Detecting Asks in Social Engineering Attacks: Impact of Linguistic and Structural Knowledge

Bonnie J. Dorr,[1] Archna Bhatia,[1] Adam Dalton,[1] Brodie Mather,[1]
Bryanna Hebenstreit,[2] Sashank Santhanam,[3] Zhuo Cheng,[3]
Samira Shaikh,[3] Alan Zemel,[2] Tomek Strzalkowski[4]
[1]Institute for Human and Machine Cognition (IHMC), Ocala, FL, USA, {bdorr, abhatia, adalton, bmather}@ihmc.us
[2]State University of New York, Albany, NY, USA, {bhebenstreit, azemel}@albany.edu
[3]University of North Carolina, Charlotte, NC, USA, {sshaikh2, ssantha1, zcheng5}@uncc.edu
[4]Rensselaer Polytechnic Institute, Troy, NY, USA, tomek@rpi.edu

Abstract

Social engineers attempt to manipulate users into undertaking actions such as downloading malware by clicking links or providing access to money or sensitive information. Natural language processing, computational sociolinguistics, and media-specific structural clues provide a means for detecting both the *ask* (e.g., *buy gift card*) and the risk/reward implied by the ask, which we call *framing* (e.g., *lose your job*, *get a raise*). We apply linguistic resources such as *Lexical Conceptual Structure* to tackle ask detection and also leverage structural clues such as links and their proximity to identified asks to improve confidence in our results. Our experiments indicate that the performance of ask detection, framing detection, and identification of the *top ask* is improved by linguistically motivated classes coupled with structural clues such as links. Our approach is implemented in a system that informs users about social engineering risk situations.

1 Introduction

Social engineering (SE) attacks are a significant cybersecurity threat, putting individuals and organizations at risk. Social engineers attempt to manipulate users into undertaking actions such as downloading malware by clicking links (PERFORM) or providing access to money or sensitive information (GIVE). Such eliciting behaviors, or *asks*, are not always explicitly stated in text (Drew and Couper-Kuhlen 2014a), but may be gleaned through automatic techniques that employ both linguistic and structural knowledge. Natural language processing (NLP), computational sociolinguistics, and media-specific structural clues provide a means for detecting both the *ask* (e.g., *buy gift card*) and the risk/reward (or LOSE/GAIN) implied by the ask, which we call *framing* (e.g., *lose your job*, *get a raise*). These elements can be used in downstream operations for countering attacks, e.g., through bot-produced responses and actions.

Interest in conversational intelligence has surged in recent years as evidenced by the ConvAI2 NeurIPS competition (Dinan et al. 2019; Yusupov and Kuratov 2018) and related efforts in natural language dialogue (Perera et al. 2017). We build on prior Cyber-related dialogue work (Dalton et al. 2019), focusing on critical conversational objectives beyond NLP-based cyber-attack *detection* and *prediction* (Dalton et al. 2017; Perera et al. 2018; Hollingshead et al. 2019) or NLP-based *generation* of warnings and the explanations behind them (Kazakova et al. 2019). We implement a novel ask detection approach with the ultimate goal of sustaining a believable *conversation* with a social engineer to extract their true identity without putting valuable resources at risk.

Toward that end, we apply linguistic resources, including *Lexical Conceptual Structure* (e.g., relating verbs such as *give*, *donate*) (Dorr and Olsen 2018; Dorr and Voss 2018) and *categorial variation* (e.g., relating word variants such as *reference*, *refer*) (Habash and Dorr 2003). We also leverage structural clues such as links (URLs, email addresses) and their proximity to identified asks. Although the email channel is our starting point, we adopt such techniques with an eye toward extensibility to channels beyond email (e.g., texting), but at the same time we identify techniques specific to email, such as email signature and quoted reply removal. No training data are needed for the approach described herein.

Table 1 shows representative system output produced by our approach for four (presumed) SE emails. *Framing* output sets the stage for the ask, i.e., the purported threat (LOSE) or benefit (GAIN) that the social engineer wants the potential victim to believe is dependent on compliance or lack thereof. The output shows three PERFORM asks and one GIVE ask. Information in parentheses () refers to one or more links that the potential victim might choose to click.[1] Links are detectable from html mark-up whether or not their textual form is used in the email body. Information in brackets [] refers to the ask/framing category of interest to the social engineer (e.g., finance_money, personal, credentials).[2]

We describe multiple experiments to determine the impact of linguistic and structural knowledge on ask/framing detection. Two key findings are: (1) Both linguistically motivated classes (e.g., *Judge, Want, Send)* and verbal processing (e.g., filtering on the basis of part of speech) improve precision

[1]For space reasons, abbreviated forms "jw11@..." and "http..." are used in the table.

[2]Experiments reported herein do not focus on such categories, but these are shown for illustrational purposes. (Future work will investigate the impact of such categories on response generation.)

Email	Framing	Ask	Conf
I am stuck at the airport. Please help me out by sending $500.	LOSE stuck() []	PERFORM help() [finance_money]	0.8
It is a pleasure to inform you that you have won $1.5M. Contact me. (jw11@example.com)	GAIN won() [finance_money]	PERFORM contact (jw11@...) []	0.9
Your dog could win prizes. Vote now.	GAIN win() []	PERFORM vote (http...) []	0.9
After you submit, we will pick finalists in each category. Users will vote on their favorite three winners.	GAIN pick() []	GIVE vote() []	0.6

Table 1: Representative system output for four emails: framings, asks, and confidence scores. Parentheses () designate clickable links (emails, URLs,...) and brackets [] indicate the ask/framing type of interest to the social engineer.

and recall of ask and framing detection; (2) Structural clues such as links provide a fuller context for asks (e.g., *click here*) and improve precision and recall of "top ask" detection (i.e., identification of critical asks for downstream response generation) from individual emails. These asks are then converted to a threat-intelligence representation where they make up the matchable pattern. Our approach is implemented and currently under evaluation by prominent transition partners. An accompanying risk model ties the target, and what is being asked of them, to an attack signature that can be mapped to a threat actor; the mapping then reveals other stages of the attack and what the motivation might be.

The next section presents the foundational background for asks and framing. We subsequently present our approach, focusing on the linguistic and structural knowledge adopted into our solution and describing our algorithm with representative examples. We present a range of experiments and results and discuss the upshot of our experiments and present related work, contrasting prior approaches to our own. We conclude with ideas for future work. By-products of this study are available at a website henceforth referred to as *Ask Detection* webpage, for a larger project called PANACEA: https://social-threats.github.io/panacea-ask-detection/.

2 Foundational Elements: Asks and Framing

The notions of *ask* and *framing* are defined as they pertain to our task of ask/framing detection in the SE context.

2.1 What is an Ask?

The notion of an *ask* is closely related to the notion of a request (Zemel 2017). Four related phenomena are subsumed under the heading of an *ask*: a) *proposal*, b) *offer*, c) *request* and d) *suggestion* (Couper-Kuhlen 2014). An *ask* is an action that "should be taken broadly to include other ways of asking than speaking" (Drew and Couper-Kuhlen 2014b). The nature of the action requested will influence both the form of the *ask* and the nature of the expected and actual response to the *ask* (Drew and Couper-Kuhlen 2014b). We identify two ask types: a topical descriptor that specifies the kind of thing being asked for (GIVE), and an action associated with its delivery or production (PERFORM). In SE attacks, asks are routinely used to recruit recipients to perform actions that provide money, information or system access.

Perhaps most importantly, an ask elicits relevant responses from the recipient, thus providing an opportunity to generate responses that elicit information about the attacker from the attacker. The same social obligation to respond to a request that a social engineer uses to elicit responses can be used to elicit information about the social engineer.

We have conducted pilot experiments with both staged and "live" social engineering attacks and noted that general conversational norms are observed, as expected. While one or both sides of the conversation engage in deception, they still need to accomplish their objectives, e.g., keeping the conversation alive, and obtaining information they want. Correspondingly, our approach treats ask/framing combinations as sense-making practices. All actors rely on shared knowledge of, and shared ability to recognize, how asks and framings fit together in conventional ways to produce meaningful interactions (Garfinkel and Sacks 1970; Pomerantz and Fehr 2011; Pomerantz et al. 2017; Sacks 1992; Schegloff 2007). We rely on this understanding of communicative practice in designing our system to both recognize and implement legitimate and deceptive messaging.

2.2 What is Framing?

Framing refers to linguistic and social resources used to persuade the recipient of an ask to comply and perform the requested social action. An ask creates a social obligation to respond, but does not necessarily provide an adequate basis for compliance with the ask. An *ask* must be "contextualized" to be persuasive (Huma, Stokoe, and Sikveland 2019). Recognizing and producing *framing* thus significantly shapes the kind of response the *ask* makes relevant.

Framing and the particulars of ask-construction enable creation of a "benefactive stance" (Clayman and Heritage 2014) that motivates compliance and also enables and constrains the ways that recipients respond (Huma, Stokoe, and Sikveland 2019). This may involve asking a question that makes production of an answer conditionally relevant. Alternatively, the benefit or cost that purportedly may accrue to the recipient (based on compliance or non-compliance) provides the basis upon which performance of the *ask* can be decided. Framing consists of just these specifications of benefit (GAIN) and cost (LOSE).

Social engineers rely on the fact that we routinely (often without much consideration) conform to social interaction conventions based on the credibility of the framing used. The efficacy of framing is not limited only to victims of SE attacks. Asks and framing directed at social engineers make them targets of counter-attacks.

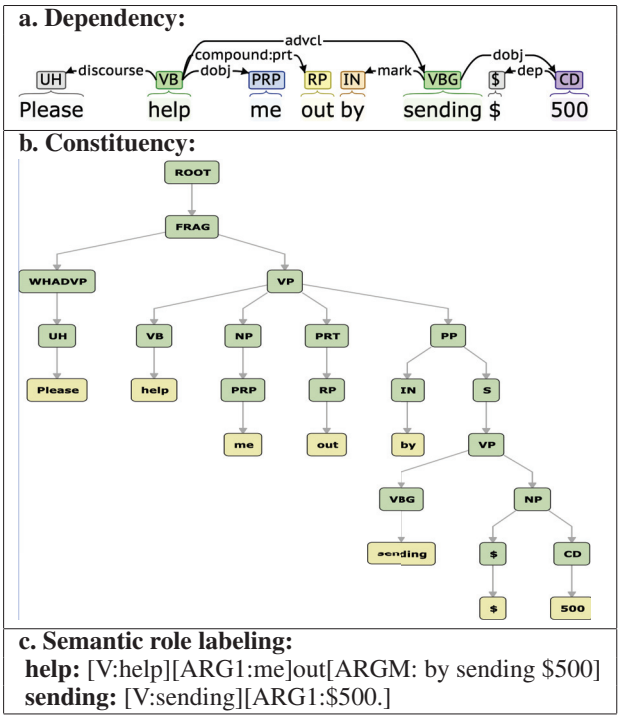

Figure 1: Dependency, Constituency, and Semantic Role Labeling for *Please help me out by sending $500*

3 Approach to Ask/Framing Detection

This section describes our approach to ask/framing detection, starting with the application of linguistic knowledge and structural knowledge, and finishing with a description of the algorithmic steps and confidence score.

3.1 Application of Linguistic Knowledge

a. Basic Language Tools: Detection of the actions associated with asks and framing relies on constituency parses and dependency trees, both taken from Stanford CoreNLP (Manning et al. 2014). As shown in Figures 1a and 1b, both *help* and *sending* are verbal (VB and VBG, respectively) and thus are extracted as candidate ask actions. Arguments are extracted using *semantic role labeling* (SRL) (Gardner et al. 2017) (see Figure 1c), e.g., ARG1 ($500) is identified as an argument of *sending*.[3] The dependency tree may not always yield all possible verbs, so back-off to the constituency tree is sometimes needed. Also, when SRL is not able to populate all arguments, the dependency tree is used to determine arguments for each ask or framing action.

b. Thesaurus.com and Lexical Conceptual Structure: Two thesaurus-like baselines and an extended verb classification (see Ask Detection webpage) are used to test linguistic and structural constraints on ask/framing detection. The first is a standard but relatively robust thesaurus (thesaurus.com) that yields four lists based on asks (PERFORM, GIVE) and framings (LOSE, GAIN). Examples are shown below (with total verb count in parentheses):

- PERFORM (44): achieve, act, do, execute, perform
- GIVE (55): commit, donate, grant, provide
- LOSE (41): expend, forefeit, squander, yield
- GAIN (53): clean, get, obtain, profit, reap

The second baseline is the STYLUS variant of Lexical Conceptual Structure (LCS) (Dorr and Voss 2018; Dorr and Olsen 2018), which groups verbs into classes according to syntactic behavior and underlying semantic structure (Levin 1993). For example, verbs corresponding to GIVE include those in classes such as *Contribute*, *Future Having*, and *Fulfilling*. Some examples are shown below:

- PERFORM (214): ask, bring, execute, rate, redeem
- GIVE (81): administer, contribute, donate, furnish
- LOSE (615): penalize, stick, punish, ruin
- GAIN (49): accept, earn, grab, win

Due to their size and coverage, in comparison to thesaurus.com, LCS classes are likely to predict more true positives, but also more true negatives, during ask/framing detection. The benefit and main distinguishing feature of LCS is its extensible organizational structure, which facilitates rapid modifications due to the grouping of similar verbs.

An extended LCS classification (LCS+) was thus produced from suspected scam/impersonation emails collected by PANACEA team members. Verbs from these emails were tied into particular LCS classes with matching semantic peers and argument structures (one person-day of effort). Modifications and examples are shown below:

- PERFORM (6 del, 44 added): connect, copy, notify
- GIVE (no changes): Orig. LCS covers GIVE verbs
- LOSE (174 del, 11 added): deny, forget, surrender
- GAIN (no changes): Orig. LCS covers GAIN verbs

c. Categorial Variation: We incorporate CATVAR (*Categorial Variation Database*) (Habash and Dorr 2003) to map between different parts of speech, e.g., *winner*(N) $\rightarrow$ *win*(V). Basic Language Tools (see above) focus primarily on processing verbal parts of speech (VB, VBG, VBD,...) which may miss some asks/framings, e.g., *you can reference your gift card* is an implicit ask to examine a gift card associated with the potential victim. CATVAR enables identification of the ask action as *refer*, a PERFORM verb.[4]

d. Verbal Processing: Verbal processing eliminates spurious asks containing verb forms such as *sent* or *signing* in *We sent you this email because you're signing up for a new account*. Verbal constraints rule out ask candidates tagged with parts of speech VBG or VBD, thus eliminating *sent* (VBD) and *signing* (VBG) as asks. These constraints do not apply to framing candidates, which may carry any verbal part of speech, e.g., *you won, you are winning*.

[3] Arguments are assigned ask/framing categories mentioned earlier (e.g., finance_money, personal, credentials).

[4] To our knowledge, no stemming or lemmatization is as comprehensive as CATVAR, which incorporates numerous resources (Brown, NOMLEX, WordNet, etc.), providing multiple variants for any event. For example, *develop* is classified with *developer* (N), *developing* (AJ), *development* (N), *developmental* (AJ), *developmentally* (AV), and several others (a total of 16 variants). The Ask Detection webpage includes a link to CATVAR.

3.2 Application of Structural Knowledge

a. Email structure and links: It is common for emails to include both HTML and plain text parts with similar semantic and pragmatic content. Social engineers exploit the conventional similarity between HTML and plain text by substituting malicious links and contact information where trusted resources are expected by the user (Hadnagy and Fincher 2015). This attack pattern for deception motivates us to focus on the `text/html` MIME type when it is available, activating pre-processing before the linguistic elements of the message are analyzed. Processed HTML parts result in text that is line split whenever `div`, `p`, `br`, or `ul` tags are encountered for improved sentence splitting. A unique tag is inserted as a placeholder for hyperlinks that may be recovered later as needed. Image tags are replaced with their alt text. Any element that conventionally or canonically indicates styling, scripting, quoting, replying, or a signature is removed.

b. Link Positioning: Social engineers employ many techniques to entice the potential victim to click on links, including a wide range of different link positionings:

- Click here
  Ask: PERFORM click(<URL>)[5]
- Click here
  Ask: PERFORM click(<URL>)
- Get ready to vote for the best-looking dog.
  Ask: PERFORM vote(<URL>)
- Contact me. I'm around Mon. (jw11@example.com)
  Ask: PERFORM contact(jw11@example.com)

In the first three cases above, *Basic* link processing assigns the link to the appropriate ask: the links are embedded in the sentence containing the ask (PERFORM) and its associated action (*click* or *vote*). By contrast, the fourth entails *Advanced* link processing to tie together a link with its corresponding ask-containing sentence, which is separated by intervening material. As we will see shortly, handling both cases increases ask-detection confidence, thus improving the detection of top asks on a per-email basis.

3.3 Algorithmic Steps and Confidence Score

Because our approach uses linguistically-motivated rules coupled with structural knowledge, no training data are needed. Algorithmic steps are described below.

a. Detect Ask/Framing actions: The first step for ask/framing detection is to extract the main action for each clause (recursively) from the dependency tree and the constituency parse shown earlier in Figure 1a,b. This is achieved first through application of basic language tools and also through application of CATVAR to detect actions that may be implicit in non-verbal forms, such as *reference* (which maps to the PERFORM form *refer*). Verbal constraints are then applied to rule out past and progressive actions (VBD/VBG) as asks. If ruled out, the action is considered as a framing candidate. If not ruled out, a priority scheme is applied, attempting to match the action against asks PERFORM and GIVE, in that order, from the lexical resource of interest (the thesaurus or LCS/LCS+). If this fails, an attempt is made to match the action against framings LOSE and GAIN, in that order, using the same lexical resource.

This priority scheme was devised to support overlapping ask actions, e.g., *send* is both a GIVE and PERFORM in LCS+, but in the context of a clickable link, it is deemed a PERFORM. In this way, *structural knowledge* influences the *linguistic choice* of ask. Similar overlap exists for framing, e.g., *retrieve* is both a GAIN or a LOSE, depending on the perspective of interest. Given that our application of ask detection is designed for SE interactions, it is assumed that a loss is intended for the potential victim (not for the social engineer); thus, LOSE is tested ahead of GAIN.

b. Determine Ask/Framing Arguments: Following the detection of an ask or framing action, basic language tools identify the arguments. For example, semantic role labeling identifies *$500* as ARG1 in the sentence *sending $500* (see Figure 1); this becomes an ask argument that is subsequently assigned an ask category as described next.

c. Assign Ask/Framing Category: Categories are associated with asks and framings, e.g., *sending $500* yields a GIVE ask with argument $500, which is in the finance_money category. Other examples are shown below:

- *...using your gift card*: scam_gift
- *Sign-up with your login and password*: credentials
- *...confirm with us via this email...*: personal

The categories are hierarchically organized, with a total number of 13 categories. From this categorization it is possible to deduce the likely goals of a would-be attacker, for use in downstream response generation.[6]

d. Detect Links: Links are detected through either basic or advanced link processing and these are associated with ARG1 of the ask (found by basic processing tools). The existence of a link boosts the confidence score for its associated ask. For example, a detached link is found via advanced link processing for: *Contact me. I'm around Mon. (jw11@example.com)*. Here, *me* is associated with the contact email address.

e. Apply Confidence Score: Application of confidence scores is based on preliminary trial-and-error studies and intuitions gleaned from processing development data. Observations are: (1) Past tense events are found not to be asks, thus assigned low or 0 confidence; (2) Non-past-tense events are more prevalently observed to be PERFORM asks if an ask category is specified (e.g., *finance_money* for *sending $500* above), thus assigned high confidence (0.8); (3) The vast majority of asks associated with URLs (e.g., jw11@example.com tied to *me* above) are found to be PERFORM asks, thus assigned a highest confidence (0.9); (4) GIVE combined with any ask category (e.g., *contribute $50* above) is less frequently found to be an ask, thus assigned slightly lower confidence (0.75); and (5) GIVE by itself is even less likely found to be an ask, thus assigned a confidence of 0.6 (e.g., *donate often*). (Automatic confidence scoring, training on actual data, is an area of future work.)

[5] <URL> refers to the URL embedded in the link.

[6] Although the full description of ask/framing categories is out of scope for this paper, these categories provided hints to the human adjudicator for the generation of our validation set.

f. Select Top Ask: Upon completion of the processing above, *Top Ask* selection produces the most important asks at the aggregate level of a single email. This is crucial for downstream processing of the framing and ask (i.e., automatic response generation). Asks are sorted based on their confidence scores, bringing those with the highest scores to the top. Those tied for first place are returned as the "top asks" for the email. For example, "PERFORM contact me (jw11@example.com)" is returned as the top ask for *Contact me. I'm around Mon. (jw11@example.com)*.

4 Evaluation Experiments and Results

This section describes a range of different experiments to demonstrate the impact of linguistic and structural knowledge on ask/framing detection. It has been argued that the evaluation of dialogue systems is best achieved through comparison to a large set of human-generated responses (Gupta et al. 2019). However, with the intermediate step of ask/framing detection—a well-defined sub-problem of the larger dialogue task in a social-engineering setting—it is possible to achieve a very informative evaluation using a single validation set with straight-forward labels.

We produce a validation set through human adjudication and correction (by a computational linguist) of initial ask/framing labels automatically assigned by our system to SRL-processed clauses from a held-out test set of 20 emails (472 clauses). The emails contain examples of everyday spam, targeted attacks by would-be social engineers, and test emails. The resulting validation set is used as a form of *ground truth* (see Ask Detection webpage) against which we measure clause-level precision/recall/F, as described below.

The adjudication task has a throughput of about 50 labeled clauses per hour, for a validation set size of 472 clause-level outputs in one person-day of work. Three types of labels are adjudicated and corrected for each clause-level output to produce the validation set: (a) ask labels for all asks identified in each email; (b) framing labels for all framings identified in each email; and (c) the top ask (or set of top asks), as determined by the confidence score, for each email—i.e., those considered the most critical, by a human adjudicator, for downstream response generation.

Our experiments focus on the range of true positives/negatives (TP, TN) and false positives/negatives (FP, FN) that arise with different combinations of linguistic and structural knowledge used for ask/framing detection.[7] We judge the success of our approach at the clause level (472 extracted verb-argument excerpts) using metrics of Precision (P), Recall (R), and F-measure (F). Test conditions for our experiments are shown in Table 2 for cases 1–6, excluding case 0 (thesaurus baseline). Each column includes all features of the prior column.

Table 3 provides all experimental results. McNemar tests were applied to determine statistical significance of performance changes between consecutive cases (McNemar 1947). Asterisked(*) rows have statistically significant improvements at 5% level.[8] Statistically significant improvements were found in moving from LCS to LCS+ (for Ask and Framing), in moving from LCS+ to LCS+Verbal (for Ask), and in moving from LCS+Verbal+CATVAR to Basic Link Processing (for TopAsk).

	1	2	3	4	4	6
Orig LCS Classes	Y					
Class expansions	Y	Y				
Verbal Processing	Y	Y	Y			
CATVAR	Y	Y	Y	Y		
Basic Link	Y	Y	Y	Y	Y	
Advanced Link	Y	Y	Y	Y	Y	Y

Table 2: Experimental conditions for Cases 1–6

5 Results Analysis: Failures and Successes

Experiments reveal that LCS+ improves detection of both asks and framings, with more than a six-fold increase in F over a standard thesaurus for ask detection (0.482). Higher results are obtained for ask-detection with verbal processing (0.496), and even higher when CATVAR is added (0.508). Two types of structural (link) processing techniques impact the top-ask results, ultimately achieving an F-score of 0.456, four times higher than that of the thesaurus-only baseline (0.094). We highlight certain cases of failure and success:

Case 0: Thesaurus.com yields 3 asks and 9 framings, and identifies 3 top asks. F-scores are modest, 0.30 or less for asks, framing, and top asks, e.g., *take part in this conference* is correctly identified as PERFORM, but *sign up* is missed.

Case 1: Original LCS Classes yield more correct asks and framings in comparison to Case 0, with more than a two-fold improvement in F for asks (from 0.072 0.157) and a 40% increase for framing (from 0.305 to 0.424). Unlike Case 0, LCS classes detect *sign up* as a PERFORM ask. However, false positives increase, particularly for asks (from 8 to 28), e.g., LCS incorrectly deems *we value your participation* to be a PERFORM ask. Correct top asks climb from 3 to 9, a three-fold improvement, e.g., LCS assigns *Did you send money?* as a top ask, where Case 0 fails to do so.

Case 2: LCS+ Classes increase correct asks from 8 to 34, and increase false positives from 28 to 34, over the original LCS with overall F-score improvements. For example, LCS+ correctly identifies *contact me* as a PERFORM, where the original LCS does not, but also incorrectly identifies *you would rather not receive* as a PERFORM. False positives for framing are reduced over the original LCS (30 to 10), e.g., the original LCS, but not LCS+, incorrectly detects LOSE for *dog could be a star*. False positives for correct top asks increase from 10 to 18, e.g., *we sent this email* is deemed a PERFORM, mitigated by verbal processing (below).

Case 3: Verbal Processing is coupled with LCS+, to filter past and progressive forms of predicted asks, thus eliminating sentences like *we sent this email* as an ask. This verbal constraint significantly reduces false positives for asks (34 to

[7] See TP/TN/FP/FN tallies under Experimental Details on the Ask Detection webpage.

[8] Tested values were correct responses (TP or TN) vs. incorrect responses (FP or FN), significance of change in total error rate.

Type	TP	TN	FP	FN	P	R	F
Case 0: Thesaurus Only							
Ask:	3	392	8	69	0.273	0.042	0.072
Framing:	9	422	25	16	0.265	0.360	0.305
TopAsk:	3	411	8	50	0.273	0.057	0.094
Case 1: Original LCS Classes							
Ask:	8	378	28	58	0.222	0.121	0.157
Framing:	14	420	30	8	0.318	0.636	0.424
TopAsk:	9	409	10	44	0.474	0.170	0.250
Case 2: LCS+ Classes							
Ask:*	**34**	365	34	39	0.500	0.466	0.482
Framing:*	**15**	437	10	10	**0.600**	**0.600**	**0.600**
TopAsk:	14	401	18	39	0.438	0.264	0.329
Case 3: LCS+&Verbal							
Ask:*	29	384	15	44	0.659	0.397	0.496
Framing:	**15**	437	10	10	**0.600**	**0.600**	**0.600**
TopAsk:	13	407	12	40	0.520	0.245	0.333
Case 4: LCS+&Verbal&CATVAR							
Ask:	30	384	15	43	**0.667**	**0.411**	**0.508**
Framing:	**15**	437	10	10	**0.600**	**0.600**	**0.600**
TopAsk:	13	407	12	40	0.520	0.245	0.333
Case 5: LCS+&Verbal&CATVAR&BasicLink							
Ask:	30	384	15	43	**0.667**	**0.411**	**0.508**
Framing:	**15**	437	10	10	**0.600**	**0.600**	**0.600**
TopAsk:*	17	411	8	36	0.680	0.321	0.436
Case 6: LCS+&Verbal&CATVAR&AdvancedLink							
Ask:	30	384	15	43	**0.667**	**0.411**	**0.508**
Framing:	**15**	437	10	10	**0.600**	**0.600**	**0.600**
TopAsk:	**18**	411	8	35	**0.692**	**0.340**	**0.456**

Table 3: Impact of combinations of linguistic and structural knowledge on ask/framing detection.

15) over LCS+ alone.[9] Correct top asks drop slightly (14 to 13) but false positives reduce significantly (18 to 12), e.g., *is being sent* is eliminated as a top ask. Although verbal constraints do not increase correct ask types, fewer false positives enables more effective downstream response generation. Verbal constraints combined with structural processing (below) yield more correct top asks than LCS+ alone, with F increasing about 21% (14 to 17) for basic link processing, and 5% (17 to 18) for advanced link processing.

Case 4: CATVAR raises true positives for asks slightly (29 to 30) without increasing false positives (15). Top asks remain at 13 and framings do not change. CATVAR adds one true ask—a case where *reference*(N) is mapped to *refer*(V). One explanation for CATVAR's unanticipated low impact is that social engineers generally do not employ verb-nominal combinations. Example emails developed among PANACEA team members include multi-word constructions such as *You **emerge winner** of $1M*. This is a place where CATVAR would have mapped to the verb *win*. However, true SE data are more likely to use phrases such as *You have won*.

Cases 5 and 6: Basic/Advanced Link Processing leverages structural knowledge, which does not impact correct asks (as expected), but directly impacts true and false positives for top asks. Basic Link processing increases correct top asks significantly (13 to 17) and also significantly reduces false positives (12 to 8), yielding significant F-score improvement (0.333 to 0.436). For example, basic processing eliminates *fail to bring* as an ask and advanced processing adds one more correct top ask (18): *open a tutorial*.

[9]Framing is not subject to verbal constraints; thus the number of framings does not change (cases 2–6 remain the same for framing).

6 Related Work

Security in online communication is a challenging problem, due to several issues: (1) the attacker's speed outpaces the ability of defenders to maintain indicators (Zhang et al. 2006); (2) the quality of phishing sites are high enough that users ignore alerts (Egelman, Cranor, and Hong 2008); (3) User training falls short as users quickly forget the material and fall prey to previously studied attacks (Caputo et al. 2013); and (4) defensive system maintainers may not always take into account the context, motivations, and socio-economic status of the targeted user (Oliveira et al. 2017).

Numerous studies (Bakhshi, Papadaki, and Furnell 2008; Karakasilitiosis, Furnell, and Papadaki 2006) have demonstrated human susceptibility to SE attacks. Moving from bots that detect SE attacks to those that produce "natural sounding" responses and engage the attacker to elicit identifying information is the next advance in this arena. We take the first step in our work on ask/framing detection, to be used for downstream processing by conversational agents (CAs).

CAs suffer from the problem of generating dull and generic responses (Gao et al. 2019; Santhanam and Shaikh 2019). To address this issue, topic models are used to produce focused responses augmenting existing neural based approaches to CAs (Dziri et al. 2018). Targeted responses employ self-disclosure as a strategic approach for building an engaging conversation (Ravichander and Black 2018). For SE detection, topic models (Bhakta and Harris 2015) and NLP of conversations (Sawa et al. 2016) are leveraged. However, all of these approaches are limited to a pre-defined set of topics, constrained by the training corpus.

Prior work on detecting and predicting persuasion in discussions (Hidey and McKeown 2018) leverages argument structure as a determining factor for judging when a persuasive attempt might be successful. This work has been adopted for (subreddit) forum discussions specifically dedicated to changing opinions (ChangeMyView subreddit). Our work is related to this but aims to achieve effective dialogue for countering (rather than adopting) persuasive attempts.

Many approaches address issues above via machine learning. Structured knowledge representations of scripts embedded in latent space are used to detect and compare similar events (Li and Goldwasser 2019). This is useful for determining whether an email resembles a password reset email typically sent from an organization's IT department. Comparison of multiple responses in a chatbot setting has been shown to improve correlation of evaluation metrics with human judgement (Prakhar Gupta and Bigham 2019). This can be used to select a chatbot model that performs better in an open domain. However, unlike our approach, these approaches require extensive model training.

Closest to our work is text-based semantic analysis for detection of SE attacks (Kim et al. 2018). Our work differs in

that it focuses not just on *detecting* an attack, but on *engaging* with an attacker in ways that leverage *asks* and *framings*. Whereas a chatbot might be employed to warn a potential victim that an attack is underway, e.g., based on malicious content, our chatbots are designed to communicate with a social engineer in ways that elicit identifying information. Extraction of asks/frames supports generation of responses to the attacker—with an end goal of eliciting attributable information about the attack in order to identify them.

7 Conclusions and Future work

We appeal to foundational notions of *ask* and *framing* to characterize SE attacks for downstream response generation. We conduct experiments to determine the impact of linguistic and structural knowledge on ask and framing detection, coupled with top ask selection. We make use of parts of speech, categorial variations, and verb classes informed by lexical-conceptual structure to yield significant precision and recall improvements for ask and framing detection. We also apply structural clues such as link detection for improved recall of "top asks" for individual emails.

As noted earlier, categorial variations (CATVAR) provide only small improvements to ask-detection performance. Certain types of multi-word expressions merit further investigation within the SE domain, most notably light verb constructions, which are not yet accommodated in our approach: *take note* vs. *notice*, *take into account* vs. *account for*, etc.

We also expect to enrich our LCS+ classes through the addition verbs that do not currently appear in LCS+. For example, several thesaurus verbs for GIVE are missing from the LCS+ verbs for GIVE: *allow, commit, endow, sell*. A future goal is to use a class-based combination of multiple resources to improve ask-detection performance compared to using either resource alone.

We are conducting experiments to demonstrate the utility of ask detection in an extrinsic evaluation of a conversational agent. We hypothesize that ask detection enables more targeted responses than would otherwise be generated and that the degree of sophistication of the ask-detection component correlates with the reliability of response generation. Example conversations are included on the Ask Detection page.

Acknowledgments: This work was supported by Defense Advanced Research Projects Agency (DARPA) under Contract No FA8650-18-C-7881. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of AFRL, DARPA, or the U.S. Government.

References

Bakhshi, T.; Papadaki, M.; and Furnell, S. 2008. A practical assessment of social engineering vulnerabilities. In *Human Aspects of Information Security and Assurance (HAISA)*.

Bhakta, R., and Harris, I. G. 2015. Semantic analysis of dialogs to detect social engineering attacks. *Proc. 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)* 424–427.

Caputo, D. D.; Pfleeger, S. L.; Freeman, J. D.; and Johnson, M. E. 2013. Going spear phishing: Exploring embedded training and awareness. *IEEE Security & Privacy* 12(1):28–38.

Clayman, S., and Heritage, J. 2014. Benefactors and beneficiaries: Benefactive status and stance in the management of offers and requests. In Drew, Paul and Couper-Kuhlen, Elizabeth., ed., *Requesting in social interaction*. John Benjamins Publishing Company. 55–86.

Couper-Kuhlen, E. 2014. What does grammar tell us about action? *Pragmatics* 24(3):623–647.

Dalton, A.; Dorr, B. J.; Liang, L.; and Hollingshead, K. 2017. Improving cyber-attack predictions via unconventional sensors discovered through information foraging. In *Proc. 2017 International Workshop on Big Data Analytics for Cyber Intelligence and Defense*.

Dalton, A.; Zemel, A.; Masoumzadeh, A.; Bhatia, A.; Dorr, B.; Mather, B.; Hebenstreit, B.; Al-Shaer, E.; Ellisa Khoja, E. C. J.; Bunch, L.; Vlahovic, M.; Liu, P.; Pirolli, P.; Shah, R.; Cartacio, S.; Shaikh, S.; Santhanam, S.; Dhaduvai, S.; Strzalkowski, T.; and Karimi, Y. 2019. Modeling social engineering risk using attitudes, actions, and intentions reflected in language use. In *Proc. Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, FL, USA, May 19-22 2019.*

Dinan, E.; Logacheva, V.; Malykh, V.; Miller, A. H.; Shuster, K.; Urbanek, J.; Kiela, D.; Szlam, A.; Serban, I.; Lowe, R.; Prabhumoye, S.; Black, A. W.; Rudnicky, A. I.; Williams, J.; Pineau, J.; Burtsev, M.; and Weston, J. 2019. The second conversational intelligence challenge (convai2). *CoRR* abs/1902.00098.

Dorr, B. J., and Olsen, M. B. 2018. Lexical conceptual structure of literal and metaphorical spatial language: A case study of push. In *Proc. First International Workshop on Spatial Language Understanding*, 31–40.

Dorr, B., and Voss, C. 2018. STYLUS: A resource for systematically derived language usage. In *Proc. First Workshop on Linguistic Resources for Natural Language Processing*, 57–64. Santa Fe, New Mexico, USA: Assoc. for Computational Linguistics.

Drew, P., and Couper-Kuhlen, E. 2014a. *Requesting in social interaction*. John Benjamins Publishing Company.

Drew, P., and Couper-Kuhlen, E. 2014b. Requesting – from speech act to recruitment. In Drew, Paul and Couper-Kuhlen, Elizabeth., ed., *Requesting in social interaction*. John Benjamins Publishing Company. 1–34.

Dziri, N.; Kamalloo, E.; Mathewson, K. W.; and Zaiane, O. 2018. Augmenting neural response generation with context-aware topical attention. *arXiv preprint arXiv:1811.01063*.

Egelman, S.; Cranor, L. F.; and Hong, J. 2008. You've been warned: An empirical study of the effectiveness of web browser phishing warnings. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, 1065–1074. New York, NY, USA: ACM.

Gao, J.; Galley, M.; Li, L.; et al. 2019. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval* 13(2-3):127–298.

Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N. F.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. S.

2017. Allennlp: A deep semantic natural language processing platform. In *Proc. of Workshop for NLP Open Source Software (NLP-OSS)*.

Garfinkel, H., and Sacks, H. 1970. On formal structures of practical actions. In McKinney, John C. and Tiryakian, Edward A., ed., *Theoretical sociology: Perspectives and developments*. Appleton Century Crofts. 337–366.

Gupta, P.; Mehri, S.; Zhao, T.; Pavel, A.; Eskénazi, M.; and Bigham, J. P. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. In *Proc. 20th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2019)*.

Habash, N., and Dorr, B. J. 2003. A categorial variation database for english. In *In Proc. Human Language Technology and North American Assoc. for Computational Linguistics (NAACL) Conference*, 96–102.

Hadnagy, C., and Fincher, M. 2015. *Phishing Dark Waters*. Wiley Online Library.

Hidey, C., and McKeown, K. 2018. Persuasive Influence Detection: The Role of Argument Sequencing. In *Proc. Thirty-Second AAAI Conference on Artificial Intelligence*, 5173–5180.

Hollingshead, K.; Dorr, B. J.; Dalton, A.; and Barton, M. 2019. Does outrage signal cyber attacks? predicting "bad behavior" from sentiment in online content. In *Proc. Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, FL, USA, May 19-22 2019.*, 185–190.

Huma, B.; Stokoe, E.; and Sikveland, R. O. 2019. Persuasive conduct: Alignment and resistance in prospecting "cold" calls. *Journal of Language and Social Psychology* 38(1):33–60.

Karakasilitiosis, A.; Furnell, S. M.; and Papadaki, M. 2006. Assessing end-user awareness of social engineering and phishing. In *Proc. Australian Information Warfare and Security Conference*.

Kazakova, V. A.; Hwang, J. D.; Dorr, B. J.; Wilks, Y.; Gage, J. B.; and Memory, A. 2019. Splain: Augmenting cybersecurity warnings with reasons and data. In *Proc. Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, FL, USA, May 19-22 2019.*

Kim, M.; Song, C.; Kim, H.; Park, D.; Kwon, Y.; Namkung, E.; Harris, I. G.; and Carlsson, M. 2018. Catch me, yes we can!-pwning social engineers using natural language processing techniques in real-time.

Levin, B. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. The University of Chicago Press.

Li, C., and Goldwasser, D. 2019. Encoding social information with graph convolutional networks forPolitical perspective detection in news media. In *Proc. 57th Annual Meeting of the Assoc. for Computational Linguistics*, 2594–2604. Florence, Italy: Assoc for Computational Linguistics.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S. J.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *Assoc. for Computational Linguistics (ACL) System Demonstrations*, 55–60.

McNemar, Q. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* 12(2):153–157.

Oliveira, D.; Rocha, H.; Yang, H.; Ellis, D.; Dommaraju, S.; Muradoglu, M.; Weir, D.; Soliman, A.; Lin, T.; and Ebner, N. 2017. Dissecting spear phishing emails for older vs young adults: On the interplay of weapons of influence and life domains in predicting susceptibility to phishing. In *Proc. 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, 6412–6424. New York, NY, USA: ACM.

Perera, I. E.; Allen, J. F.; Galescu, L.; Teng, C. M.; Burstein, M. H.; Friedman, S. E.; McDonald, D. D.; and Rye, J. M. 2017. Natural Language Dialogue for Building and Learning Models and Structures. In *Proc. Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 5103–5104.

Perera, I.; Hwang, J.; Bayas, K.; Dorr, B.; and Wilks, Y. 2018. Cyberattack prediction through public text analysis and minitheories. In *Proc. 2018 International Workshop on Big Data Analytics for Cyber Intelligence and Defense*, 3001–3010.

Pomerantz, A., and Fehr, B. J. 2011. Conversation Analysis: An approach to the analysis of Cocial interaction. In van Dijk, Teun., ed., *Discourse Studes: A Multidisciplinary Approach*. Sage. 165–190.

Pomerantz, A.; Raymond, G.; Lerner, G.; and Heritage, J. 2017. Inferring the purpose of a prior query and responding accordingly. *Enabling human conduct: Studies of talk-in-interaction in honor of Emanuel A. Schegloff* 61–76.

Prakhar Gupta, Shikib Mehri, T. Z. A. P. M. E., and Bigham, J. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. In *Proc. 20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Stockholm, Sweden: Assoc. for Computational Linguistics.

Ravichander, A., and Black, A. W. 2018. An empirical study of self-disclosure in spoken dialogue systems. In *Proc. 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, July 12-14, 2018*, 253–263.

Sacks, H. 1992. *Lectures on Conversation, Volumes 1 & 2*. Blackwell.

Santhanam, S., and Shaikh, S. 2019. A survey of natural language generation techniques with a focus on dialogue systems-past, present and future directions. *arXiv preprint arXiv:1906.00500*.

Sawa, Y.; Bhakta, R.; Harris, I.; and Hadnagy, C. 2016. Detection of social engineering attacks through natural language processing of conversations. In *Proc. 2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, 262–265.

Schegloff, E. A. 2007. *Sequence Organization in Interaction: A Primer in Conversation Analysis*. Cambridge University Press.

Yusupov, I., and Kuratov, Y. 2018. NIPS conversational intelligence challenge 2017 winner system: Skill-based conversational agent with supervised dialog manager. In *Proc. 27th International Conference on Computational Linguistics*, 3681–3692. Santa Fe, New Mexico, USA: Assoc. for Computational Linguistics.

Zemel, A. 2017. Texts as actions: Requests in online chats between reference librarians and library patrons. *Journal of the Assoc. for Information Science and Technology* 67(7):1687–1697.

Zhang, Y.; Egelman, S.; Cranor, L. F.; and Hong, J. I. 2006. Phinding phish: Evaluating anti-phishing tools. Technical report, Carnegie Mellon University.