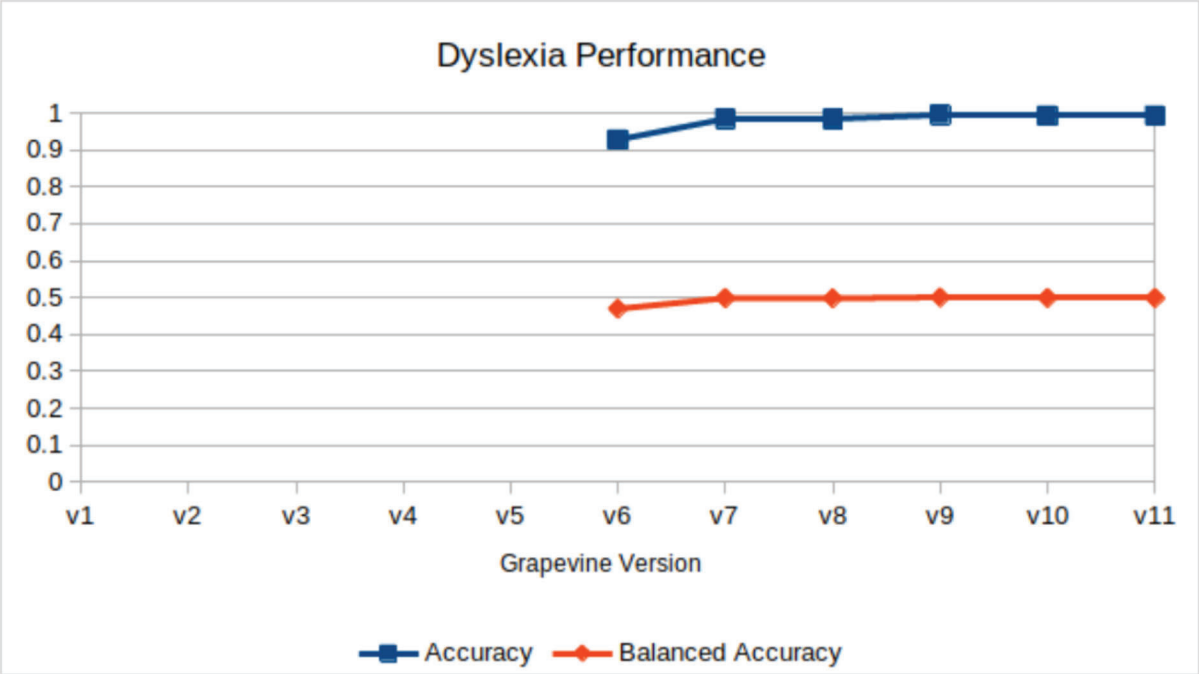


Figure 26. Accuracy of Dyslexia TA1 Classifer.

4.2.10 Whois

The Whois TA1 classifier targets a strong signal present in many attacks: publicly available registration information about domains that appear in the message or in the account of the attacker. Whois uses a third-party service to perform a whois lookup, which reveals such data points as who registered the domain, their contact information and when the domain was created (malicious domains are often recently created and short-lived). It also leverages ReCourse's knowledge base to identify previously analyzed domains, whitelisted domains, etc. Whois was included in the TA1 pipeline after the March dry-run evaluation.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

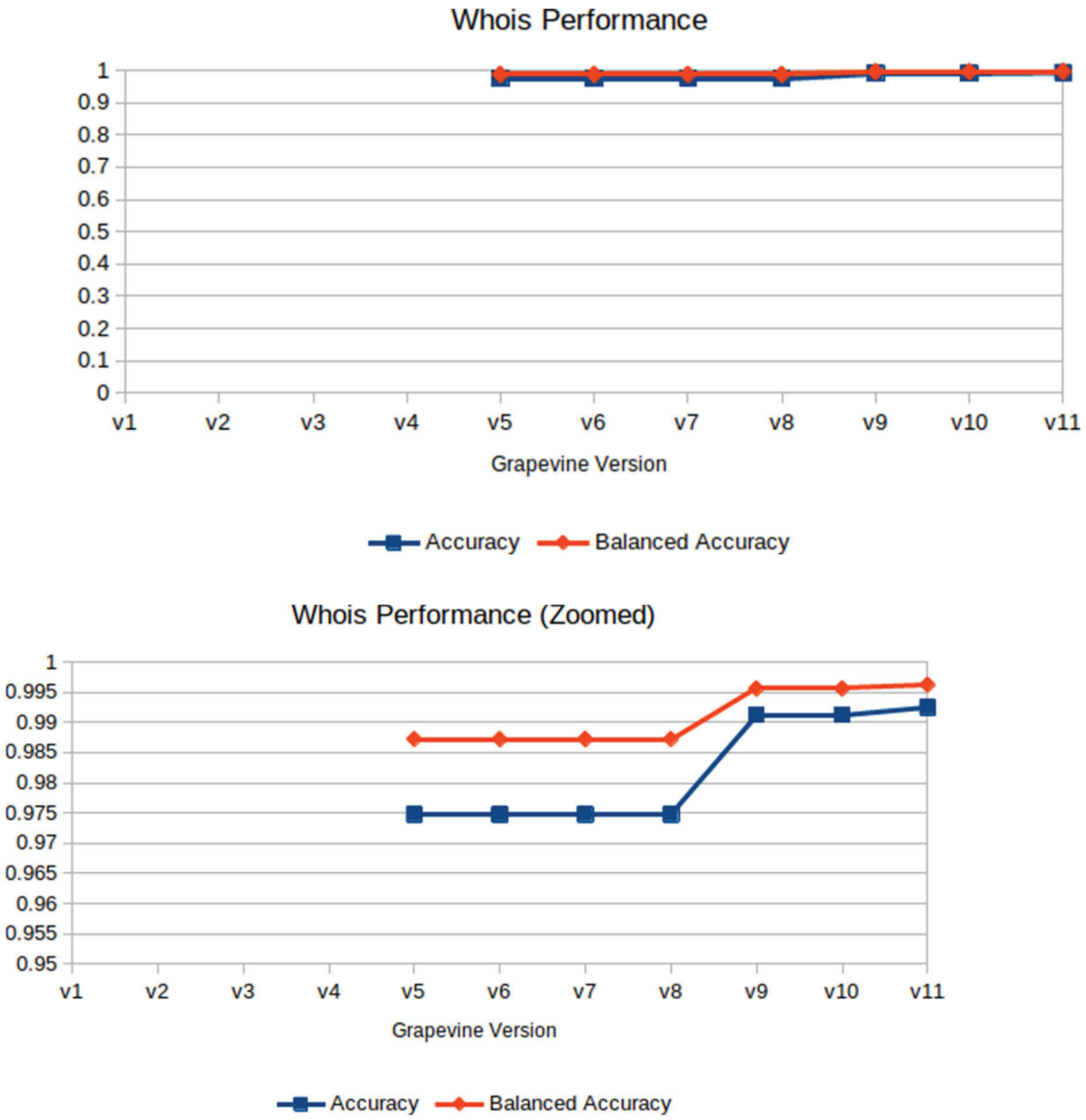


Figure 27. Accuracy of Whois TA1 Classifier.

4.2.11 Additional Methods

4.2.11.1 Shapelet Classifier

This classifier learns a dictionary of "shapelets," i.e., discriminative subsequences, from time series in the training data (the subsequences are not unique). The closest distance between each shapelet and a time series defines a new feature representation, known as the shapelet- transformation. The model is adapted from [Grabocka 2014] and draws heavily from tslearn's open source shapelet library.

4.2.11.2 LSTM-FCN Classifier

This classifier combines the feature representations learned from three neural network stacks to produce a classification of a time series. Specifically, the three stacks are: 1)

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

embeddings of timefeatures (month, day of the month, day of the week, hour of day, minute of an hour, and second of a minute) followed by a linear layer, 2) an attention-based LSTM layer followed by dropout, and 3) three sequences of a 1D convolution followed by batch normalization. The model is adapted from [Karim 2019].

4.2.11.3 Information Diffusion Methods

This approach tracks the diffusion of pieces of content (e.g., links or n-grams) over time through a network of online accounts and uses this signal to measure influence and coordination among the accounts. The primary tool explored in this area is the Multivariate Hawkes Process [Chen 2017], which models events in continuous time. The model, as applied to this domain, is based on the assumption that if one account posts a piece of content, the other accounts that are heavily influenced by the source account are more likely to post that content in the near future. Using an optimization-based approach, we were able to estimate the influence network of a large number of accounts. As this approach takes both time and content into account, the results could be used to identify threats and detect coordination that is not well-revealed by other approaches, complementing the time- or content-only approaches present in the program.

4.3. TA2

For TA2, the ReCourse system could run in one of three modes:

- Fully autonomous
- Fully HITL
- A mixture of autonomous and HITL

In the first scenario, all message threads are assigned to a bot so that all responses in the thread come from the same bot technology. If a particular bot fails, another bot may be selected at random to take over the conversation. Depending on the bot technology assigned, a Fingerprint link may be included in some of the responses. This was the mode used for the full component of the evaluation.

In the second scenario, no messages receive automatic replies; instead, all replies must be made by a human. The ReCourse User Interface (UI) provides suggested responses from all of the bot technologies, and the operator can choose to use one of those responses or craft their own response. In either case, they can opt to include a Fingerprint link.

In the third scenario, some of the accounts being monitored by ReCourse can be put in autonomous mode while the remaining accounts are handled by the operator. This was the mode used for the dialogue portion of the program evaluation.

When Fingerprint links are included in responses, Grapevine monitors for clicks and when one is recorded, it reports the results to STIX. Those results include a variety of data points that can be helpful in identifying the attacker.

4.3.1. Natural Language Style Transfer

Style transfer of natural language text is an important capability for the ASED program as it allows the decoupling of task-based dialogue generation from the generation of text that seems appropriate to the attributed persona. This permits multiple dialogue systems to generate candidate responses that feed into a single style-transfer model, which converts the text to the target style. This also encourages the development of a wide range of dialogue generation

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

techniques, which we saw in the program. In pursuit of this goal, we explored two approaches: 1) Marionette, a sequence-to-sequence model using parallel corpora, and 2) Ventriloquist, a denoising auto-encoder model that avoids the need for parallel training data.

4.3.1.1 Marionette

This approach treats style transfer as a particular case of translation and aims to apply the state-of-the-art tools in neural machine translation to this problem. The most direct application of this idea requires a parallel corpus consisting of pairs of text that have the same meaning but are expressed in style A and style B respectively.

Marionette implements a sequence-to-sequence transformer architecture similar to that in [Vaswani 2017], which takes text in style A and returns text in style B while preserving the underlying meaning of the text. We explored approaches for creating such parallel training datasets using techniques from paraphrase detection [Brockett 2005; Dolan 2005]. Pre-trained language models are fused with the output of the sequence-to-sequence model providing a basic knowledge of English upon which the learned model imposes its style-specific features.

While this approach proved successful when sufficient training data was available, creating the necessary datasets for a large number of styles in a variety of domains proved to be a difficult research task. Given the lack of open source text style transfer datasets, we were able to produce paraphrases, but the diversity and control over style was lacking. For these reasons, we favored the auto-encoder approach implemented by Ventriloquist which is able to take advantage of a much wider range of training data.

Simple Paraphrasing	
what is required	what is necessary
its obligation to	its commitment to

Formal ⟷ Informal	
i would appreciate it if you please wait until i read that questionnaire	i would like it if you wait until i read that questionnaire

Figure 28. Style Transfer Examples from Marionette.

4.3.1.2 Ventriloquist

This approach implements a denoising auto-encoder that utilizes a transformer architecture with relative position attention in both the encoder and decoder modules. The output of the decoder module is also optionally fused with a pre-trained GPT-2 language model for regularization

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

[Radford 2019].

Ventriloquist was inspired by and bootstrapped from extensive previous work on natural language style transfer, including retrieval approaches [Li 2018], fusion techniques [Sriram 2017; Stahlberg 2018], back-translation [Lample 2018], adversarial training [Hu 2017], and adversarial alignment of hidden states [Shen 2017].

The codebase supports multiple input noising strategies (MASS pre-training [Song 2019], word attribute selection and n-gram attribute selection), multiple encoder and decoder architectures (transformers, LSTMs and mixed architectures), multiple training paradigms (fused language model, back-translation and an adversarial loop), multiple loss functions (cross-entropy and a differentiable lower bound on the expected BLEU score), and multiple decoding approaches (greedy and top-k).

The objective function for the basic Ventriloquist model was simply a reconstruction loss, and therefore, the amount of training data we could use for each style was bounded only by computational considerations. The examples in Figure 29 begin to demonstrate this increased scalability, as we translate to four styles instead of two. Additionally, although this screenshot was taken at an early iteration of model development, it already demonstrates increased dexterity in generating interesting stylized responses. The model's training procedure directly encourages this agility, as it learns a unique embedding for each style on which the model was trained. To produce a stylized response, the model combines a unique style embedding with a shared content embedding (learned from the reconstruction of all the styles), and thus maintains fidelity to the original content while injecting more style-specific semantics.

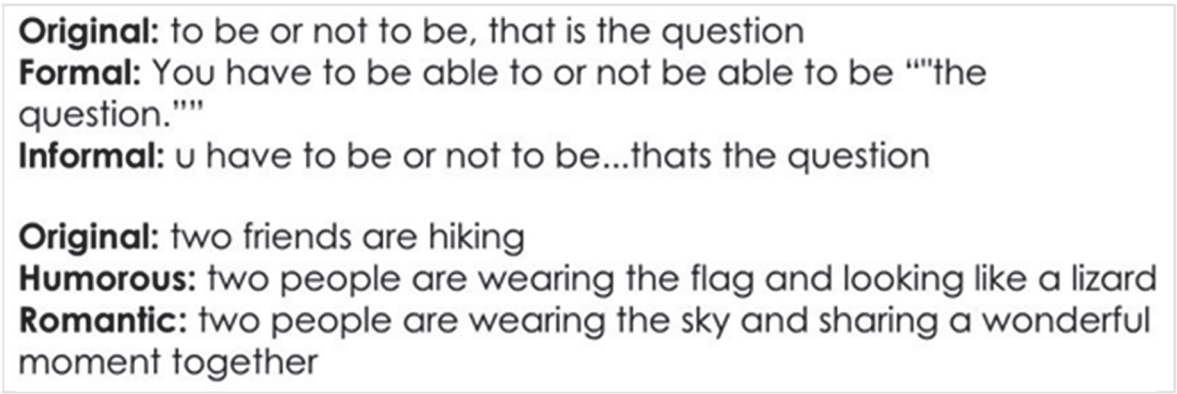


Figure 29. Style Transfer Examples from Ventriloquist.

4.3.2. Automated Dialogue Modeling

For the TA2 portion of the ASED project, we experimented with several dialogue modeling techniques to develop systems capable of automatically engaging in conversation with users perceived as sending malicious messages. The primary aim of this branch of research was to develop fully automated chat systems capable of not only engaging in realistic conversation with a human participant, but also requesting or eliciting useful information from this participant. Due to the high complexity of this end-goal, simplistic templated or rule-based systems were too restrictive to produce the wide range of possible conversational responses needed. We therefore leveraged several techniques stemming from work in deep-learning-based dialogue modeling. In particular, we focused on several applications of sequence-to-sequence neural network

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

architectures.

4.3.2.1 Seq2seq Models

Though seq2seq models are a relatively new modeling technique compared to more traditional statistical learning approaches, there has been a large surge in the application of these neural networks to sequence-modeling tasks. Originally developed for NMT, where the goal is to automatically generate a fluent translation from a source language into a target language (e.g., from French to English), these networks were quickly adapted for other linguistic tasks including dialogue or conversational modeling [Sutskever 2014].

Work in NMT has typically utilized two neural networks trained jointly on the sequence modeling task - an encoder that learns to compress a sequence of words into a fixed-length vector representation, and a decoder that is conditioned on the output of the encoder and generates a response word by word. Almost all initial work in developing these models has used recurrent neural networks for both the encoder and decoder of these networks, due to the ability of RNNs to effectively model sequential dependencies. However, alternatives have been proposed utilizing either convolutions [Gehring 2017] or purely attention-based mechanisms [Vaswani 2017] for both the encoder and decoder networks. While these architectural advances were initially explored for NMT, they have subsequently been applied to other tasks including dialogue modeling.

4.3.2.2 Fused Seq2seq Models

While many variants of seq2seq models have been successfully applied to problems in NMT and dialogue, there has also been some work for fusing these models with pre-trained language models to reduce training time and encourage convergence [Sriram 2017] [Stahlberg 2018].

These techniques typically combine output log-probabilities from the pre-trained language model with those of the decoder in the seq2seq model, allowing the language model to act as a "guide" either during seq2seq training, inference, or both.

In our dialogue work, following [Stahlberg 2018] we explored fusing pre-trained language models with seq2seq models to stabilize training, and obtain more realistic output from our dialogue models. In our experiments we evaluated several fusion techniques to find the most effective way for our dialogue models to leverage information from the pre-trained language models without over-relying on the language model. We also provided an analysis of trends in the weight updates in the decoder of the seq2seq model, showing that these weights were still effectively updated or "learned" despite incorporating input from the language model. These results were reported in a paper submitted to an NLP conference, and a slightly abridged version of these results is included in Section 4.2.

4.3.3. Fingerprint

Fingerprint is a stand-alone application developed by Uncharted to aid in the identification of attackers. It provides an API through which an operator using ReCourse can upload images or PDF files and then send links to those assets to attackers. Links are associated with a specific attacker to allow tracking over time.

The Fingerprint application presents as a file sharing site to unsuspecting attackers. However, when they follow a link to an asset, a number of potentially identifying data points are collected, including browser fingerprint, Internet Protocol (IP) address (and thereby geographic location),

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

user agent, etc.

Grapevine then uses the Fingerprint API to report obtained data to the STIX TA2 endpoint.

Browser fingerprinting can aid in the identification of attackers who are operating on multiple channels, launching multiple attacks or targeting multiple victims. In the October evaluation, this technique allowed the ReCourse team to identify that the same browser was involved in at least two separate attacks.

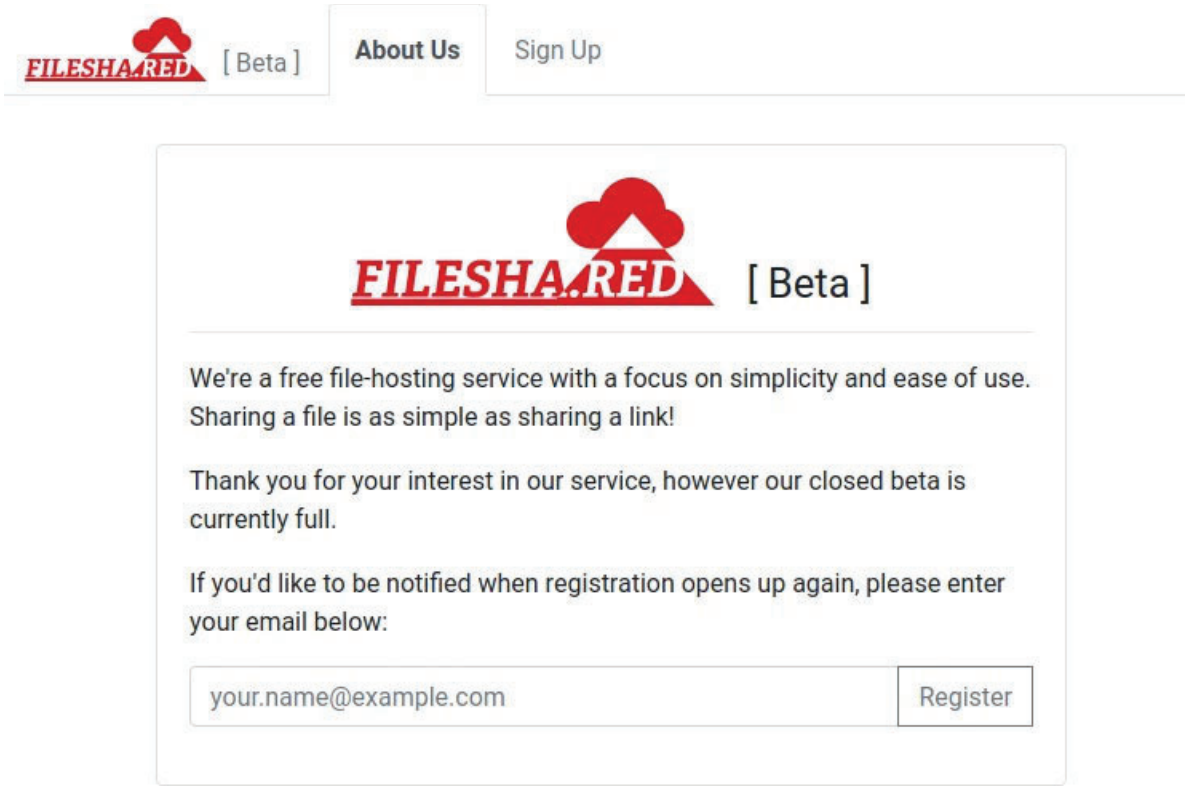


Figure 30. Fingerprint Site as it Appears to an Unassuming Attacker.

Figure 31. Fingerprint Results Surfaced in ReCourse.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Data Point	Value
Fingerprint Hash	a12d52c15270d63fa2e62566aad30c7e
User Agent	Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36
IP Address	197.242.112.107
Latitude	6.4531
Longitude	3.3958
City	Lagos
Region	Lagos
Country	Nigeria
ISP	Spectranet Limited
Language	en-US
Time Offset	-60
Timezone	Africa/Lagos
Hardware Concurrency	2
CPU Class	not available
Device Memory	not available
Screen Resolution	768x1366
Available Resolution	768x1366
Color Depth	24
Session Storage	true
Local Storage	true
Indexed DB	true
Add Behavior	false
Open Database	true
Platform	Win32

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Fonts	Arial, Arial Black, Arial Narrow, Arial Unicode MS, etc.
Plugins	Chrome PDF Plugin, Chrome PDF Viewer, Native Client, etc.
Canvas Winding	yes
WebGL Vendor and Renderer	Google Inc.~ANGLE (AMD Radeon HD 6310 Direct3D11 vs_5_0 ps_5_0)
WebGL Extensions	ANGLE_instanced_arrays, EXT_blend_minmax, etc.
Touch Support	false
Ad Block	false
Has Lied Languages	false
Has Lied Resolution	false
Has Lied OS	false
Has Lied Browser	false
Audio	124.0434474653739

Figure 32. Example Data Points Obtained from a Fingerprint.

4.3.4 Gabby

Gabby is a chatbot developed by Uncharted. It is based on the Transformer network [Vaswani 2017], and was trained on a subset of dialog turns from the PolyAI Reddit dataset [PolyAI-LDN 2019]. The Transformer uses a simplified encoder/decoder architecture to encode "attention" (i.e., linguistic context of a word or phrase) as well as to boost training speed compared to previous generation NLP Neural Networks. Gabby's dialog system then uses the Beamsearch [Wikipedia contributors 2020] algorithm to convert decoded probabilities into the chatbot's speech output.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

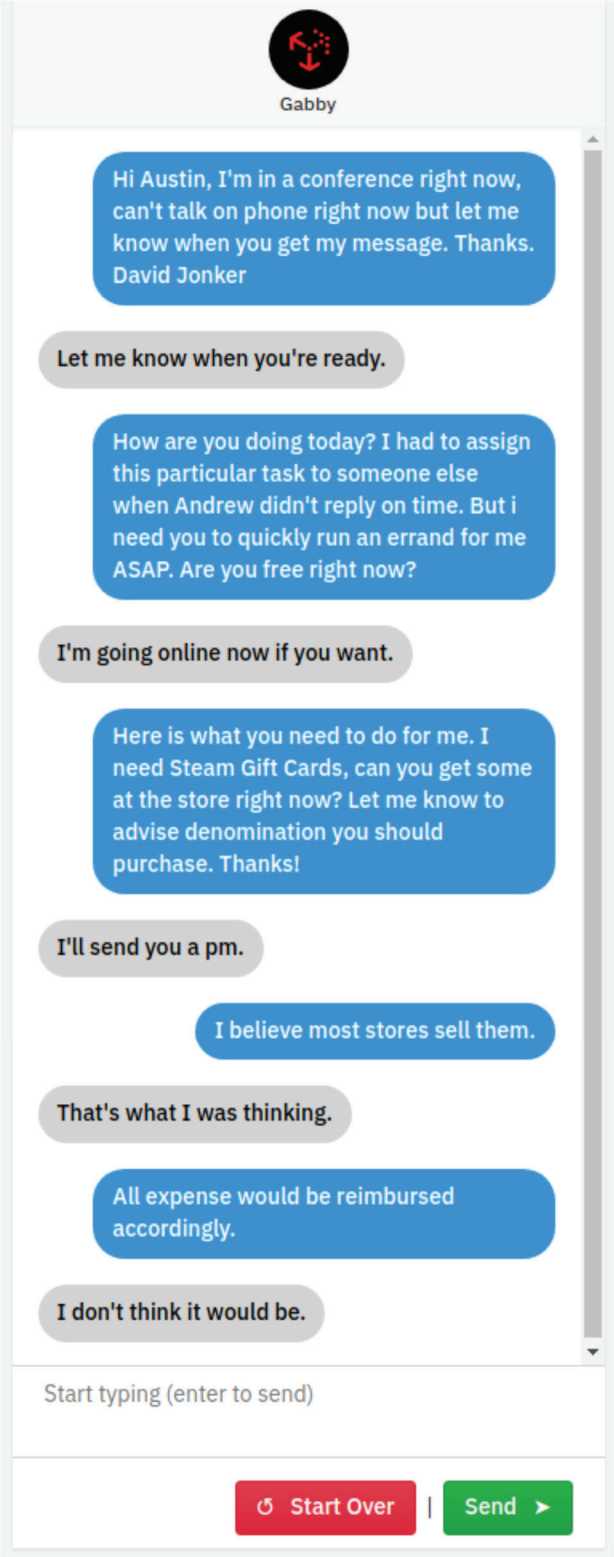


Figure 33. Sample Conversation with Gabby.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

4.3.5 Fezzik

Several enhancements were made to the Gabby architecture and the result was a new chatbot called Fezzik. The main improvements were as follows:

- **Use of OpenAI's Generative Pre-Trained (GPT) Transformer language models as a pre-trained base**. This allowed for rapid fine-tuning of new Fezzik chatbots via transfer learning as opposed to re-training Gabby's "vanilla" Transformer model from scratch each time [Huggingface 2020].
- **Conversational History.** Gabby had no "memory" of previous turns in a conversation. Fezzik's model can attend to the previous X dialog turns when formulating its response (X = 2 by default).
- **Augmented Training Data**. Additional training data was used that included samples of multi-turn dialog (needed to properly train the model to use conversational history): PolyAI Reddit [PolyAI-LDN 2019], DailyDialog [Li 2017] and PersonaChat [Facebook Research 2019] datasets.
- **Limitless Vocabulary**. Chatbots trained on whole word tokens respond with an "UNK" error if they see an unknown word. Fezzik solves this issue by using GPT's byte-pair tokenization library instead of whole word tokens.
- **Varied Responses**. Fezzik's decoder uses Nucleus Sampling instead of Beamsearch for a more diverse vocabulary. The table below shows that Fezzik has a distinct-2 score -- a measure of vocabulary diversity -- over 30x higher than Gabby.

	Ground Truth Responses	Gabby	Fezzik
Avg Words / Msg	13.6	5.7	11.4
Distinct-2 Score	0.320	0.004	0.150

Figure 34. Comparison of Gabby and Fezik against Ground Truth Responses.

4.3.6 Integration with Strategies for Investigating and Eliciting Information from Nuanced Attackers (SIENNA) (Bolt Beranek and Newman Inc. [BBN] Technologies)

SIENNA is a partial TA2 solution from BBN Technologies. ReCourse integrates with SIENNA by providing its suggested replies in the UI for the HITL scenario and by using its recommendations as one of the options in the fully automated approach. Both modes of operation were successfully used in the October evaluation.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

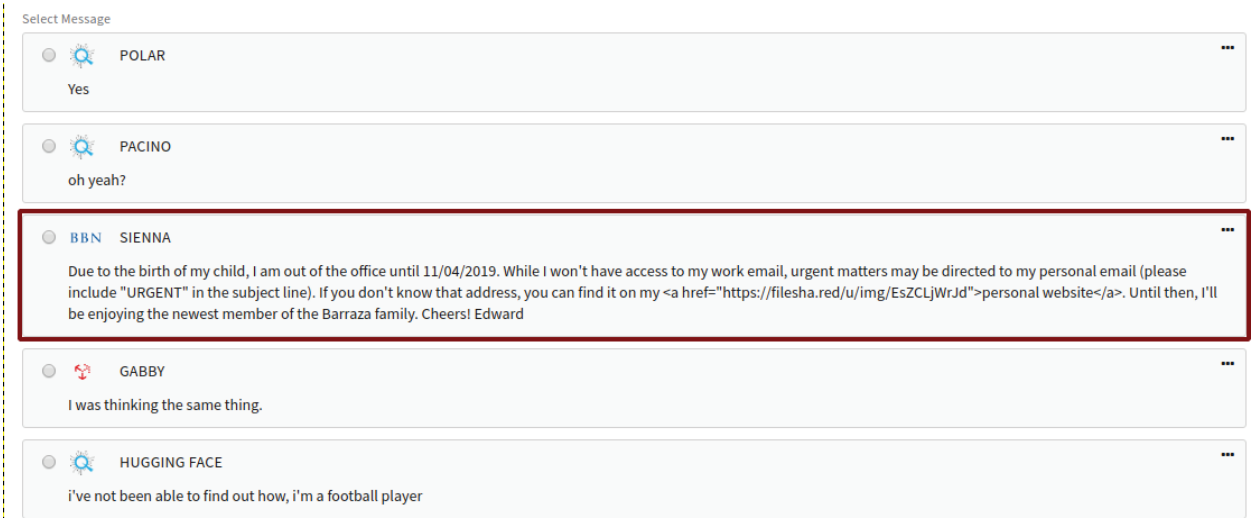


Figure 35. SIENNA-Supplied Suggested Reply for HITL Scenario

4.3.7 Integration with Continuously Habituating Elicitation Strategies for Social-Engineering-Attacks (CHESS) (Hughes Research Laboratory [HRL])

CHESS which is developed by HRL, is a TA2 system that ReCourse integrates with in order to provide human- in-the-loop operators with recommendations for engagement strategies. These recommendations are surfaced in the UI when the operator is preparing to engage with a suspected attack. The CHESS integration was used successfully in the HITL portion of the October evaluation.

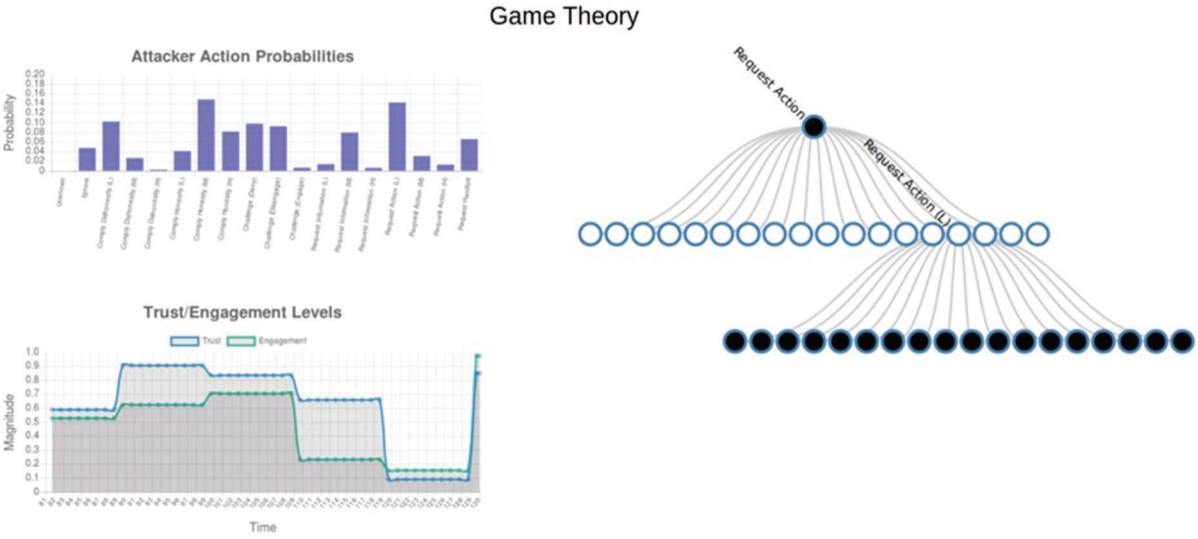


Figure 36. Example Game-Theory-Based Strategies from CHESS (image from HRL).

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

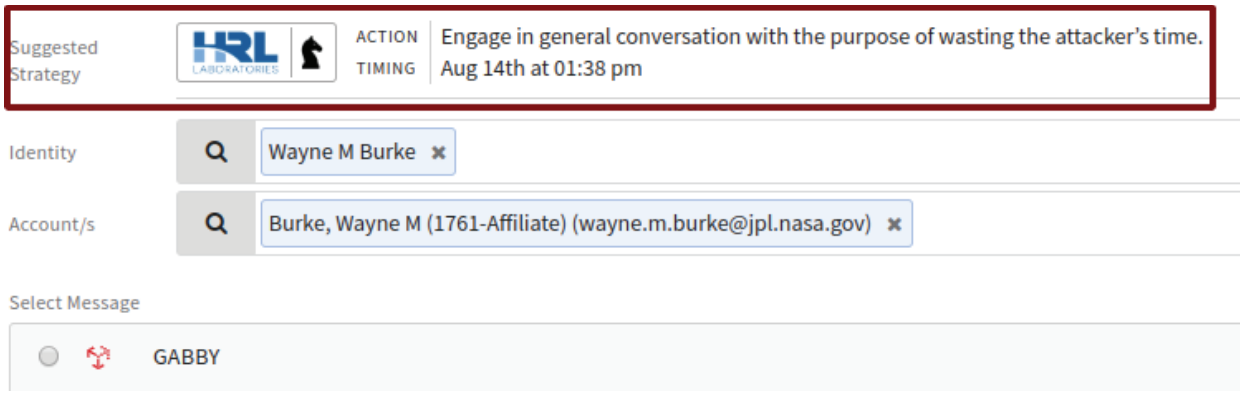


Figure 37. Example CHESS Results being Surfaced in ReCourse.

Future work was planned for ReCourse to serve as an integration point between HRL's CHESS and BBN's SIENNA, by providing the latter with strategy suggestions from the former to aid in the generation of appropriate responses.

4.4. ReCourse

ReCourse is a combined TA1+TA2 platform for situational awareness of the attack surfaces of the enterprise; scalable HITL bot and persona management for both detection and investigation; cross-channel monitoring and bot dialogue for detecting attacks; and automated and semi-automated cross-channel actor engagement for investigative information elicitation.

ReCourse creates new, generalizable, scalable methods for inclusion of human cognition and feedback in orchestrating novel conversational agents across enterprise channels.

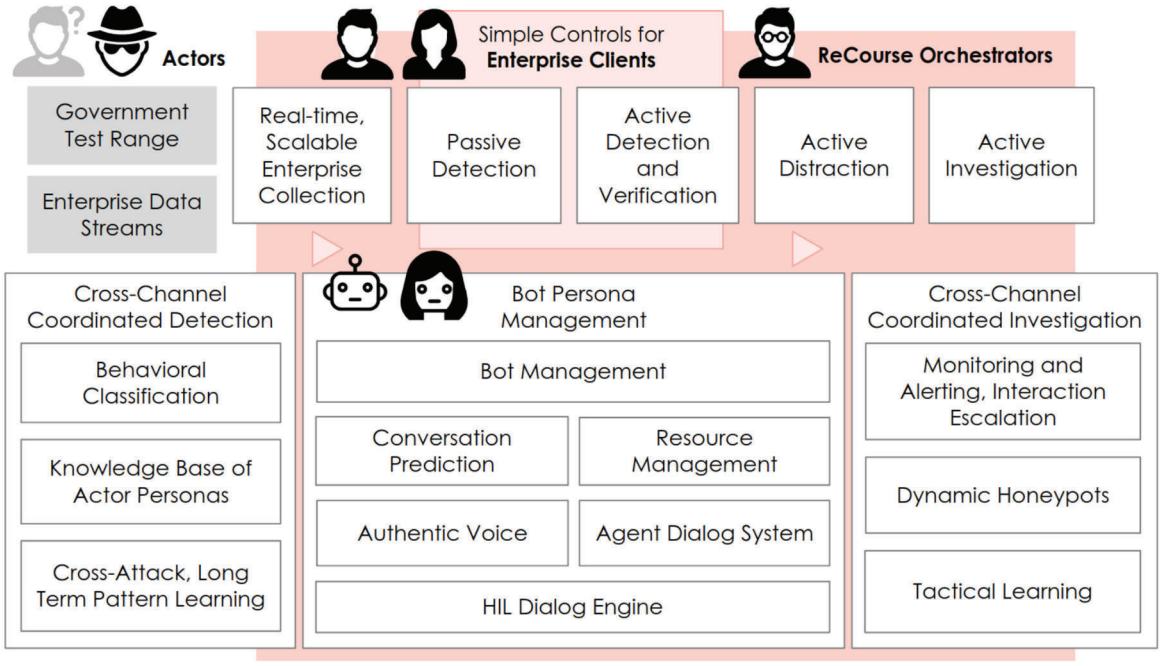


Figure 38. ReCourse Functional Architecture.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

4.4.1. Architecture

The ReCourse ecosystem consists of multiple microservices that communicate via RESTful APIs, gRPC [gRPC 2020] and Kafka [Apache Kafka 2017] topics. Messages come in on a Kafka topic and the Grapevine component manages the normalization and concurrent TA1 classification before sending those results to STIX and the augmented message to the ReCourse Knowledge Base. In autonomous mode, upon receiving an attack message in the Knowledge Base, ReCourse engages the TA2 systems to reply automatically, potentially with a Fingerprint link. The ReCourse UI surfaces the messages to an HITL operator and provides access to the TA1 results as well as suggested responses and response strategies from the TA2 systems. Responses are sent out using endpoints such as the Chute service.

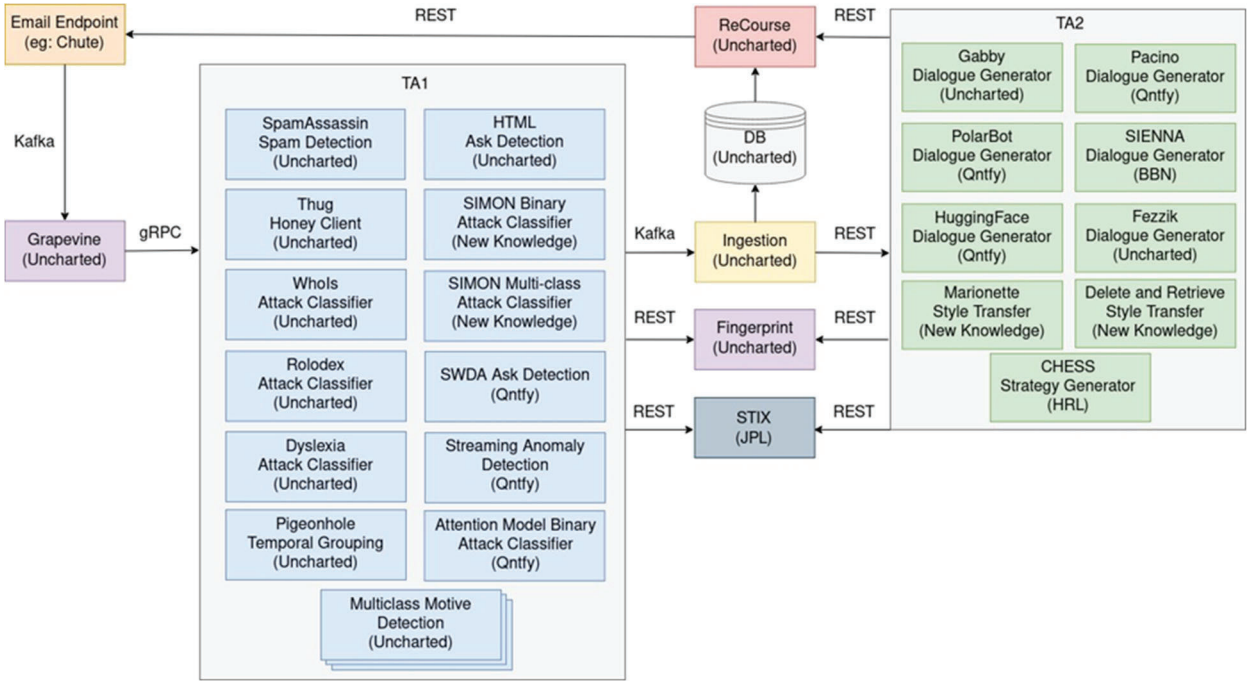


Figure 39. ReCourse System Architecture.

4.4.1.1. Grapevine

Grapevine provides streaming orchestration of the TA1 microservices. It pulls incoming messages off of a Kafka [Apache Kafka 2017] topic and normalizes them into a format that ReCourse uses across all channels and platforms. The normalized message is then sent concurrently to the various TA1 classifiers using gRPC [gRPC 2020]. As the classifications return, they are added to the message to allow them to be surfaced in the UI. Once all of the classifications have been collected, an ensemble approach is used to make the final determination of whether or not the message is an attack. The result of that decision is sent to the STIX endpoints and the normalized, augmented message is put on another Kafka topic for ingestion into the ReCourse database.

Grapevine also integrates with the Fingerprint application to send any information that was

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

obtained about an attacker to the STIX TA2 endpoint.

4.4.1.2. Ingestion/Autoreply

ReCourse's ingestion process serves two purposes: 1) it inserts the normalized messages into the database; and 2) it triggers the auto-reply process when the system is running in autonomous mode. In the latter case, if the message is part of a new thread, it randomly chooses a TA2 bot to engage the attacker. Otherwise, it attempts to reuse the same bot that was used previously. The automated responses may include links from the Fingerprint service mentioned above.

4.4.1.3 Chute

The chute microservice is a headless email client for Internet Message Access Protocol (IMAP) and Simple Mail Transfer Protocol (SMTP) It is designed to handle sending and receiving messages from many accounts in parallel, allowing ReCourse to communicate with external email servers. To send messages, the service exposes a RESTful API that translates ReCourse JSON messages into the email standard format [IETF 1996]. Given IMAP account configurations, Chute is able to read messages from email accounts and push them to Grapevine through a Kafka queue. In addition to connecting to standard IMAP and SMTP accounts, Chute was configured to send and receive messages from ASED evaluation infrastructure, Inbucket [Inbucket 2018] and Mail-in-a-box [Tauberer 2020]. The many configurations of this service enable rapid deployment of ReCourse to development systems, evaluation infrastructure and live environments.

4.4.1.4 Other components

4.4.1.4.1 Persona Management Platform (PMP)

This platform supports the curation, coordination, and programming of autonomous and semi-autonomous cross-channel conversational agents that can be deployed across a range of defensive capabilities, from detection, to information solicitation (honeypot personas), to active response. Specifically, the persona data model is defined by Name and SocialProfile data objects in which each SocialProfile object can represent a uniquely trained persona model.

The management platform itself handles HTTP requests to provide Create, Read, Update and Delete (CRUD) operations on a MongoDB instance containing the aforementioned personas. Available routes include getting specific personas, getting specific SocialProfile models, and adding connections between different personas.

4.4.1.4.2 Ibex Named Entity

This service is a wrapper for spaCy's named-entity recognition tool. Given a text document, Ibex recognizes and classifies named entity mentions into predefined categories (for example, a person, book title, country, product, etc.). The information extraction capability in the ReCourse system was used as a steppingstone for building a comprehensive information summarization tool, which was on Yonder's roadmap for upcoming evaluations.

4.4.2 Design

The design goals of ReCourse were to create a novel mixed-initiative platform to scalably coordinate, monitor and selectively moderate automated, conversational, enterprise-scale bots for defense against social engineering attacks. ReCourse combines advanced analytics with intuitive and scalable visualizations of activity to deliver threat awareness and unprecedented capability to

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

evaluate and shape bot tactics at the global enterprise level. An HITL system ensures ongoing adaptation to changes in adversarial tactics, and elimination of false positives, ultimately leading to dramatically improved success rates in the defense against social engineering.

The vocabulary, goals and workflows of users were collected through monthly and quarterly workshops with user representatives.

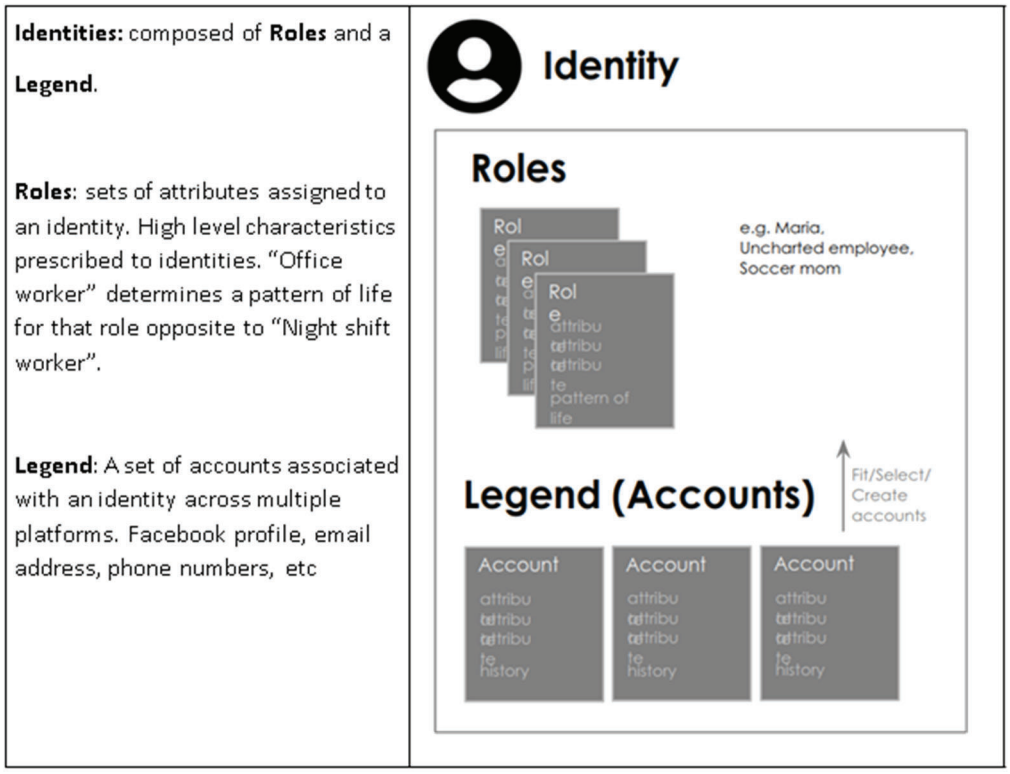


Figure 40. User-Driven Nomenclature for Identities.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Goals: e.g,. “increase trust,” “determine associated actors”

Strategies: “prolong conversation,” “ask for more information”

Resources: “click this link,” “open this file,” phone number used for texting, alternative **Account** resources

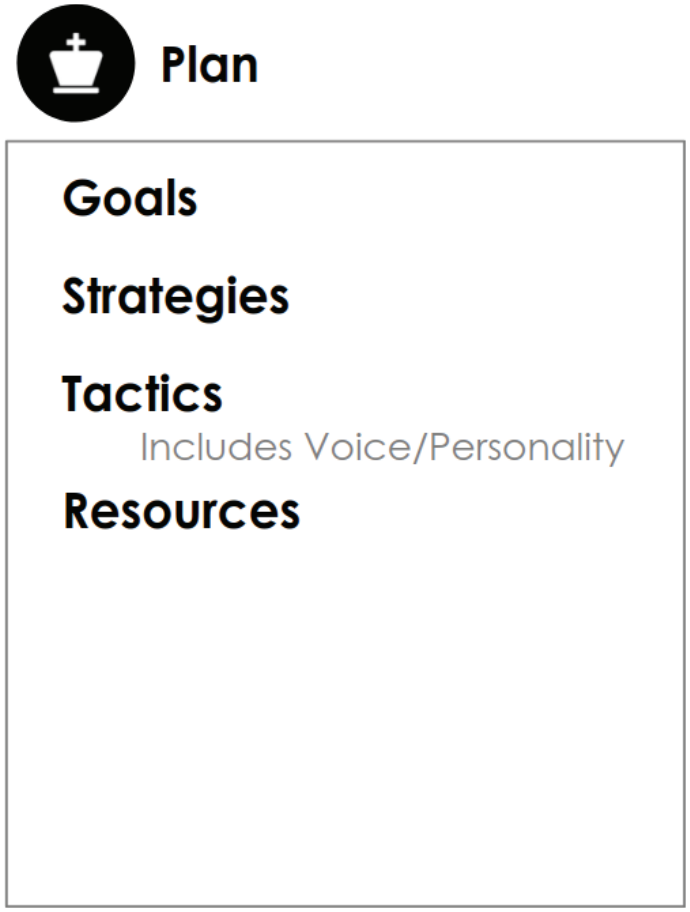


Figure 41 User-Driven Nomenclature for Plans

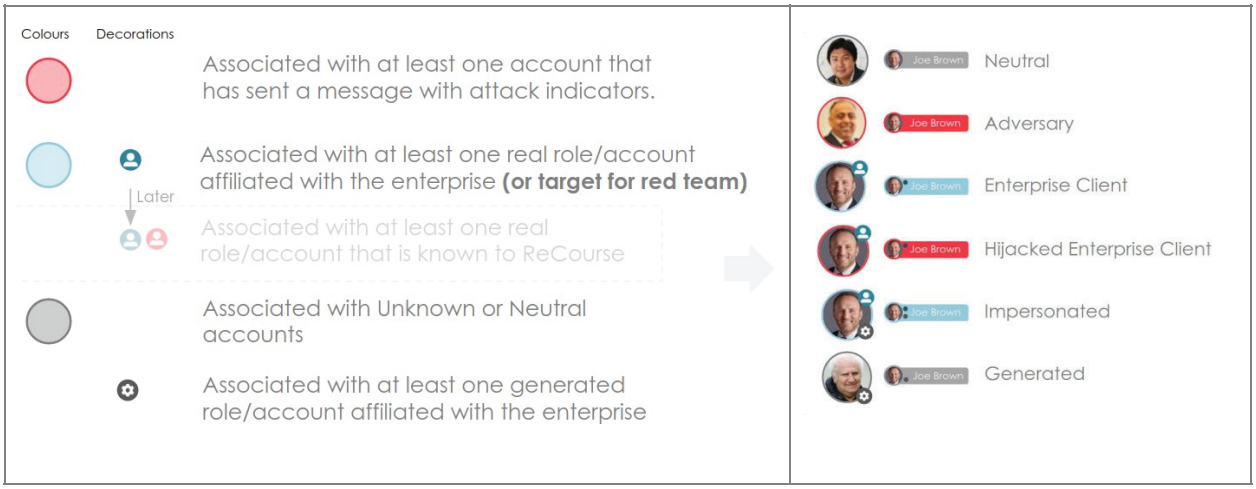
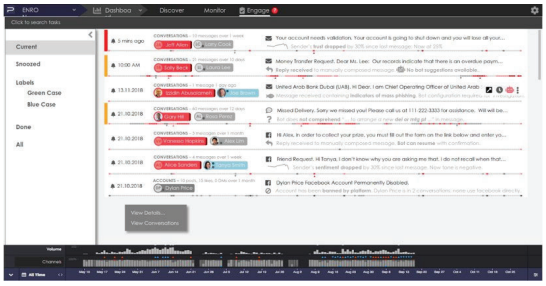


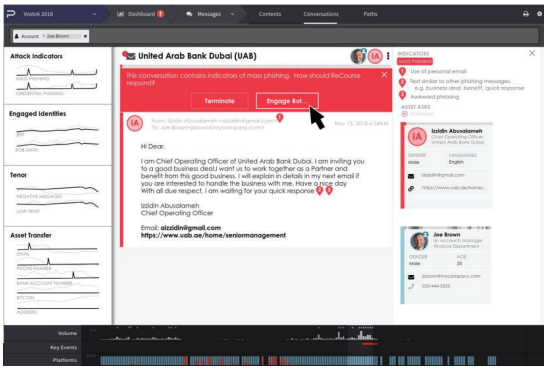
Figure 42. Visual Vocabulary for States and Actor Types

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

4.4.2.1 Engagements View

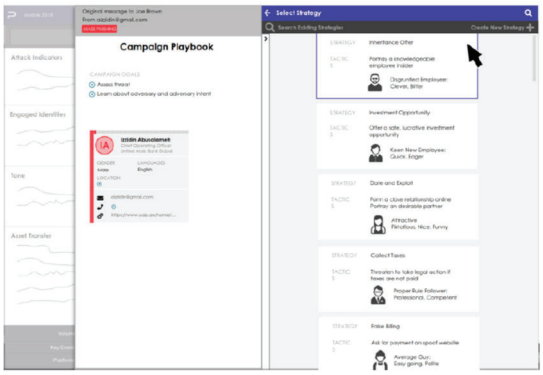


The engagements view provides a triaged view of the most important conversations and engagements requiring immediate user attention. The allows an orchestrator to see the automated conversations that require user intervention that cannot be solved by automated bot strategies. For example, acquiring a phone number resource for this conversation to continue, filling out a custom form that a bot cannot understand, or manually regaining trust in a conversation where trust has been lost. From the engagements view, an orchestrator may pivot into the messages view in order to gain full context of the conversation.

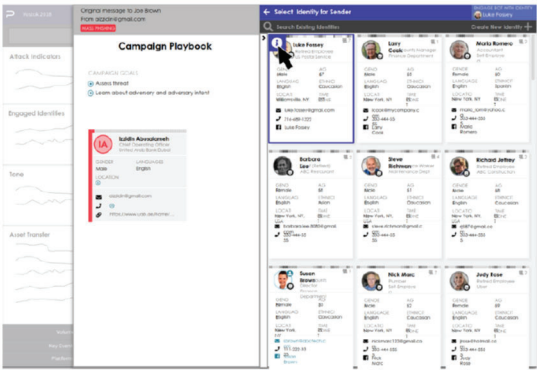


Additionally, the engagements view shows high-risk incoming attacks into the system. The orchestrator can choose to engage a by pivoting to the conversations view.

4.4.2.2 Bot Engagement Workflow



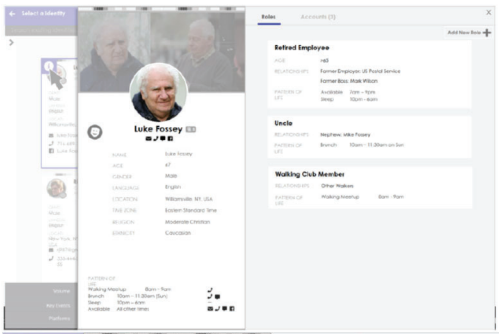
Upon bot engagement, we first select a strategy to construct the campaign playbook. The playbook can automatically detect goals such as assessing threat and extracting additional information to inform the attackers intents. Strategies are suggested by the system to coincide with the goals of the engagement campaign. Strategies will have a tactic, for example, “portray a knowledgeable insider” that will guide our conversation bots on how to engage.



Identities are recommended and chosen by the orchestrator. Additional identities can be created from within the identity creation workflow or tailored to suit the needs more specifically.

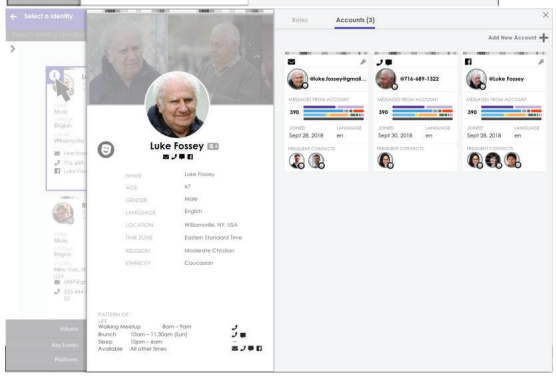
Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Roles assigned to the identities are shown in the details view. Aggregate attribute information is shown about the roles and accounts. Additional roles or information may be manually entered by the orchestrator.



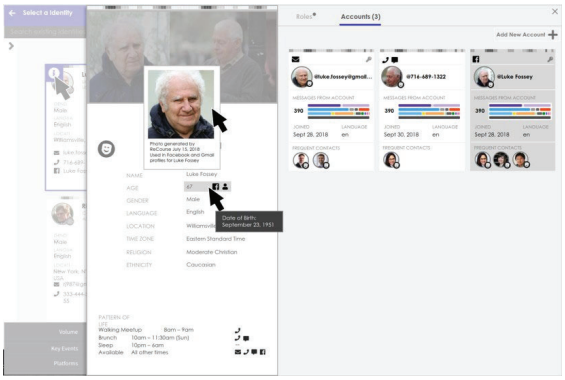
Accounts added to the identities are shown in the details view.

Credentials such as usernames/ passwords for the accounts are accessible from the account cards. An orchestrator may also request new accounts.

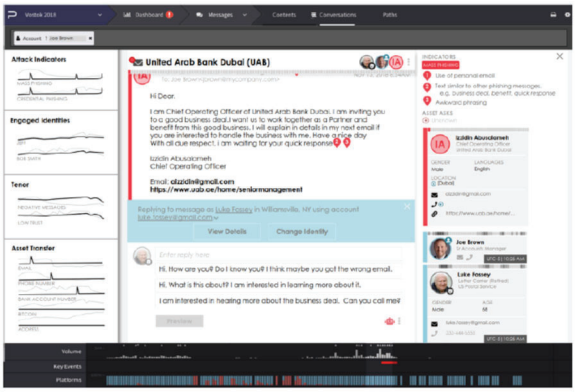


Provenance of attributes and photos are shown in the account details.

These attributes may be manually entered or can be synced from the accounts themselves. Attributes that have been synced from the accounts themselves (thus, committed to the legend) are not editable by the orchestrator.

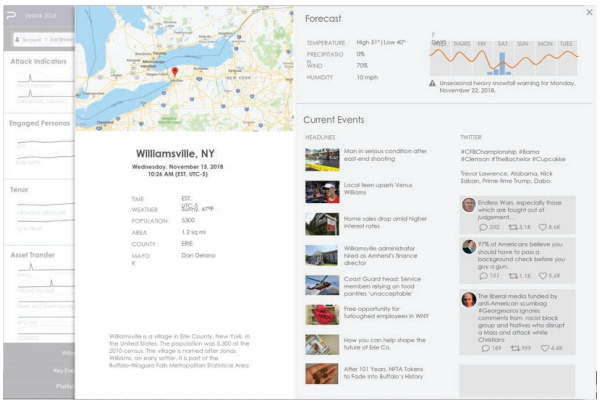


Once an identity has been set up/ selected, it is a ready for engagement. Orchestrators and users can guide bot replies using recommendations and context summaries. Replies to the conversation may be manually composed.



Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Locale and interest-based information views are available from the conversation. This can help users and orchestrators in guiding bots and manually editing conversation suggestions.

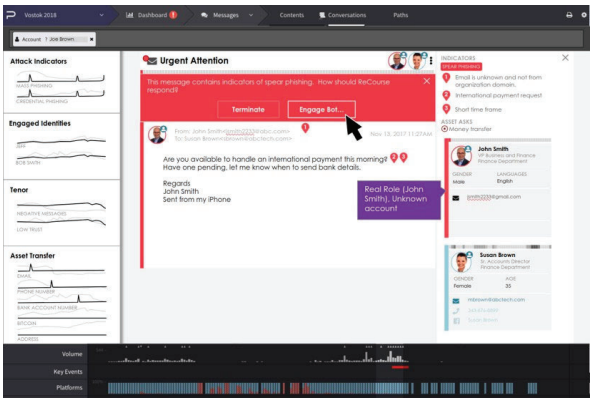


4.4.2.3 Conversations View

Incoming activity is flagged by the system as suspicious. A user can choose to engage a bot via the bot engagement workflow described above. Indicators are shown clearly displaying to the user why the system has flagged the conversation as suspicious. This includes multiple detection models as well as ask detection to discover the attacker's intents. Summaries of attacks types can be easily visualized along the context bar at the top of the conversation view. This shows the flow of the conversation over time as well as highlighting the attacks detected by the system.

4.4.2.4 Experimental Sandbox

ReCourse was designed to detect and defend the enterprise against social engineer attacks. Much of the training was done using offline data sets. However, to test the real-time effectiveness of the solution, and put it through real user scenarios, an Experimental Sandbox mode was created. In Sandbox mode, two ReCourse instances were created - a Blue team and a Red team - and their event streams were connected. This allowed humans, analytics and bots to interact in a closed system. Human operators could stage real attacks using ReCourse Red, and we could observe how ReCourse Blue reacted.



4.5. Program Results

4.5.1 Orchestration and Deployment

All deployments were automated through Ansible, containerized and orchestrated through Docker, Swarm and Kubernetes.

4.5.1.1 Docker

We used Docker for containerization. The ReCourse system was very container heavy, having 41 different containers to deploy, comprising all the different components from Uncharted and our subcontractors and research partners. Docker was chosen as it is the industry standard for containerization, and Uncharted has a lot of experience with it.

4.5.1.2 Swarm

For the dry run, we used Swarm for orchestration. We also used Swarm for our local testing

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

deployment. Swarm was chosen as it is compatible with docker-compose. It is also very easy to run the Swarm locally for testing purposes.

Our local testing deployment consisted of deploying to Swarm running on our on-site OpenStack cloud. In addition to running all of the services required for ReCourse, our on-site testing deployment also needed to run a kafka service, which ReCourse requires to process incoming and outgoing messages. This was required to facilitate communication between the Red Team and Blue Team versions of ReCourse. We used the wurstmeister/kafka docker container for this, which was Swarm ready [Wurstmeister 2020].

In the context of the March 2019 Dry Run, we used Ansible to automate the provisioning of virtual machines (VM) in the OpenStack environment provided. We then provisioned a Docker Swarm on the created VMs in order to install the required containers and stacks to operate the various ReCourse systems. For the dry run evaluation, a Kafka cluster was provided with topics pre-made for each performer.

4.5.1.3 Kubernetes

For the evaluation, we were required to re-implement our orchestration in Kubernetes. Also, all our services would need to run as a specific user inside the container. This required us to recreate most of our containers to support running as a non-root user and support the user we would be required to run as. In some cases, this was as simple as just changing the user in a dockerfile, but some containers required changes to the underlying code to function.

The changeover to Kubernetes required us to convert all our yaml files for swarm to kubernetes manifests. Although both of these files are Yami Aint't Markup Language (YAML) files, they are not compatible with each other. There are some tools that will do the conversion for you that we investigated, however they were not very reliable and so it was determined fairly early on that we would have to hand convert all our swarm YAMLs to manifests.

All our services were set up in Kubernetes as deployments and services, enabling us to quickly modify deployments as needed and have Kubernetes take care of tearing down old versions and putting up new ones.

The networking model in Kubernetes is 'simpler' in that every container can just see the other with no extra setup needed by us, which made this part of working with Kubernetes easy.

During the evaluation, it was realized that we would need a way to back up our progress in case of a system failure, since our database had to run inside a container running in Kubernetes.

Kubernetes provides a system for running jobs, where on a specified interval a container can be spun up, ran and then shut down. We took advantage of this to create a job that would dump our database to a file to solve the issue of backups.

4.5.2 Dry Run Evaluation

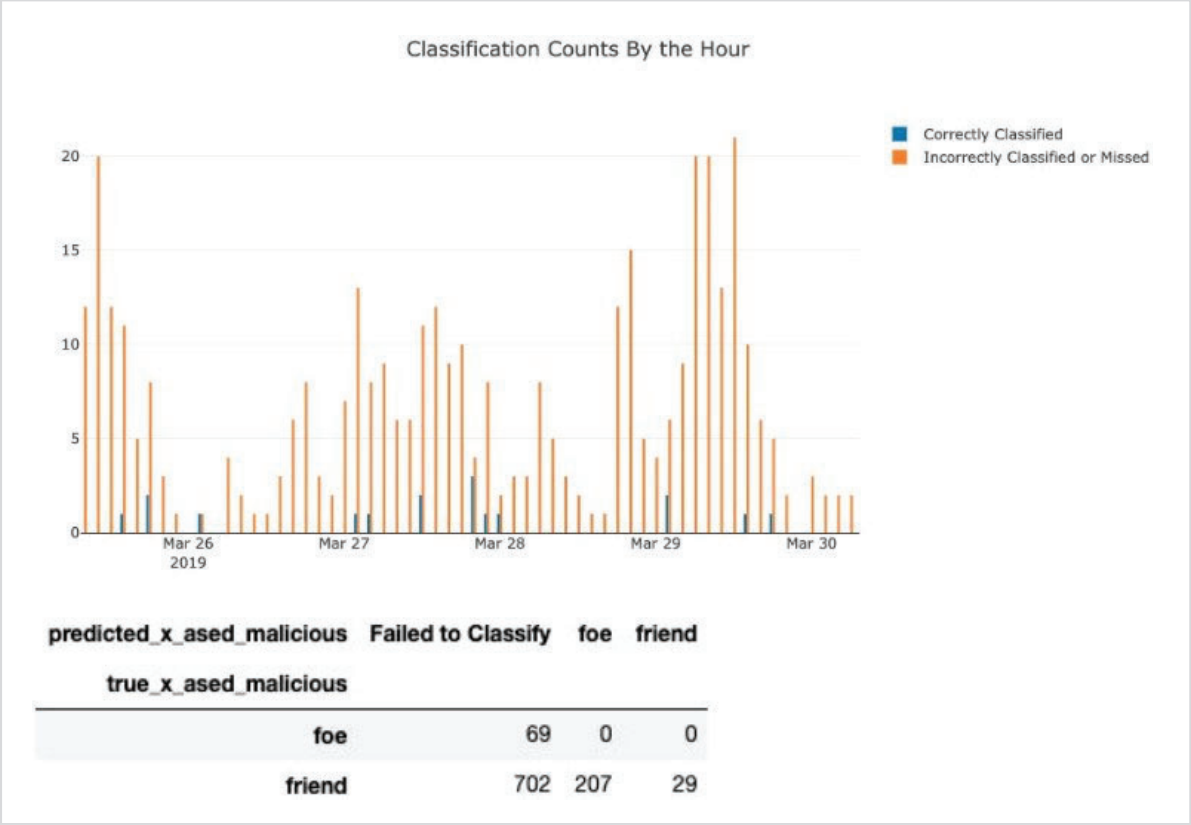
ReCourse was one of the five combined TA1/TA2 systems to participate in the Dry Run Evaluation.

In the TA1 component, it did not succeed in detecting any of the foe messages and misclassified a portion of friend messages as foe. As such, its metrics were as follows:

- Foe Classification Precision: 0.0
- Foe Classification Recall: 0.0

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

- Foe Classification F-Score: 0.0
- Foe Classification Prob Detection / Prob False Alarm: ~0.14
- Severity Misclassification Cost: ~1.043



predicted_x_ased_malicious	Failed to Classify	foe	friend
true_x_ased_malicious			
foe	69	0	0
friend	702	207	29

Figure 43. Dry Run Evaluation TA1 Results (image from JPL).

In the TA2 component, ReCourse, like most performers, did not detect any of the seven threat actors.

Regarding system reliability, the report noted:

> "Correct classifications spread out over the course of the week suggests that the performer does have the capability to classify and submit bundles properly using the message IDs provided in the Kafka queues."

After the dry run evaluation, many improvements were made to ReCourse's TA1 and TA2 systems, which allowed it to perform much better in the program evaluation.

For the dry-run in March, we presented TA1 components focusing on supervised classification techniques for phishing detection. These components included the SIMON binary and multi-class text classifiers and the shapelet time series classifier. Of these components, we have shown results below on a held-outset from the dry-run for the SIMON binary text classifier and the RRCF streaming anomaly detection (presented at the August 2019 workshop). We note that neither TA1 component was able to successfully identify the manually generated TRSS attack emails, which was likely substantially influenced by the lack of representative training data. To help alleviate this problem, we added the investigation of semi-supervised and unsupervised

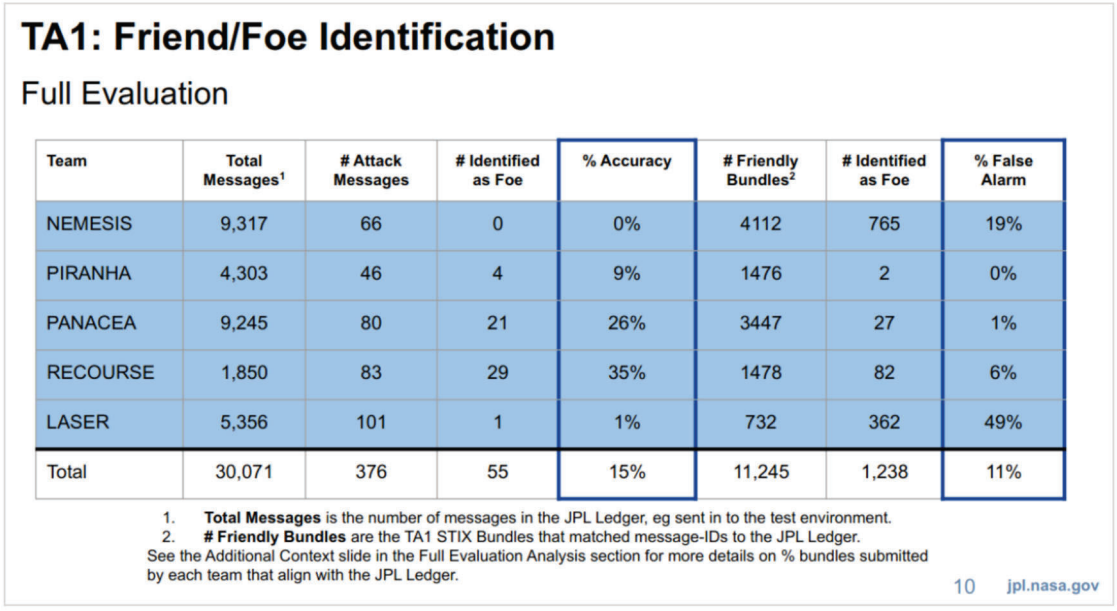
Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

approaches for phishing detection to our proposed research roadmap. Additionally, we also presented a demo of our shapelet classifier at the workshop.

4.5.3 Program Evaluation

The ReCourse system participated in both the full and the dialogue components of the summer evaluation and performed well in both. In particular, it was the top-performing system in the full evaluation.

In the full evaluation, in which performers were unable to access their systems, ReCourse had the highest accuracy at 35% while maintaining a low false alarm rate of 6%. The overall average for those two metrics were 15% and 11%, respectively.



TA1: Friend/Foe Identification

Full Evaluation

Team	Total Messages[1]	# Attack Messages	# Identified as Foe	% Accuracy	# Friendly Bundles[2]	# Identified as Foe	% False Alarm
NEMESIS	9,317	66	0	0%	4112	765	19%
PIRANHA	4,303	46	4	9%	1476	2	0%
PANACEA	9,245	80	21	26%	3447	27	1%
RECOURSE	1,850	83	29	35%	1478	82	6%
LASER	5,356	101	1	1%	732	362	49%
Total	30,071	376	55	15%	11,245	1,238	11%

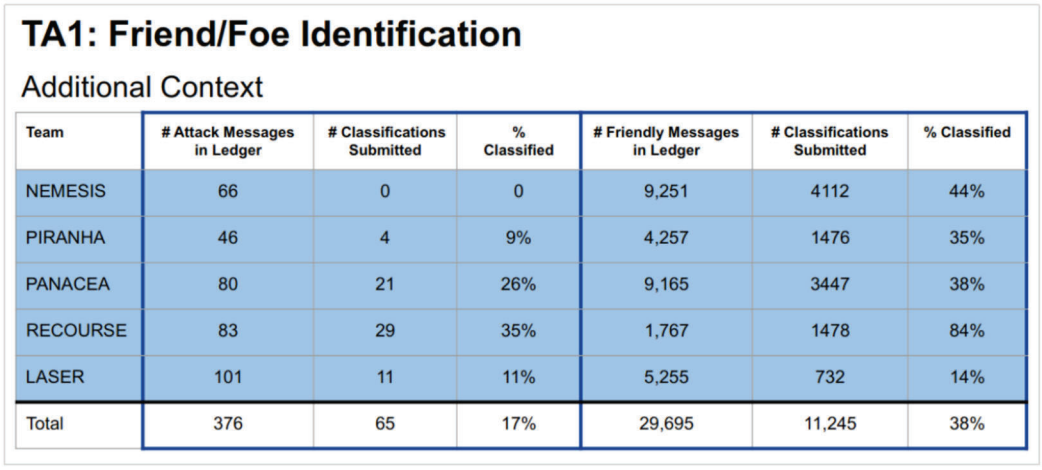
1. **Total Messages** is the number of messages in the JPL Ledger, eg sent in to the test environment.
2. **# Friendly Bundles** are the TA1 STIX Bundles that matched message-IDs to the JPL Ledger.

See the Additional Context slide in the Full Evaluation Analysis section for more details on % bundles submitted by each team that align with the JPL Ledger.

10 jpl.nasa.gov

Figure 44. Full Evaluation TA1: Friend/Foe Identification (image from JPL).

ReCourse also had the highest classification rate for friendly messages at 84%, compared to the overall average of 38%.



TA1: Friend/Foe Identification

Additional Context

Team	# Attack Messages in Ledger	# Classifications Submitted	% Classified	# Friendly Messages in Ledger	# Classifications Submitted	% Classified
NEMESIS	66	0	0	9,251	4112	44%
PIRANHA	46	4	9%	4,257	1476	35%
PANACEA	80	21	26%	9,165	3447	38%
RECOURSE	83	29	35%	1,767	1478	84%
LASER	101	11	11%	5,255	732	14%
Total	376	65	17%	29,695	11,245	38%

Figure 45. TA1: Full Evaluation Friend/Foe Identification - Additional Context (image from JPL).

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

In the TA2 component of the full evaluation, ReCourse was the only team to respond to attackers. It responded to 32 messages with one response passing the program's Turing Test.

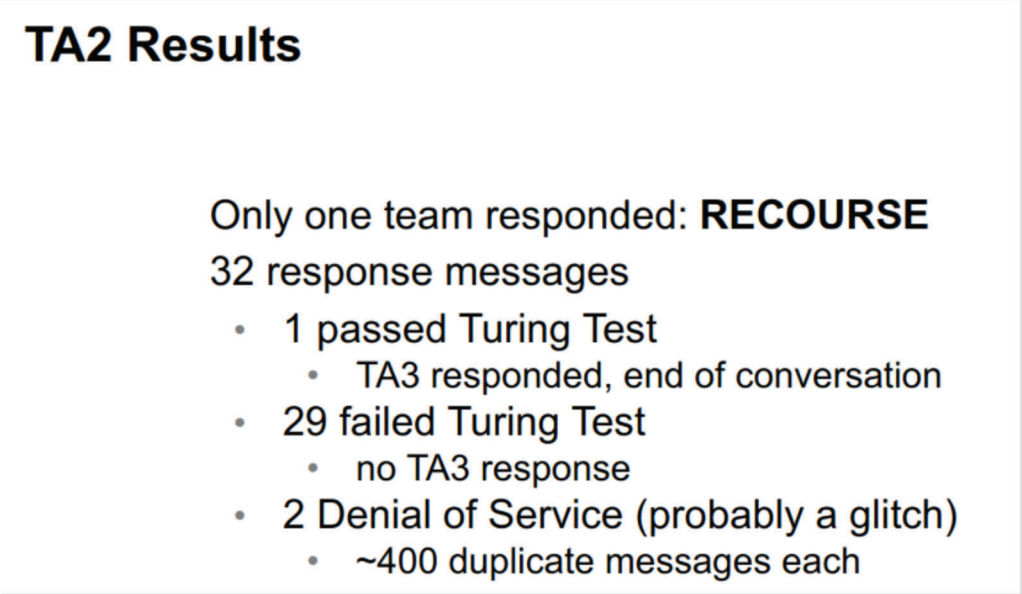


Figure 46. Full Evaluation TA2 Results (image from JPL).

There was a technical issue with two of the responses getting sent repeatedly, however it was eventually solved through coordination with Data Machines.

Additionally, ReCourse had the highest threat actor identification rate at 89%, compared to the overall average of 27% (with three teams getting 0%).

TA2 - Full Evaluation (w/out message ID massaging)

Team	Bundles Submitted[1]	Message ID matches[2]	TA3 Threat Actor Count	TA2 Threat Actors Attributed[3]	TA2 Threat Actor Identification Rate[4]
NEMESIS	22653	0	9	0	0%
PIRANHA	326	4	9	4	44%
PANACEA	3	0	9	0	0%
RECOURSE	186	**19**	9	**8**	**89%**
LASER	15367	0	9	0	0%
Total	37571	24	26 (shared)	12	**Avg:** 27%

1. Number of TA2 bundles submitted during the evaluation period (9/23/19 - 10/22/19). Could include historical emails. TA2 bundles contain multiple observed-data objects (messages) and thus Message-IDs.
2. Number of Message-IDs contained in the TA2 bundles that match those in TA3 bundles, aka ground truth. TA3 provided 109 Message-IDs.
3. Against provided TA3 attribution data (excludes additional attributions made by TA2s).
4. Based not on threat-actor IDs but on a TA2s ability to make an attempt in attributing detected threat actors. This percentage is not reflective of a TA2s ability to associate particular emails with a threat actor.

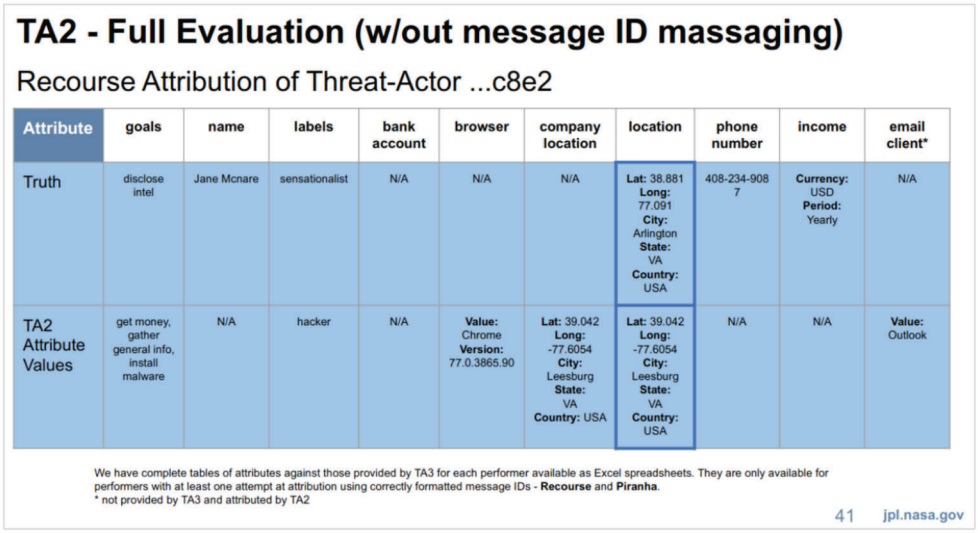
40 jpl.nasa.gov

Figure 47. Full Evaluation TA2 Results Continued (image from JPL).

Using TA2 components like Fingerprint, ReCourse was able to obtain identifying information

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

about TA3 attackers. This included attributes provided by TA3, such as location, and data points not provided, such as browser and IP address.



TA2 - Full Evaluation (w/out message ID massaging)

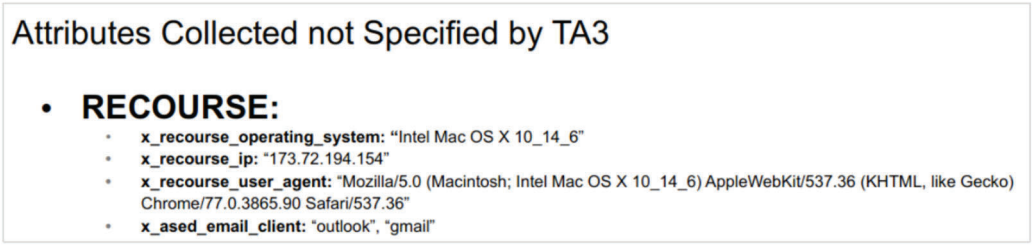
Recourse Attribution of Threat-Actor ...c8e2

Attribute	goals	name	labels	bank account	browser	company location	location	phone number	income	email client*
Truth	disclose intel	Jane Mcnare	sensationalist	N/A	N/A	N/A	**Lat:** 38.881 **Long:** 77.091 **City:** Arlington **State:** VA **Country:** USA	408-234-9087	**Currency:** USD **Period:** Yearly	N/A
TA2 Attribute Values	get money, gather general info, install malware	N/A	hacker	N/A	**Value:** Chrome **Version:** 77.0.3865.90	**Lat:** 39.042 **Long:** -77.6054 **City:** Leesburg **State:** VA **Country:** USA	**Lat:** 39.042 **Long:** -77.6054 **City:** Leesburg **State:** VA **Country:** USA	N/A	N/A	**Value:** Outlook

We have complete tables of attributes against those provided by TA3 for each performer available as Excel spreadsheets. They are only available for performers with at least one attempt at attribution using correctly formatted message IDs - **Recourse** and **Piranha**.
* not provided by TA3 and attributed by TA2

41 jpl.nasa.gov

Figure 48. Full Evaluation TA2 Attribution Example (image from JPL).



Attributes Collected not Specified by TA3

- **RECOURSE:**
  - **x_recourse_operating_system:** "Intel Mac OS X 10_14_6"
  - **x_recourse_ip:** "173.72.194.154"
  - **x_recourse_user_agent:** "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90 Safari/537.36"
  - **x_ased_email_client:** "outlook", "gmail"

Figure 49. Full Evaluation TA3 Attributes obtained by ReCourse (image from JPL).

ReCourse also succeeded in capturing flags in both the bot-only and human-bot threads of the dialogue evaluation.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

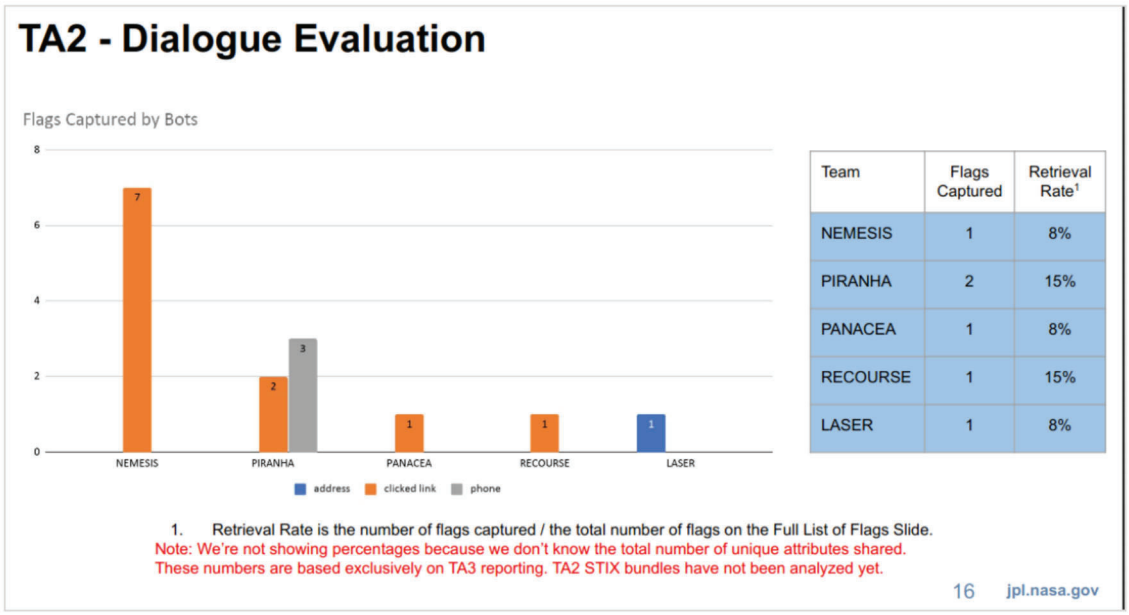


Figure 50. Dialogue Evaluation: Flags Captured by Bot Systems (image from JPL).

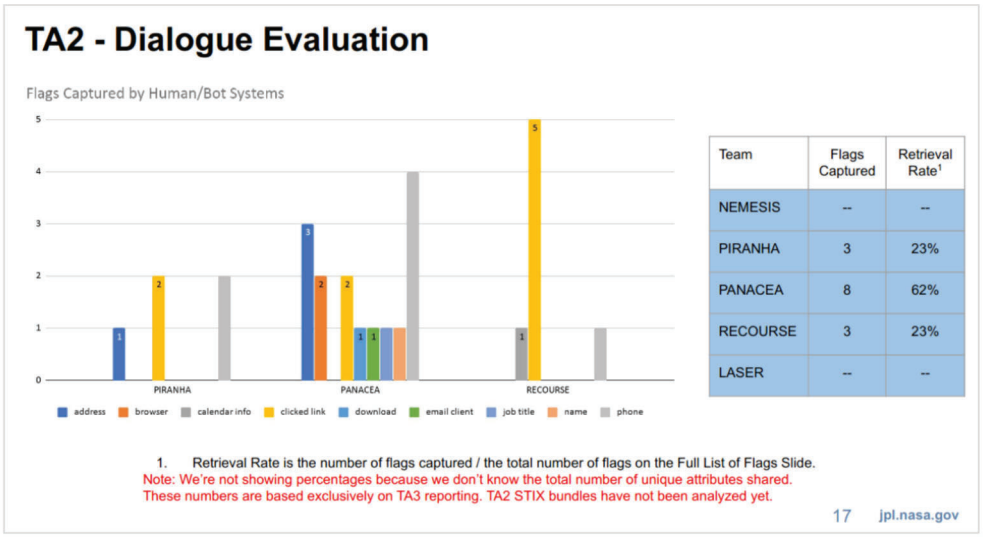


Figure 51. Dialogue Evaluation: Flags Captured by Human/Bot Systems (image from JPL).

Similar to the situation in the full evaluation, in the dialogue evaluation ReCourse captured attributes about the attackers that are not reflected in the figures above.

Regarding the bots, the TA3 team reported that the responses from ReCourse tended to be simple and did not move the conversation forward.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

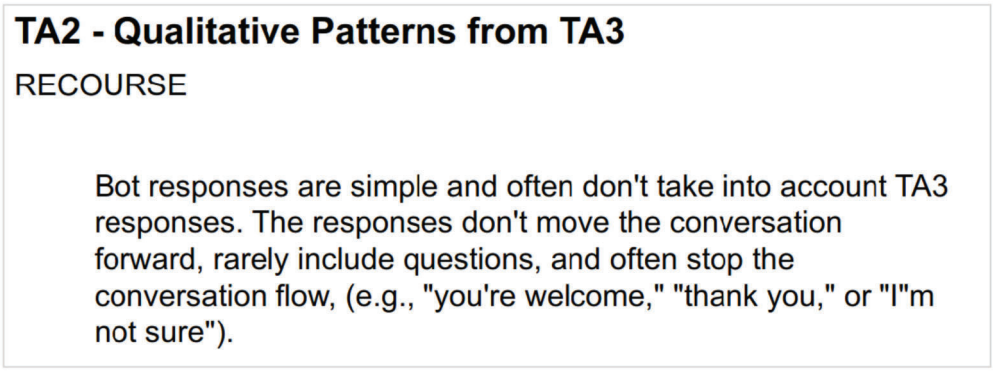


Figure 52. Dialogue Evaluation: Qualitative Patterns from TA3 (image from JPL).

The work being done by Uncharted on Fezzik was intended to address this feedback by taking the conversational context into account and outputting more varied responses through the use of Nucleus Sampling.

Yonder built two natural language style transfer systems (Marionette and Ventriloquist) that transformed text from a dialogue system to match the target style. Marionette implemented two personas (i.e., target styles), formal and informal, while Ventriloquist, implemented four personas, romantic, humorous, formal, and informal. We were the only team to produce a thorough style transfer system and we believe that it has the potential to be a valuable asset for the program. In addition to the TA2 components, we extended the SIMON binary text classifier (friend, foe) to a multi-class motive attack classifier (acquire credentials, acquire personally identifiable information (PII), annoy recipient, build trust, elicit fear, access social networks, gather general information, get money, and install malware). We also submitted the RRCF streaming anomaly detection algorithm, which is an attractive component for ASED because it works on streaming data, performs well on high-dimensional time-series, handles duplicate effectively, and presents an anomaly score with a clear statistical meaning.

Qntfy participated in multiple program-wide workshops and evaluation events. intended to assess research progress and comparative system performance. These activities included a “dry-run” evaluation where procedures were developed for testing and software integration. At the Fall 2019 Evaluation, Qntfy delivered a malicious message classifier as well as multiple functioning dialogue systems for integration in the ReCourse tool. The ReCourse system provided the ability to employ multiple dialogue systems in concert. To this end Qntfy delivered an API for an open-source transfer learning-based conversational model developed by HuggingFace [Wolf 2019].

This model provided a baseline capability for near-state-of-the-art dialogue. We also provided a dialogue model trained on the Cornell Movie Dialogue Corpus [Danescu-Niculescu-Mizil 2011]. This system, named Pacino, is a fused language/sequence-to-sequence dialogue model which is described in further detail elsewhere in this document.

4.6 Publications

- Hagerman et al. (2019, July). Visual Analytic System for Subject Matter Expert Document Tagging using Information Retrieval and Semi-Supervised Machine Learning. International Conference Information Visualization, Paris, July 2019
- Dhamani et al. (2019, June). Using Deep Networks and Transfer Learning to

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Address Disinformation. Poster presented at the AI for Social Good Workshop at the International Conference of Machine Learning, Long Beach, CA.

- We submitted our research on fused dialogue models to the 2020 Conference for the Association of Computational Linguistics. See Attachment 1 for full submission.

4.7. Open Source Repositories

The following open source repositories were created for applications and components developed under the ASED program:

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Table 1. Open Source Repositories

ReCourse-App • Chess (HRL) Integration • Chute (Email Actuator) • Fingerprint • Goth • Grapevine (pipeline) • News API • Seemail • Stix Integration	https://github.com/unchartedsoftware/recourse-app
ReCourse Deployment	https://gitlab.ased.io/cdickson/recourse-deployment
SIMON Text Classifiers	https://github.com/uncharted-recourse/NK-email-classifier
Streaming Anomaly Detection	https://github.com/uncharted-recourse/Streaming-Anomaly-Detection
Shapelet Classifier	https://github.com/NewKnowledge/sloth/blob/master/Sloth/classify.py
LSTM-FCN	https://github.com/NewKnowledge/LSTM-FCN
Ibex Named Entity	https://github.com/uncharted-recourse/d3m_ibex
Persona Management Platform	https://github.com/uncharted-recourse/persona-management-platform
qntfy-ask-detection	https://github.com/uncharted-recourse/qntfy-ask-detection
polarbot	https://github.com/uncharted-recourse/polarbot
polarbot-null	https://github.com/uncharted-recourse/polarbot-null
keras-spam-predictor	https://github.com/uncharted-recourse/keras_spam_predictor

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

5.0 CONCLUSION

For the ASED program, Uncharted created ReCourse, a novel mixed-initiative platform to scalably coordinate, monitor and selectively moderate automated, conversational, enterprise-scale bots for defense against social engineering attacks. We designed intuitive and scalable visualizations of activity to deliver threat awareness and unprecedented capability to evaluate and shape bot tactics at the global enterprise level. We integrated advanced analytics into an HITL system to ensure agility and adaptation to changes in adversarial tactics.

We built a platform for situational awareness of the attack surfaces of the enterprise; for scalable HITL bot and persona management for both detection and investigation; for cross-channel monitoring and bot dialogue for detecting attacks; and for automated and semi-automated cross-channel actor engagement for investigative information elicitation. ReCourse is also a strong platform for Red Team exercises for evaluating, improving and learning about the effectiveness of these strategies.

Uncharted merged best-of-breed and novel components and systems for classification and detection of asks and attacks. We created state-of-the art conversational bots and integrated them with goal-based orchestration to augment passive techniques of identification and attribution.

Yonder submitted three TA1 components: the SIMON binary text classifier, the SIMON multi-class text classifier, and the RRCF streaming anomaly detection algorithm. Despite not having much success identifying manually generated TRSS emails, both TA1 components

perform fairly well on held-out validation sets from the March 2019 dry-run. Additionally, we also developed time-series classification components (shapelet and Long Short-Term Memory Fully Convolutional Networks [LSTM-FCN] classifiers) and the Ibex-named entity recognition tool. Yonder's primary contributions to the TA2 components are the two style transfer systems (Marionette and Ventriloquist). We note that we are the only team in the ASED program who developed a comprehensive style transfer system and believe that it has the potential to be extremely beneficial to the program. To support the style transfer system(s), we started building an exhaustive persona management platform. Lastly, Yonder also supported the ASED program by curating social media datasets with user interaction for ask-detection research. Yonder's planned contributions included unsupervised detection methods, graph-based multi-channel models and information diffusion techniques, an enhanced style transfer system and persona management platform, and additional curated social media datasets.

Qntfy's research agenda achieved promising results in both the attack detection and investigation domains. While our larger research goals were more ambitious than what was achieved in limited development time, this work laid the basis for what could become powerful analytic solutions.

The broader ReCourse system also took a mixed-initiative approach solving the stated ASED program goals and the methods described here would fully support that approach. Academic research focused on large language models, dialogue, and reinforcement learning is advancing faster than it ever has before. This suggests that major advancements, some of which are described in the preceding recommendations section, are entirely attainable.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

6.0 RECOMMENDATIONS

Uncharted recommends an open research challenge in the form of several hackathons building on the work of the ASED program. This challenge could bring diverse academic and industry partners together to provide more realistic datasets and apply lessons learned from the program and industry at large, and resultant technologies developed to advance the agenda of the program. Seedling efforts could result from this approach to address specific threat vectors such as entity verification, validation of media (e.g. linking efforts with the Semantic Forensics (SemaFor) DAPRA program), propaganda and narrative detection developed under the DARPA QCR program, ongoing SocialSim program, and upcoming DARPA programs.

The biggest challenge that Yonder faced in building the TA1/TA2 components is the lack of available training data. As previously mentioned in the discussion section, neither TA1 component had much success in identifying manually generated TRSS emails despite presenting high accuracy on the validation set because of the misalignment between the attack messages from the test set and the messages that the classifiers were trained on. We recommend giving the performers access to similarly generated emails on their systems and/or encourage performers to explore unsupervised techniques in TA1. Similarly, large corpora for training language models do not tend to be representative of online communication nor do they capture the kind of "asks" that are expected of TA2 dialogue systems. To help overcome some of this, we submitted a Reddit thread dataset to the broader ASED program and we were exploring various other social media datasets at Yonder to submit for ask-detection research.

Additionally, a challenge associated with the training of the style transfer systems was finding appropriate "persona" datasets. Additionally, the curation of high quality "persona" datasets was a research problem in and of itself. The program could benefit from a working group dedicated to helping address these issues. In regards to training large deep learning models, ASED did not have any budget for computational resources, which are very costly. At Yonder, we weren't able to train our models as extensively as desired because of internal limitations on computational resources. We would recommend allocating either a budget for cloud computing costs or computational resources for teams. Lastly, it was difficult to define evaluation metrics for the dialogue systems (and in particular, the style transfer systems).

Although this is a current area of research, the primary evaluation metrics were neural machine translation metrics (e.g., the BLEU score). These metrics often rely on ranking "objectively better" translations which are less straightforward in open-domain dialogue and style transfer, where one is meant to potentially alter the text significantly. . Therefore, we would recommend that the evaluation team and a working group collaborate to define evaluation metrics that better assess the systems.

The existing development suggests several avenues for continued research and experimentation, particularly in the dialogue domain. Future TA1 work would greatly benefit from the collection and annotation of additional datasets. Email datasets (particularly for spam classification problems) were the predominant source of training examples for TA1 systems. This limited model development, as the program goals were largely focused on more sophisticated social-engineering attacks which often lack many of the characteristics of traditional spam attacks.

Additionally, while the program initially conducted integration and evaluation using attacks through email, multi-channel attacks were already planned for the next evaluation cycle. For

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

most of the expected attack vectors little or no data is available for the development of classification systems. The reliance on the STIX protocol and the requirement to report information about attack severity and type were also limiting factors. The scale of severity and classes of interest were poorly understood. Additional specification of system objectives would benefit further development in this area. Additionally, richer datasets that provide sequence labels, multi-class labels, and severity scores.

In particular, the following dialogue-related items are recommended for additional research, development, and integration with extant ReCourse technologies:

Further development of fusion methods for boosting model performance in low-resource domains. Initial experiments outlined in this document demonstrated that methods previously employed in machine translation systems could also be used to improve model performance on dialogue tasks. Language model fusion techniques help reduce training time and encourage realistic response generation. These methods are critical as training data specific to the dialogue domains of interest to ASED maybe limited or biased. The ability to improve model performance by using information from language models trained on very large corpora presents the opportunity to mitigate the risks posed by data limitations, while also reducing compute requirements. Additional fusion techniques could also be investigated including attentional interfaces between the language model output and the seq2seq decoder.

While we explored fusion techniques between language models and recurrent seq2seq models, we did not have the opportunity to experiment with self-attentive Transformer-based seq2seq models. These architectures have been shown to achieve state-of-the-art performance on their own, and it's possible that we could see even greater improvements in our own use case by applying the same language model fusion techniques with these Transformer models.

Finally, while we explored fusing pre-trained language models with seq2seq models on the Cornell Movie Dialogue corpus, there is a large set of untapped work that could better leverage transfer learning for this task. That is, in our experiments we pre-trained language models on the same training data that the seq2seq model was trained on. However, since language modeling requires no labels this technique could utilize any large-scale text corpus for pre-training, including longer passages available from sources such as Wikipedia or the web. Previous work transfer learning work in NLP has shown that pre-trained language models, or slot-filling models as in BERT, have helped boost performance on downstream tasks [Devlin 2018] [Radford 2018]. Pre-training these language models on even larger corpora could help speed up convergence of dialogue models and improve results even further since these language models would have been exposed to a large amount of text unseen by the dialogue model.

New decoder approaches for selecting responses better tuned for ASED use-cases. Traditional objective functions and decoder techniques are ill-suited for the types of dialogue required by ASED solutions. ASED dialogue systems need to generate diverse utterances and favor longer, richer responses, rather than short, high probably utterances. Research has demonstrated the efficacy of alternative objective functions [Li 2016] and decoding strategies, including in conjunction with large language models [Holtzman 2020]. While we had some success implementing these approaches, including beam search, top-k sampling, and nucleus sampling, future research should prioritize the further development of techniques and implementations that address this need [Roberts 2020].

Data and model architectures for goal-oriented dialogue applications.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

ASED use-cases for dialogue systems extend beyond chit-chat style models for time wasting purposes. The ability to strategically shape a conversation in order to build trust and elicit additional information from an attacker is also required. The capture-the-flag style dialogue evaluation performed at the Fall 2019 workshop formalized this requirement. This type of "distant goal" learning task is largely unstudied in dialogue systems research and open source datasets do not exist for the training of such systems. The most analogous task is negotiation dialogue, where the concept of dialogue rollouts has been shown to improve performance [Lewis 2017]. Goal-oriented dialogue systems have typically focused on tasks like question answering and information retrieval. Future work should consider how to develop corpora that are better suited for the ASED problem set. Additional data would enable the exploration of a range of approaches, including end-to-end systems, dialogue state managers, reinforcement learning approaches, and combinations of these methods.

Early work in open-domain dialogue has explored training end-to-end systems without using the typical supervised learning paradigm. This work seeks to address two problems stemming from the supervised learning approach: (a) dialogue models trained in a supervised fashion tend to produce bland or repetitive outputs, and (b) the standard supervised training regime does not optimize for long-term conversation-level objectives such as engagement or cohesiveness. These are challenging problems, but initial work has proposed reinforcement learning [Li 2016c] and generative adversarial training as possible solutions to these problems [Li 2017]. The adversarial learning approach is particularly interesting since it solves the need for a delayed reward by utilizing the output of the discriminator network as the reward signal for the generator network (e.g. the model producing the actual responses). More extreme attempts have tried to automate the process of obtaining reward signals through "model self-play" in which two models engage in dialogue without human involvement [Shah 2018]. Other techniques for obtaining delayed rewards for reinforcement learning have utilized automatic metrics such as sentence-level BLEU scores, and there are interesting opportunities in the social media domain where metadata signals could be incorporated as reward for training these models (e.g. number of likes on Twitter, number of up-votes on Reddit).

Further research on directed dialogue systems is also warranted. Research on this problem has focused on addressing issues like repetition, but also encouraging question-asking and other conversational behaviors [See 2019]. A mixed-initiative approach to sequences of dialogue turns with attackers could help systems meet program objectives. Similar approaches have been taken to develop dialogue managers for negotiation tasks [English 2005]. Furthermore, this is yet another area where very large language models have demonstrated significant promise [Raffel 2019]. The pretraining objective of Google's T5 model may be particularly suited for this type of controllable text generation. The Google team created a new task called sized fill-in-the blank which allows for sentence completion using a specified number of words.

6.1 Data Annotation

In addition to exploring the above modeling techniques, a significant portion of future work should focus on annotation of additional ASED-specific corpora for training new models. We present details for these suggested efforts below.

6.1.1 Task-oriented Dialogue

Obtaining annotations and dialogue examples for task-oriented settings would be most helpful to focusing the work of training further downstream chit-chat models that are more effective at

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

achieving many of the program's goals such as automatically eliciting information from an end-user. Such annotated data would also provide conversation-level information about what successful vs. unsuccessful dialogues look like, and these additional signals could potentially be incorporated into training future models.

The work of [Lewis 2017] proposes an interesting framework for collecting such data by framing the annotation task as a game between two participants. Each participant is presented with a common set of objects; however, each participant has different values attached to each object.

The goal is to have the two participants negotiate with one another in natural language through a text interface to gain the most objects that will maximize their end score (the sum of all points associated with the objects obtained for a given user). This data annotation setup "gamifies" the labeling task by putting users in a negotiation task, and thus users are not required to explicitly provide labels or understand the underlying machine learning application. As long as both users understand how to play the game, researchers can obtain labels through metadata associated with the negotiation sequence (e.g. the number of points accrued by a user) and the end result of the negotiation (e.g., successful vs. unsuccessful).

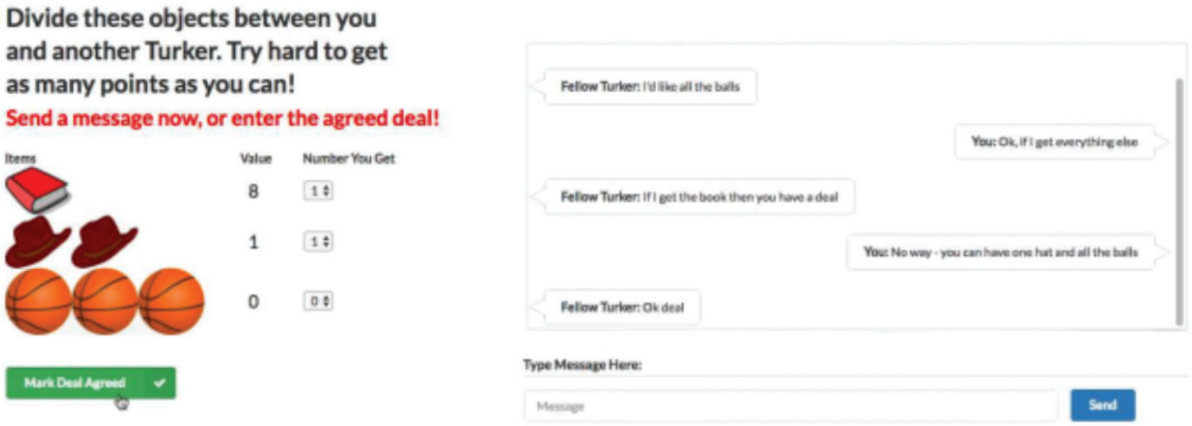


Figure 53. Annotation Interface for Dialogue Negotiation Task (Lewis 2017).

A very similar annotation game could be developed for the information-soliciting goals of the ASED program. Annotators could be given certain goals instead of objects as in the above work (e.g. obtain user's first name, get user to confirm address, etc.) and negotiate in a similar dialogue setting to try to obtain this information. These efforts could be streamlined with the "capture the flag"-style evaluation conducted in previous program evaluations and could inform how different pieces of information should be weighted (for example, perhaps obtaining login information is more valuable than getting a user to confirm their name). Furthermore, this annotation scheme isn't restricted to any domain of writing or conversation and is easily extensible to the social media or short message service (SMS) message scenarios.

6.1.2 Entity-Specific Sequence Labeling

In addition to the above efforts to facilitate task-oriented generative chatbots, having gold standard annotations for ASED-specific entities or linguistic entities would greatly facilitate both (a) automatic enrichment and detection of malicious messages, and (b) enrichment of dialogue models. These aspects could include entities such as account numbers and banks names, or program-specific actions such as "credential ask" or "solicit account info." Having fine-grained,

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

word-level labels for these entities would allow for training powerful sequence labeling models similar to those utilized in NLP problems such as a part-of-speech (POS)-tagging or named entity recognition.

Many techniques for these sequential models are available and there may even be opportunities for bootstrapping existing state-of-the-art models via transfer learning. An example of what such word-level entity labeling might look like is given in Figure 54 below, with words appearing in the top row and proposed example labels appearing on the bottom row.

I'm	sending	the	deposit	to	Bank	of	America	.
O	O	O	B-ACT	O	B-FIN	I-FIN	I-FIN	O

Figure 54. Example of Word-Level Tagging for Proposed ASED-specific Entities using Standard BIO format Common to NER Tasks.

B-ACT corresponds to "action word" while B-FIN and I-FIN are used to label individual words that constitute a financial institution.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

7.0 REFERENCES

Anti-Phishing Working Group. Phishing Activity Trends Report. http://docs.apwg.org/reports/apwg_trends_report_h1_2017.pdf. 2017.

Apache Kafka. Apache Kafka. Retrieved from https://kafka.apache.org/. 2017.

Apache SpamAssassin. Apache SpamAssassin: Welcome. Retrieved from https://spamassassin.apache.org/. 2018.

Asghar, N., et al. Deep Active Learning for Dialogue Generation. arXiv preprint arXiv:1612.03929v5 [cs.CL]. 2017.

Bangor, A., et al, An Empirical Evaluation of the System Usability Scale, International Journal of Human-Computer Interaction, 24(6), 2008

Beutel, A., et al. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. InProceedings of the 22nd international conference on World Wide Web 2013 May 13 (pp. 119-130). 2013.

Blum, A., et al. Lexical Feature Based Phishing URL Detection Using Online Learning. Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security, 2010.

Bordes, A., et al. Learning end-to-end goal-oriented dialog. arXiv preprint, arXiv:1605.07683v4 [cs.CL]. 2017.

Bothe, C., et al. A context-based approach for dialogue act recognition using simple recurrent neural networks. arXiv preprint arXiv:1805.06280. 2018.

Brockett, C., & Dolan, W. B. Support vector machines for paraphrase identification and corpus construction. In Proceedings of the third international workshop on paraphrasing (IWP2005). 2005.

Brooke, J., SUS: A "Quick and Dirty": Usability Scale, In Usability Evaluation in Industry, Jordan, B. eds, Taylor & Francis, London, 1996.

Chen, S., et al. The multivariate Hawkes process in high dimensions: Beyond mutual excitation. arXiv preprint arXiv:1707.04928. 2017.

Chu, W., et al. Protect Sensitive Sites from Phishing Attacks Using Features Extractable from Inaccessible Phishing URLs. IEEE International Conference on Communications (ICC), 2013.

Cohen, W.W. Enron Email Dataset. Retrieved from https://www.cs.cmu.edu/~enron/. 2015.

Cukierski, W. The Enron Email Dataset | Kaggle. Retrieved from https://www.kaggle.com/wcukierski/enron-email-dataset. 2016.

Danescu-Niculescu-Mizil, C. & Lee, L. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011. 2011.

Dell'Aera, A. buffer/thug: Python low-interaction honeyclient. Retrieved from https://github.com/buffer/thug. 2020.

Devlin, J., et al. Bert: Pre-training of deep bidirectional transformers for language understanding.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

arXiv preprint arXiv:1810.04805. 2018.

Dhamija, R. and J. Tygar. The Battle Against Phishing: Dynamic Security Skins. In Proceedings of the 2005 symposium on Usable privacy and security.2005.

Dhamija. R. and J. Tygar. Why Phishing Works. In the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press (CHI). 2006.

Dhingra, B., et al. Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access. arXiv:1609.00777v3 [cs.CL]. 2017

Dolan, W. B., & Brockett, C. Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005). 2005.

Dwork, C. Differential Privacy. ICALP'06 Proceedings of the 33rd international conference on Automata, Languages and Programming - Volume Part II. 2006

Dwork C. and A. Smith. Differential Privacy for Statistics: What We Know and What We Want to Learn. Journal of Privacy and Confidentiality: Vol. 1: Iss. 2, Article 2, 2010.

English, M. and Heeman, P. Learning mixed initiative dialog strategies by using reinforcement learning on both conversants. In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. 2005.

Evers, J. User Education Is Pointless. http://news.com.com/2100-7350 3-6125213.html. 2006.

Facebook Research. facebookresearch/ParlAI. Retrieved from https://github.com/ facebookresearch/ParlAI/tree/master/projects/personachat. 2019.

Gallos, L.K., et al. Anomaly detection through information sharing under different topologies. EURASIP Journal on Information Security. (1):5. 2017.

Garera, S., et al. A Framework for Detection and Measurement of Phishing Attacks. In WORM '07: Proceedings of the 2007 ACM workshop on Recurring Malcode, 2007.

Gatt, A. and E. Krahmer. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. arXiv preprint arXiv:1703.09902. 2017.

Gaurav, M. and A.J, Mishra. Anti-Phishing Techniques: A Review. International Journal of Engineering Research and Applications (IJERA) Vol. 2, Issue 2, 2012.

Gehring, J., et al. Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 1243-1252). August 2017.

Ghazvininejad, M., et al. A Knowledge-Grounded Neural Conversation Model. arXiv preprint arXiv:1702.01932. 2017.

Grabocka, J., et al. Learning time-series shapelets. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 392-401). ACM. August 2014.

Grainger, T., et al. The Semantic Knowledge Graph: A compact, auto-generated model for real-time traversal and ranking of any relationship within a domain. In Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on. 2016.

Grau, S., et al. Dialogue act classification using a Bayesian approach. In 9th Conference Speech and Computer. 2004.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

gRPC. gRPC – A high-performance, open source universal RPC framework. Retrieved from https://grpc.io/. 2020.

Guha, S., et al. Robust random cut forest based anomaly detection on streams. In International conference on machine learning (pp. 2712-2721). June 2016.

Guthrie, D. Unsupervised Detection of Anomalous Text. Doctoral dissertation, University of Sheffield. 2008.

Halawa, H., et al. Harvesting the Low-hanging Fruits: Defending Against Automated Large-Scale Cyber-Intrusions by Focusing on the Vulnerable Population. New Security Paradigms Workshop (NSPW). ACM, 2016.

Hiraoka, T., et al. Active Learning for Example-based Dialog Systems. In Jokinen K., Wilcock G. (eds) Dialogues with Social Robots. Lecture Notes in Electrical Engineering, vol 427. Springer, Singapore. 2017.

Holtzman, et al. The curious case of neural text degeneration. Eighth International Conference for Learning Representations. 2020.

Hong J. The State of Phishing Attacks. Communications of the ACM, Vol. 55 No. 1, 2012.

Hu, Z., et al. Toward controlled generation of text. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 1587-1596). JMLR. org. August 2017.

Huang, H., et al. Network traffic anomaly detection. arXiv preprint arXiv:1402.0856. 2014.

Huggingface. huggingface/transfer-learning-conv-ai. Retrieved from https://github.com/huggingface/transfer-learning-conv-ai. 2020.

IETF. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. Retrieved from https://tools.ietf.org/html/rfc2046. 1996.

Inbucket. Inbucket disposable webmail. Retrieved from https://www.inbucket.org/. 2018.

Jurafsky, D., et al. Switchboard Dialog Act Corpus. Retrieved from https://web.stanford.edu/~jurafsky/ws97/. 1997.

Karim, F., et al. Multivariate LSTM-FCNS for time series classification. Neural Networks , 116, 237-245. 2019.

Khanpour, H., et al. Dialogue act classification in domain-independent conversations using a deep recurrent neural network. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (pp. 2012-2021). 2016.

Klimt, B., & Yang, Y. Introducing the Enron corpus. In CEAS. July 2004.

Kumaraguru, P., et al. Getting Users to Pay Attention to Anti-Phishing Education: Evaluation of Retention and Transfer. APWG eCrime Researchers Summit, 2007.

Kumaraguru, P., et al. Teaching Johnny not to fall for phish. ACM Trans. Internet Technol. 10, 2, Article 7, 2010.

Lample, G., et al. Multiple-attribute text rewriting. 2018.

Lewis, M., et al. Deal or no deal? End-to-end learning for negotiation dialogues. arXiv preprint arXiv:1706.05125. 2017.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Li, J., et al. A diversity-promoting objective function for neural conversation models.In Proceedings of NAACL 2016. 2016a.

Li, J., et al. A Persona-Based Neural Conversation Model. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 2016. 2016b.

Li, J., et al. Deep Reinforcement Learning for Dialogue Generation. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, 2016. 2016c.

Li, J., et al. Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547. 2017.

Li, J., et al. Delete, retrieve, generate: A simple approach to sentiment and style transfer. arXiv preprint arXiv:1804.06437. 2018.

Li, X., et al. A User Simulator for Task-Completion Dialogues. arXiv preprint, arXiv:1612.05688v3 [cs.LG]. 2017.

Li, X., et al. End-to-End Task-Completion Neural Dialogue Systems. arXiv preprint, arXiv:1703.01008v3 [cs.CL]. 2017.

Li, Y., et al. Dailydialog: A manually labelled multi-turn dialogue dataset. arXiv preprint arXiv:1710.03957. 2017 Oct 11.

Liu, C. W., et al. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023. 2016.

Malhotra, P., et al. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148. 2016.

Medvet, E. and E. Kirda. Visual-Similarity-Based Phishing Detection. Proceedings of the 4th International Conference on Security and Privacy in Communication Networks. 2008.

MITRE Corporation, The. About CVE. Retrieved from https://cve.mitre.org/about/. 2020.

OpenNMT. OpenNMT - Open-Source Neural Machine Translation. Retrieved from https://opennmt.net/.

Pan, Y. and X. Ding. Anomaly Based Web Phishing Page Detection. 22nd Annual Computer Security Applications Conference. 2006.

PolyAI-LDN. PolyAI-LDN/conversational-datasets. Retrieved from https:// github.com/PolyAI-LDN/conversational-datasets/tree/master/reddit. 2019.

PostgreSQL Global Development Group, The. PostgreSQL: Documentation: 12: F.31. pg_trgm. Retrieved from https://www.postgresql.org/docs/current/pgtrgm.html. 2020.

Potts, C. The Switchboard Dialog Act Corpus. Retrieved from http:// compprag.christopherpotts.net/swda.html. 2011.

Radev, D. CLAIR collection of fraud email, ACL Data and Code Repository, ADCR2008T001, http://aclweb.org/aclwiki. 2008.

Radford, A., et al. Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/ languageunsupervised/language understanding paper. pdf. 2018.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Radford, A., et al. Language models are unsupervised multitask learners. OpenAI Blog , 1 (8). 2019.

Raffel, C., et al. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683. 2019.

Rajeswar, S., et al. Adversarial Generation of Natural Language. arXiv preprint arXiv: 1705.10929. 2017.

Ratner, A., et al: Snorkel: Rapid Training Data Creation with Weak Supervision. Proceedings of the VLDB Endowment, 11(3), 269-282, 2017.

Roberts, A. and C. Raffel. Exploring Transfer Learning with T5: the Text-To-Text Transfer Transformer. Google AI Blog. Retrieved from https://ai.googleblog.com/ 2020/02/exploring-transfer-learning-with-t5.html. February 24, 2020.

Scholtz, J., Beyond Usability: Evaluation Aspects of Visual Analytic Environments, IEEE Visual Analytics Science and Technology, 2006.

Schulz, H., et al. A Frame Tracking Model for Memory-Enhanced Dialogue Systems. arXiv preprint, arXiv:1706.01690v1 [cs.CL]. 2017.

See, A., et al. What makes a good conversation? How controllable attributes affect human judgements. In Proceedings on NAACL 2019.

Seymour, J., Tully P. Weaponizing Data science for Social Engineering: Automated E2E spear phishing on Twitter. Black Hat USA. 2016.

Shah, P., et al. Interactive Reinforcement Learning for Task-oriented Dialogue Management. Deep Learning for Action and Interaction Workshop, NIPS 2016.

Shah, P., et al. Building a conversational agent overnight with dialogue self-play. arXiv preprint arXiv:1801.04871. 2018

Shao, L., et al. Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models. arXiv preprint arXiv:1701.03185v2 [cs.CL]. 2017.

Shao, L., et al. Generating Long and Diverse Responses with Neural Conversation Models. arXiv preprint arXiv:1701.03185. 2017.

Shen, T., et al. Style transfer from non-parallel text by cross-alignment. In Advances in neural information processing systems (pp. 6830-6841). 2017.

Sheng, S., et al. An Empirical Analysis of Phishing Blacklists. CEAS 2009 - Sixth Conference on Email and Anti-Spam. 2009.

Song, K., et al. Mass: Masked sequence to sequence pre-training for language generation. arXiv preprint arXiv:1905.02450. 2019.

Sordoni, A., et al. A Neural Network Approach to Context-sensitive Generation of Conversational Responses. arXiv preprint arXiv:1506.06714. 2015.

Sriram, A., et al. Cold fusion: Training seq2seq models together with language models. arXiv preprint arXiv:1708.06426. 2017.

Stahlberg, F., et al. Simple fusion: Return of the language model. arXiv preprint arXiv:1809.00125. 2018.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Su, P., et al. Learning from Real Users: Rating Dialogue Success with Neural Networks for Reinforcement Learning in Spoken Dialogue Systems. In Interspeech, Dresden, Germany. ISCA. 2015.

Sugany, V. A Review on Phishing Attacks and Various Anti Phishing Techniques. International Journal of Computer Applications (0975 – 8887) Volume 139 – No.1, 2016.

Sutskever, I., et al. Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112). 2014.

Tatman, R. Fraudulent E-mail Corpus | Kaggle. Retrieved from https:// www.kaggle.com/rtatman/fraudulent-email-corpus. 2017.

Tauberer, J. Mail-in-a-Box. Retrieved from https://mailinabox.email/. 2020.

Thomas, K., et al. Data Breaches, Phishing, or Malware?: Understanding the Risks of Stolen Credentials. ACM SIGSAC Conference, 2017.

Thought Vector. Customer Support on Twitter | Kaggle. Retrieved from https:// www.kaggle.com/thoughtvector/customer-support-on-twitter. 2017.

Vaswani, A., et al. Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008). 2017.

Waddel, K. The Twitter Bot That Sounded Just Like Me. The Atlantic. 2016. https://www.theatlantic.com/technology/archive/2016/08/the-twitter-bot-that-sounds- just-like-me/496340/.

Wen, T.H., et al. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. arXiv preprint arXiv:1508.01745. 2015.

Wen, T. H., et al. Latent intention dialogue models. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 3732-3741). 2017.

Wenyin, L., et al. Detection of Phishing Webpages based on Visual Similarity. 14th International Conference on the World Wide Web. 2005.

Whittaker, C., et al. Large-Scale Automatic Classification of Phishing Pages. Proceedings of the Network and Distributed System Security Symposium, NDSS 2010, San Diego, California, USA, 2010.

Wikipedia contributors. Beam search. In Wikipedia, The Free Encyclopedia. Retrieved 15:13, April 6, 2020, from https://en.wikipedia.org/w/index.php? title=Beam_search&oldid=942391463. 2020.

Williams, J., et al. Rapidly scaling Dialog Systems with Interactive Learning. In Proceedings of IWSDS. 2015.

Wolf, T., et al. TransferTransfo: A transfer learning approach for neural network based conversational agents. arXiv preprint arXiv:1901.08149. 2019.

Wright, W. and T. Kapler. Visualization of Blue Forces Using Blobology, 2002 International Command and Control Research and Technology Symposium, June 2002.

Wu, M., et al. Do Security Toolbars Actually Prevent Phishing Attacks? ACM Press, CHI 2006, 22-27 April 2006.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Wurstmeister. wurstmeister/kafka-docker: Dockerfile for Apache Kafka. Retrieved from https://github.com/wurstmeister/kafka-docker, 2020

Yu, J., & Jiang, J. Learning sentence embeddings with auxiliary tasks for cross- domain sentiment classification. Association for Computational Linguistics. 2016.

Yu, Z., et al. Learning Conversational Systems that Interleave Task and Non-Task Content. arXiv preprint, arXiv:1703.00099v1 [cs.CL]. 2017.

Zhao, J., et al. # FluxFlow: Visual analysis of anomalous information spreading on social media. IEEE Transactions on Visualization and Computer Graphics. 20(12). 2014.

Zhao, T. and M. Eskenazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. arXiv preprint arXiv:1606.02560. 2016.

Zhou, C., et al. A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630. 2015.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

8.0 LIST OF ABBREVIATIONS, ACRONYMS AND SYMBOLS

API	Application Programming Interface
ASED	Active Social Engineering Defense
AUC	Area Under the Curve
BBN	Bolt Beranek and Newman Inc.
BLEU	Bilingual Evaluation Understudy
CHESS	Continuously Habituating Elicitation Strategies for Social-Engineering-Attacks
CNN	Convolutional Neural Network
CRUD	Create, Read, Update and Delete
CVE	Common Vulnerabilities and Exposures
DARPA	Defense Advanced Research Projects Agency
EML	Electronic Email
FN	False Negative
FP	False Positive
GP	Gated Product
GPT	Generative Pre-Trained
gRPC	General-purpose Remote Procedure Call
GRU	Gated Recurrent Unit
HITL	Human-in-the-Loop
HRL	Hughes Research Laboratories
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
IP	Internet Protocol
JPEG	Joint Photographic Experts Group
JPL	Jet Propulsion Laboratory
KYC	Know Your Customer
LSTM	Long Short-Term Memory
LSTM-FCN	Long Short-Term Memory Fully Convolutional Networks
MIME	Multipurpose Internet Mail Extensions
ML	Machine Learning

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

NASA	National Aeronautics and Space Administration
NLP	Natural Language Processing
NER	Named-Entity Recognition
NMT	Neural Machine Translation
NS	Naive Sum
PII	Personally Identifiable Information
PMP	Persona Management Platform
POS	Part-of-Speech
QCR	Quantitative Crisis Response
REST	Representational State Transfer
RNN	Recurrent Neural Net
ROC	Receiver Operating Characteristic
RRCF	Random Robust-Cut Forest
SA	Situational Awareness
SemaFor	Semantic Forensics
seq2seq	Sequence-to-sequence
SIENNA	Strategies for Investigating and Eliciting Information from Nuanced Attackers
SIMON	Semantic Interface for the Modeling of Ontologies
SME	Subject Matter Expert
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SOTA	State of the Art
STIX	Structured Threat Information Expression
SWDA	Switchboard Dialogue Act Corpus
TA	Technical Area
TBVA	Tiler-Based-Visual Analytics
TN	True Negative
TP	True Positive
TRSS	Thomson Reuters Special Services
UI	User Interface
URL	Uniform Resource Locator

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

VM	Virtual Machine
WS	Weighted Sum
YAML	Yami Aint’t Markup Language

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

APPENDIX A - Investigating Language Model Fusion Methods for Open-Domain Dialogue Systems

Investigating Language Model Fusion Methods for Open-Domain Dialogue Systems

Anonymous ACL submission

Abstract

Various fusion techniques combining pre-trained language models with generative sequence-to-sequence models have been explored in the domains of speech and translation systems to great effect. However, application of these methods to open-domain dialogue systems remains an under-explored area. In this work we evaluate several fusion techniques for incorporating language model input into sequence-to-sequence dialogue models during training. We find that training dialogue models with fusion training regimes boosts overall performance on open-domain tasks compared to a sequence-to-sequence baseline and aids in optimizing models more quickly, potentially decreasing training time. Additionally, we present an analysis of decoder weights using Procrustes and projection-weighted canonical correlation analysis (PWCCA), demonstrating that simple language model fusion techniques can be used effectively in dialogue modeling without over-reliance on the pre-trained LM input.

1 Introduction

Dialogue modeling has shown remarkable progress following the use of neural sequence-to-sequence (Seq2Seq) modeling techniques (Sutskever et al., 2014). However, training these models to generate realistic responses often requires large amounts of input-response pairs and a great deal of compute resources and training time. Most recent work in improving model training has focused on exploring new neural architectures or training regimes that augment the traditional supervised learning paradigm. While these techniques have yielded interesting and impressive results, they still require large amounts of pair-wise data that need to be collected, potentially through ad-hoc dialogue data collection involving human participants. We seek to leverage the large amount of unlabeled text data available via pre-trained language models to help alleviate this data bottleneck. Additionally, we aim to reduce training time of the final dialogue systems by combining or "fusing" information from the language models into dialogue models as they train.

In this work, we borrow a simple idea that has been used effectively in neural machine translation (NMT) and automatic speech recognition (ASR) that leverages pre-trained language models (LMs). Specifically, we evaluate several techniques for incorporating LMs into the training regime of neural Seq2Seq dialogue models (DMs) trained on an open-domain dialogue task. Following ideas such as "simple fusion" as proposed in (Stahlberg et al., 2018), we evaluate three techniques for combining LM and DM outputs during the DM training process, keeping the LM weights fixed and allowing DM weights to be updated. We investigate three variants of these fusion methods and compare each against a standard Seq2Seq benchmark. We find that all fusion techniques achieve lower perplexity and higher matching- and embedding-based evaluation scores, while encouraging model convergence and allowing for faster training times.

2 Related Work

Most previous work in fusing LM's and Seq2Seq models has primarily been conducted in the areas of automatic speech recognition (ASR), and neural machine translation (NMT), which further motivates this work's application of these techniques to dialogue modeling. We give an overview of this work below.

Fusion Work in Automatic Speech Recognition

While this current work concerns incorporating LMs into the training regime of text-only Seq2Seq architectures, it is worth summarizing previous work in automatic speech recognition (ASR) since

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

this domain has most thoroughly explored fusing LMs with Seq2Seq models. In (Gulcehre et al., 2017) the authors propose a log-linear interpolation between outputs from the speech-to-text decoder and an independently-trained n-gram LM and find limited improvements, but note that incorporating a RNN-LM may further boost performance. (Kannan et al., 2018) likewise investigate incorporating LMs with ASR Seq2Seq models via "shallow fusion" log-linear interpolation and find in some cases a 9.1% relative decrease in word-error-rate.

Other works, such as (Cho et al., 2019) investigate more sophisticated techniques for incorporating LMs into ASR model training beyond shallow fusion by fusing hidden states from a pre-trained LM with those of the ASR model during training, and demonstrate improvements in word- and character-error rates. The Cold Fusion technique (Sriram et al., 2017) proposes fusing hidden states between the LM and ASR decoder using feed-forward networks, and shows boosts in model convergence and perplexity. Similarly, (Shan et al., 2019) propose a technique they term "component fusion" where an external LM is trained on speech transcriptions and the gated outputs of this LM are concatenated with the hidden ASR decoder representations at each timestep during training of the ASR model. The authors show improvements in character-error-rates over using the Cold Fusion and other "shallow fusion" methods. Finally, (Liu et al., 2019) propose using a pre-trained LM in an adversarial training regime where the LM assigns scores to the decoded output from the end-to-end ASR model, and the ASR model is trained to maximize this LM score as well as minimize the negative log-likelihood objective for the decoder of the ASR Seq2Seq model.

Fusion Work in Neural Machine Translation

Our work most closely resembles (Stahlberg et al., 2018) who investigate techniques for fusing LMs and Seq2Seq models in the context of neural machine translation (NMT). In it the authors build on the work in (Sriram et al., 2017) and apply a simple weighted-sum fusion approach to several datasets in NMT, demonstrating this approach's effectiveness over more sophisticated fusion techniques in the ASR literature. Other work, such as the multi-task regime in (Domhan and Hieber, 2017), impose a separate target-only language modeling task on the decoder in addition to the standard Seq2Seq objective. In this scenario, the decoder must generate a response both from (a) the encoded input-language context and (b) only the previous context in the target-side data. While (Zhou et al., 2017) do not focus on utilizing LMs for fusion training, the authors still demonstrate the effectiveness of fusing representations from multiple translation systems to improve overall translation quality.

3 Methods

Language Model Pre-Training

As a first step, prior to training any fused dialogue models (DM), we train a language model (LM) on the same training corpus and evaluate on the same validation set. We train the model using the standard language modeling objective, where we seek to maximize the log-likelihood of predicting the next word in a sentence given the previous words. The model is trained to assign high log-probabilities to likely sequences as in equation 1.

$$P_{LM}(\mathbf{x}) = \sum_{t=1}^{|\mathbf{x}|} \log P_{LM}(x_t \mid x_{<t}) \qquad (1)$$

Recently, neural networks have been used to estimate this distribution over the training vocabulary, with a number of different architectures being available for solving this problem (Bengio et al., 2003; Mikolov et al., 2010). For the architecture of the LM in this work, we choose the transformer-based generative pre-training (GPT) architecture introduced in (Radford et al., 2018). We ran initial experiments with an LSTM-based LM and found all results to be better using the GPT model, and only report final dialogue results using this model fused with DMs. Further, we do not explore the use of bidirectional models such as BERT (Devlin et al., 2018) due to its being trained on a slot-filling objective as opposed to a true language modeling objective.[1] Hyper-parameters for the GPT-LM are summarized in Table 1 below.

Seq2Seq Training

We compare all models using fusion techniques to a standard Seq2Seq baseline trained without any input from a LM. As in previous work, we train these models to generate the appropriate response given an encoded input context and the ground-truth response token at each timestep (e.g. teacher-forcing).

[1] We note that architectures such as XL-NET allow bidirectional transformer models to be trained on LM objectives, and leave these investigations to future work.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

GPT-LM Architecture Hyper-parameters	
Vocab size (BPE)	40,481
Number Self-Att. Heads	8
Number Self-Att. Layers	5
Embedding dimension	512
Hidden feed-forward units	512

Table 1: Hyperparameters for GPT-LM pre-training.

That is, we seek a model that will assign high log-probability scores to appropriate responses given a certain context as in equation 2 below.

$$P_{DM}(\mathbf{y}) = \sum_{t=1}^{|\mathbf{x}|} \log P_{DM}(y_t \mid y_{<t}, \mathbf{x}) \quad (2)$$

We can think of the decoder of this model as a conditional language model that is conditioned not only on previous response tokens, but also on a learned representation of the context input $\mathbf{x}$. For this work, we use LSTMs for the encoder and decoder of all DMs.

LM - Seq2Seq Fusion

Inspired by work in neural machine translation and automatic speech recognition as in (Sriram et al., 2017) and (Stahlberg et al., 2018), we investigate several fusion methods for combining outputs from a pre-trained LM whose weights are frozen with outputs from a DM. In particular we focus exclusively on techniques for combining either probabilities or log-probabilities from both models while all weights of the DM are updated during training. We hypothesize that keeping the weights from the LM frozen during DM training will encourage faster convergence while also providing a regularizing effect. We provide further details regarding these training regimes below.

4 Experimental Setup

Dataset

We train and evaluate our models on the Cornell Movie Dialogue Corpus (Danescu-Niculescu-Mizil and Lee, 2011) which contains over 220,000 open-domain conversational exchanges from 617 movie scripts. We divide the corpus into a training and evaluation/test set, with 199,229 dialogue pairs used for training and 22,137 used for a held-out test set. Further, we ensure that training and test sets are split by conversation such that no dialogue-pairs from the same conversation appear in both the training and test set.

Training Regimes

We explore three alternative fusion techniques in our experiments. Each alternative operates on the log-probabilities output by both the pre-trained LM and the DM, and we compare these techniques to a standard LSTM-based Seq2Seq baseline without any input from an independently-trained LM.

Naive Sum In this technique we simply take the log-probabilities at each timestep from both the pre-trained LM and the DM and compute the element-wise sum over each word in the vocabulary for each timestep. These summed log-probabilities are then passed through a softmax function before computing loss and updating model weights via backpropagation.

$$\hat{\mathbf{y}} = \log P_{DM}(\mathbf{y} \mid \mathbf{x}) + \log P_{LM}(\mathbf{y}) \quad (3)$$

Weighted Sum This training regime is identical to the Naive Sum except that we introduce a tuning parameter λ that is randomly initialized to a value between 0 and 1 to control how much influence the LM log-probabilities contribute to overall predictions. The intuition is that higher values of λ indicate more influence from the LM during training. Further, we allow λ to be tuned dynamically during training instead of choosing a static value prior to starting training.

$$\hat{\mathbf{y}} = \log P_{DM}(\mathbf{y} \mid \mathbf{x}) + \lambda \log P_{LM}(\mathbf{y}) \quad (4)$$

Gated Product Here, we investigate a technique that introduces a gating function applied to the probabilities output by the pre-trained LM prior to combining with the probabilities output by the dialogue model. We ensure that both the log-probabilities from the LM and the DM are fed through the softmax function to ensure valid probability distributions over the vocabulary for each timestep. We then combine the outputs as in Equation 5 below.

$$\hat{\mathbf{y}} = P_{DM}(\mathbf{y} \mid \mathbf{x}) * \sigma(P_{LM}(\mathbf{y})) \quad (5)$$

We train all of the above variants, as well as the baseline DM, with 400-dimensional sub-word (Sennrich et al., 2015) embedding vectors learned from

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

scratch, 2 bidirectional LSTM encoder layers with 768 units in each direction, and two unidirectional LSTM decoder layers with 1536 units. Input sub-word tokens are computed using the learned sub-words from (Radford et al., 2018) for all models with a vocabulary size of 40,481 sub-words, and a maximum input length of 80 tokens for input sentences. We train the baseline and all fused DMs with the Adagrad optimization algorithm (Duchi et al., 2011) with an initial learning rate of 0.01 and anneal this learning rate by 0.7 after every 10 epochs, with a total of 50 epochs of training for each model and a batch-size of 128. All models are trained on a single Tesla V100 card.

Evaluation

In addition to measuring perplexity, we utilize several unsupervised evaluation metrics outlined in (Sharma et al., 2017) to gain insight into DM performance. [2] While previous work such as (Liu et al., 2016) highlights weak correlation between some unsupervised metrics and human ratings, incorporating several of these metrics in concert can still help us gain multiple views on model performance and comparison.

We use BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and ROUGE-L (Lin, 2004) as word-overlap metrics to capture several different aspects of performance in reproducing appropriate responses at a surface level. BLEU provides the simplest measure by computing n-gram overlap between candidate and reference sentences (effectively capturing n-gram precision), while METEOR measures the harmonic mean between precision and recall between candidates and references, and incorporates more sophisticated text processing techniques such as stemming and WordNet synonym look-ups. Finally, ROUGE-L computes an F-measure based on the longest common sub-sequence between candidates and references.

Similarly, we employ three embedding-based metrics to capture semantic aspects of system responses. We use two methods that compute sentence embeddings by combining individual word embeddings (Wieting et al., 2015; Forgues et al., 2014) (word embedding average, word embedding extrema), and SKIPTHOUGHT sentence embeddings (Kiros et al., 2015) to measure system performance. We follow previous work in computing these representations for both candidate and reference sentences, and then measuring cosine similarity between the resulting vectors. Intuitively, higher cosine similarity shows greater semantic similarity between references and candidates. This allows for high scores for candidates that may have low word-overlap with the reference, but are still semantically very similar. [3]

[2] We also utilize the open-sourced code from this work to compute these metrics, available at: https://github.com/Maluuba/nlg-eval

5 Results & Discussion

Quantitative Comparisons

Our main quantitative results are shown in Table 2. For all quantitative results, we use beam search as the decoding method with number of beams set to 4. Overall, results show that fusion techniques outperform the baseline with respect to all metrics tested. In terms of perplexity, the WEIGHTED SUM method performs best, with an absolute decrease of 8.8 points. Interestingly, however, we see very little difference between each of the fusion methods in terms of perplexity. This may indicate that there is little difference between these methods after sufficient training time.

All matching-based metrics show low performance, however all fusion methods outperform the baseline except for the GATED PRODUCT model, which suffers in terms of BLEU scores. Low overlap-metric scores are not surprising given that in the case of dialogue modeling, a model may output an entirely appropriate response that shows no overlap with a reference output sentence as outlined in (Gupta et al., 2019). Our main interest here is comparison between the baseline model and the outlined fusion techniques.

	BASELINE	NS	WS	GP
Perplexity	50.4	42.4	**41.6**	42.1
BLEU-2	0.011	0.025	**0.031**	0.003
METEOR	0.018	0.032	**0.033**	0.027
ROUGE-L	0.016	**0.031**	0.029	0.018
SKIPTHOUGHT	0.36	**0.60**	0.58	0.59
EMB AVG	0.80	**0.83**	0.81	**0.83**
EMB EXTREMA	0.44	0.50	**0.53**	0.52

Table 2: Quantitative results comparing baseline DM with proposed fusion training regimes, best scores in bold.

The WEIGHTED SUM model shows best performance over most metrics reported here, and

[3] E.g. the candidate sentence *The book was horrible.* and reference *That novel is awful.* show no word-overlap, though they are semantically extremely similar.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

falls only narrowly behind in most of the metrics where it does not perform best. Differences in performance in terms of the embedding metrics are especially interesting. All models, including the baseline, perform very similarly in terms of embedding average similarity, while the skip-thought embedding similarity shows the greatest difference between baseline and fusion models. This may be the result of skip-thought vectors being more sensitive to sentence structure due to the embedding model's objective of reconstructing adjacent sentences, given a target sentence. Conversely, the embedding average scores are remarkably similar for all models and may indicate a lack of syntactic sensitivity in evaluating model responses.

To gain better insight into dynamics of these different training regimes, we plot the loss of each model over the first 100 batch updates during training in Figure 1. In this very early training stage we see that each fusion technique appears to dampen severe jumps in loss compared to training the baseline model. Interestingly, though it does not achieve the best perplexity score on the test set, we also see the GATED PRODUCT training method appears to have the greatest effect in smoothing out these noisy updates during training, which may indicate an advantage in terms of training time over the other methods. Overall, these curves suggest a strong regularizing effect due to incorporating the language model in addition to the much faster convergence to a lower loss. As can be seen in the figure, each fusion model quickly converges to a loss well below that achieved by the baseline model, and this trend holds when looking at loss over further training batch updates.

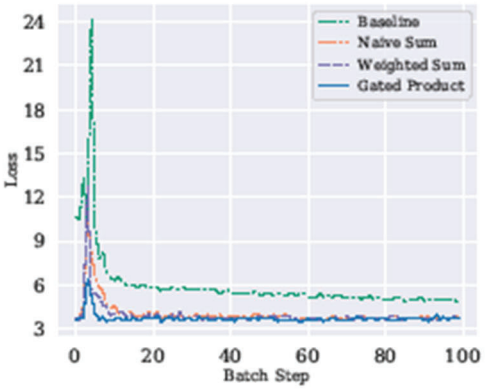


Figure 1: Convergence patterns of different training regimes over first 100 batch updates.

Qualitative Comparisons

In Table 3 we summarize several responses from examples sampled from the test set used to compute metrics in the quantitative results above. We focus on examples that appear to explicitly elicit a response in a conversational context (e.g. posing a question or requesting information from a conversation participant), as well as more subtle examples that only hint at the request for a response. We fix the decoding method for all models to isolate the effects of the training regimes and use top-k decoding with the value of k set to 10. Top-k sampling is a common approach for introducing greater diversity into the generated responses from dialogue models. It can be thought of introducing "risk" in the decoding step (e.g. sampling words that have a lower probability than the maximum at a given timestep) to generate more diverse or interesting words in the model response.

While it is difficult to draw general conclusions from these qualitative results, there are several patterns that emerge. First, we note that of all the responses shown in Table 3, only the baseline model introduces grammatical errors as in the response to Example 2 (the ungrammatical prepositional phrase *of your business* in *What the hell's the matter* ***of your business****?*) and the response to Example 3 (introducing the non-existent word ***clouman***). Further, we note concrete evidence of greater relevance in the responses from some of the fusion methods compared to the baseline. For example, in Example 2 the WEIGHTED SUM model response (*I'm sorry, but I don't want to see him.*) contains the pronoun *him*, referencing a person, *Basil*, mentioned in the input sentence, while the baseline model's response is less relevant and contains grammatical errors discussed above. Additionally, in Example 5, we see the GATED PRODUCT model output the word *trying* in reference to the input text *Try again next year*, while the baseline outputs a vague response (*What?*) that could potentially fit a number of contexts, but isn't especially relevant.

Decoder Analysis

One concern with these fusion approaches is that the decoder of the DM being trained may rely too heavily on input from the pre-trained LM result-

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Context Ex. 1	How'd they take it back at headquarters?
BASELINE	I'm not a doctor.
NAIVE SUM	You've heard it before.
WEIGHTED SUM	I don't know... I think I'd better go.
GATED PRODUCT	You know what's wrong with me.
Context Ex. 2	Basil was very concerned to know where you were last night.
BASELINE	What the hell's the matter of your business?
NAIVE SUM	I'm sorry, but what do you mean?
WEIGHTED SUM	I'm sorry, but I don't want to see him.
GATED PRODUCT	You know what I've done to him. He'll be little too.
Context Ex. 3	There is something dreadful on your soul. Unburden it to me.
BASELINE	You don't have to be a hero. I'm not one. I'm not a clouman, I don't think so.
NAIVE SUM	Oh, well... I just thought I would never be in the way.
WEIGHTED SUM	You're not a celebrity, are you?
GATED PRODUCT	I've got a lot of money, but it doesn't make sense.
Context Ex. 4	You didn't come to talk about the weather, surely.
BASELINE	You don't know me.
NAIVE SUM	Well, you'd be good on that.
WEIGHTED SUM	That's not true, sir.
GATED PRODUCT	I don't know. I've been trying to get out of here, but I'm not going to get it.
Context Ex. 5	Sorry, but it's out of my hands. Try again next year.
BASELINE	What?
NAIVE SUM	I don't want to do that. You should be the one.
WEIGHTED SUM	It's not your fault.
GATED PRODUCT	You know, I've been trying for a while.

Table 3: Example output from various training regimes. Context inputs are drawn from test set.

ing in its weights not being sufficiently updated through the training process. To test for evidence of this, we employ traditional generalized Procrustes analysis (Gower, 1975) as well as a more recent projection weighted canonical correlation analysis (PWCCA) method described in (Morcos et al., 2018) to measure similarity between activations output by the decoders of each training regime. While Procrustes distance computes a standardized squared-sum of distances between elements in the two representations, PWCCA was developed for the express use-case of neural net model analysis, and measures similarities between representations while being invariant to linear transformations of these representations.

Specifically, we take decoder activations from each fusion training regime after the first and last epoch of training and measure the Procrustes distance and PWCCA similarity between them. The intuition is that higher PWCCA similarity (and lower Procrustes distance) reveals less change in the weights during the training process suggesting higher reliance on LM weights, while lower PWCCA similarity (higher Procrustes distance) reveals greater change in these weights and thus a more effective learned tradeoff between the decoder weights and the pre-trained LM weights.

	PWCCA	Proc. Dist.
BASELINE	0.53	0.71
NAIVE SUM	0.32	0.92
WEIGHTED SUM	0.35	0.88
GATED PRODUCT	0.57	0.77

Table 4: PWCCA coefficients (left) and Procrustes distance (right) comparing decoder activations after first and last training epochs.

We use the open-sourced code provided by (Morcos et al., 2018) to measure PWCCA similarity for all fusion training methods outlined above.[4] To

[4] https://github.com/google/svcca

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

analyze the fused model decoders, we obtain activations from the last decoder layer for *only* the DM over each sentence in the test set. This ensures we only measure changes in activations due to the weights being updated in the DM. Table 4 shows results of these analyses. Most surprisingly, we see that the BASELINE model shows second-highest PWCCA similarity and second-lowest Procrustes distance between first and last training epochs, indicating smaller weight updates during training compared to other methods. This runs counter to our intuition that the baseline model without influence from the pre-trained LM would show the greatest change in decoder weights over the course of training.

We also see that all fusion models show fairly low PWCCA coefficients, with lowest correlation seen in the two summation models. An identical pattern is seen when looking at Procrustes distance. This indicates that weights in the DM are being updated for each training regime, and the fused models don't rely entirely on the pre-trained LM for their predictive power. In fact, in comparison with the BASELINE model, this suggests that inclusion of the LM actually *encourages* change in decoder weights in the NAIVE SUM and WEIGHTED SUM methods. We further note high similarity between the two NAIVE SUM and WEIGHTED SUM methods, and a comparatively higher PWCCA coefficient for the GATED PRODUCT fusion technique compared to the two summation methods. This discrepancy suggests that both of the summation fusion methods allow for DM weights to be updated more effectively than the GATED PRODUCT regime, and this result also conforms to intuition given that the summation methods are conceptually very similar.

Additionally, we investigate how decoder activations are distributed across timesteps. Figure 2 shows the average neuron activation value for all methods over each timestep. All curves are obtained after the final epoch of training. Several interesting patterns emerge in this plot, which may indicate meaningful differences in learning dynamics between these fusion methods. Most obvious is the comparatively large range of activation values for the GATED PRODUCT method compared to other fusion methods. We see that the range in activation values is even larger than the baseline model, suggesting that this gating method may introduce volatility to the training process. Conversely, we see much more restricted activation values from both the NAIVE SUM and WEIGHTED SUM methods, suggesting a strong regularizing effect from incorporating the LM. These results, coupled with faster convergence and strong quantitative performance over the baseline model, demonstrate that the two summation fusion methods appear to effectively incorporate information from the LM while also allowing DM decoder weights to be updated during training. This regularization of hidden representations may be especially important to increased performance as noted in related language modeling work in (Merity et al., 2017).

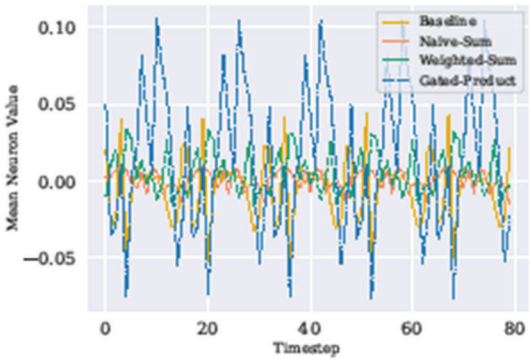


Figure 2: Mean neuron weight over each timestep after final epoch of training.

6 Conclusion

We have presented an analysis of several straightforward fusion methods for incorporating independently trained language models into training of Seq2Seq dialogue models. We have shown results that suggest these methods boost overall performance and encourage faster convergence than training a dialogue model on its own. Further, we have presented an analysis of the decoders of each fusion method to investigate the behavior of weight updates in each of these methods. These last results demonstrate that each of the fusion methods is able to effectively leverage pre-trained language models during the training phase without becoming wholly dependent on them. The inclusion of a language model in the training phase for dialogue models appears to introduce robust regularizing effects while also demonstrating the potential for reducing convergence- and overall training-time

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

for these networks.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Jaejin Cho, Shinji Watanabe, Takaaki Hori, Murali Karthick Baskar, Hirofumi Inaguma, Jesus Villalba, and Najim Dehak. 2019. Language model integration based on memory control for sequence to sequence speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6191–6195. IEEE.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In *Nips, modern machine learning and natural language processing workshop*, volume 2.

John C Gower. 1975. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Computer Speech & Language*, 45:137–148.

Prakhar Gupta, Shikib Mehri, Tiancheng Zhao, Amy Pavel, Maxine Eskénazi, and Jeffrey P. Bigham. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. *CoRR*, abs/1907.10568.

Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhijeng Chen, and Rohit Prabhavalkar. 2018. An analysis of incorporating an external language model into a sequence-to-sequence model. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5828. IEEE.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Alexander H Liu, Hung-yi Lee, and Lin-shan Lee. 2019. Adversarial training of end-to-end speech recognition using a criticizing language model. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6176–6180. IEEE.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Stephen Merity, Bryan McCann, and Richard Socher. 2017. Revisiting activation regularization for language rnns. *CoRR*, abs/1708.01009.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, pages 5727–5736.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Changhao Shan, Chao Weng, Guangsen Wang, Dan Su, Min Luo, Dong Yu, and Lei Xie. 2019. Component fusion: Learning replaceable language model component for end-to-end speech recognition system. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5361–5635. IEEE.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2017. Cold fusion: Training seq2seq models together with language models. *arXiv preprint arXiv:1708.06426*.

Felix Stahlberg, James Cross, and Veselin Stoyanov. 2018. Simple fusion: Return of the language model. *arXiv preprint arXiv:1809.00125*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Long Zhou, Wenpeng Hu, Jiajun Zhang, and Chengqing Zong. 2017. Neural system combination for machine translation. *arXiv preprint arXiv:1704.06393*.

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

ACRONYMS / GLOSSARY

API	Application Programming Interface
AUC	Area Under the Curve
BBN	Bolt Beranek and Newman Inc.
BLEU	Bilingual Evaluation Understudy
CHESS	Continuously Habituating Elicitation Strategies for Social-engineering-Attacks
CNN	Convolutional Neural Network
CRUD	Create, Read, Update and Delete
CVE	Common Vulnerabilities and Exposures
GP	Gated Product
gRPC	General-purpose Remote Procedure Call
GRU	Gated Recurrent Unit
HITL	Human-in-the-loop
HRL	Hughes Research Laboratories
JPL	Jet Propulsion Laboratory
KYC	Know Your Customer
LSTM	Long Short-Term Memory
LSTM-FCN	Long Short-Term Memory Fully Convolutional Networks
ML	Machine Learning
NLP	Natural Language Processing
NER	Named-Entity Recognition
NMT	Neural Machine Translation
NS	Naive Sum
PII	Personally Identifiable Information
PMP	Persona Management Platform
QPR	Quarterly Program Review
REST	Representational State Transfer
RNN	Recurrent Neural Net
ROC	Receiver Operating Characteristic
RRCF	Random Robust-Cut Forest Sequence-to-Sequence

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021

SIENNA	Strategies for Investigating and Eliciting Information from Nuanced Attackers
SIMON	Semantic Interface for the Modeling of Ontologies
SME	Subject Matter Expert
SOTA	State of the Art
STIX	Structured Threat Information Expression
SWDA	Switchboard Dialogue Act Corpus
TBVA	Tiler-Based-Visual Analytics
TRSS	Thomson Reuters Special Services
UI	User Interface
VM	Virtual Machine
WS	Weighted Sum

Distribution A. Approved for public release; distribution unlimited.
AFRL-2021-0469; Cleared 19 Feb 2021