The Design of Reciprocal Learning Between Human and Artificial Intelligence

ALEXEY ZAGALSKY*, DOV TE'ENI*, and INBAL YAHAV, Coller School of Management, Tel Aviv University, Israel
DAVID G. SCHWARTZ, Faculty of Social Sciences, Bar-Ilan University, Israel
GAHL SILVERMAN, Coller School of Management, Tel Aviv University, Israel
DANIEL COHEN, Faculty of Social Sciences, Bar-Ilan University, Israel
YOSSI MANN and DAFNA LEWINSKY, Faculty of Humanities, Bar-Ilan University, Israel

The need for advanced automation and artificial intelligence (AI) in various fields, including text classification, has dramatically increased in the last decade, leaving us critically dependent on their performance and reliability. Yet, as we increasingly rely more on AI applications, their algorithms are becoming more nuanced, more complex, and less understandable precisely at a time we need to understand them better and trust them to perform as expected. Text classification in the medical and cybersecurity domains is a good example of a task where we may wish to keep the human in the loop. Human experts lack the capacity to deal with the high volume and velocity of data that needs to be classified, and ML techniques are often unexplainable and lack the ability to capture the required context needed to make the right decision and take action. We propose a new abstract configuration of Human-Machine Learning (HML) that focuses on *reciprocal* learning, where the human and the AI are collaborating partners.

We employ design-science research (DSR) to learn and design an application of the HML configuration, which incorporates software to support combining human and artificial intelligences. We define the HML configuration by its conceptual components and their function. We then describe the development of a system called Fusion that supports human-machine reciprocal learning. Using two case studies of text classification from the cyber domain, we evaluate Fusion and the proposed HML approach, demonstrating benefits and challenges. Our results show a clear ability of domain experts to improve the ML classification performance over time, while both human and machine, collaboratively, develop their conceptualization, i.e., their knowledge of classification. We generalize our insights from the DSR process as actionable principles for researchers and designers of 'human in the learning loop' systems. We conclude the paper by discussing HML configurations and the challenge of capturing and representing knowledge gained jointly by human and machine, an area we feel has great potential.

CCS Concepts: • **Information systems** → **Data analytics**; • **Human-centered computing** → **Human computer interaction (HCI)**; • **Computing methodologies** → *Natural language processing*.

*These two authors contributed equally

Authors' addresses: Alexey Zagalsky, alexeyz@mail.tau.ac.il; Dov Te'eni, teeni@tauex.tau.ac.il; Inbal Yahav, inbalyahav@tauex.tau.ac.il, Coller School of Management, Tel Aviv University, Tel Aviv, Israel; David G. Schwartz, David.Schwartz@biu.ac.il, Faculty of Social Sciences, Bar-Ilan University, Ramat Gan, Israel; Gahl Silverman, gsilverman@tauex.tau.ac.il, Coller School of Management, Tel Aviv University, Tel Aviv, Israel; Daniel Cohen, daniel.cohen@biu.ac.il, Faculty of Social Sciences, Bar-Ilan University, Ramat Gan, Israel; Yossi Mann, Yosef.Man@biu.ac.il; Dafna Lewinsky, Dafna.Lewinsky@post.idc.ac.il, Faculty of Humanities, Bar-Ilan University, Ramat Gan, Israel.

2573-0142/2021/10-ART443 $15.00
https://doi.org/10.1145/3479587

Additional Key Words and Phrases: text classification; cyber-security; human intelligence; AI; context; feedback; accuracy; explainabilitiy

ACM Reference Format:
Alexey Zagalsky, Dov Te'eni, Inbal Yahav, David G. Schwartz, Gahl Silverman, Daniel Cohen, Yossi Mann, and Dafna Lewinsky. 2021. The Design of Reciprocal Learning Between Human and Artificial Intelligence. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 443 (October 2021), 36 pages. https://doi.org/10.1145/3479587

1 INTRODUCTION

The explosion of communication over the Internet and mobile channels brings with it vast amounts of data generated as posts on social media, emails, chats, text and multimedia messages. These behavioral big data provide opportunities for communication analysis with text classification that were infeasible a decade ago, but also present challenges in processing the big data [57]. One prominent area of research and application that is enabled and enhanced by communication analysis is the support of decision making and sense-making, an area that necessitates advanced automation and artificial intelligence (AI) to provide the processing capacity and speed of analysis required to make practical, often near real-time, decisions. Indeed, new AI techniques have demonstrated remarkable progress in the last decade [37]. Hence, large-scale automatic analysis of communication has now become *necessary* and *feasible* for supporting decision making.

Although the performance of automatic AI-based analysis of behavioral data has progressed, it is still unsatisfactory [36]. More specifically, automatic communication analysis methods based on text classification that are geared to action, such as the detection of suspicious behaviors, have not yielded sufficiently accurate results for operational purposes [2]. One possible approach for improving the detection performance is to "keep the human in the loop" [31, 55] in order to boost the artificial intelligence with human intelligence, going beyond the current practice of involving the human primarily during the initial training of machine learning (ML) algorithms [58]. Easier said than done, however. Indeed, in this paper we begin by analyzing the challenges of keeping the human in the loop and follow by designing the corresponding solutions. Combining human intelligence with artificial intelligence introduces challenges not only in allocating tasks between human and machine but also in communicating the knowledge from one to the other. We build on Suchman's [63] idea of *human-machine configurations* to delineate the composition and bounds of our analysis, and take a joint cognitive-system perspective, in which the human and the artificial are cooperating partners aiming to learn rather than one being a tool for the other [71]. This perspective leads to an allocation of tasks that requires mutual intelligibility [4, 62] and dialog between the human and the artificial in order to *perform and learn* more effectively.

Our overarching goal is to create a general framework for combining human and artificial intelligence in which both learn reciprocally. We borrow the notion of *reciprocal learning* from a multidisciplinary theory of learning in (human) dyads [34]. The theory was inspired by Vygotsky's claim that the development of human intelligence is achieved by interactively learning from others and co-producing an understanding of the world rather than by individually accumulating separate pieces of knowledge [66]. In applying the theory of reciprocal learning to human-machine dyads, we adopt the theory's cognitive mechanisms of dealing with complex learning. For instance, reciprocal learning includes cognitive mechanisms, such as switching between learners with different perspectives and switching between different contexts, which reinforce mutual learning and result in a multiplier effect of learning [34]. Unlike traditional ML with reinforcement, not only does the human reinforce ML, but, additionally, the machine reinforces human learning.

In this paper, we describe how we developed **design principles for supporting reciprocal-learning configurations**, and applied them to a prototype called *Fusion*. These configurations, which we call *Human-Machine Learning* (HML) configurations, are conceptual systems consisting of

humans and machines that interactively leverage their intelligences for learning to solve problems, analyze, explain, and judge. Unlike the support of reinforced ML, HML configurations must be designed to support the mutual learning of machine and human. Moreover, systems that rely on reciprocal learning are distinct from systems that support other human-computer collaborations in at least two ways. First, the long standing relative-performance criterion for allocating tasks between the human and the machine [22] must be expanded to more complex and fuzzy criteria that consider also the transfer of control and responsibility from human to machine [53]. Second, the complexity of human-machine communication is higher when facing intelligence-intensive tasks, such as gaining insights, mindful judgment, creativity, and contextualization, compared with routine, structured, or programmed tasks. To deal with the challenges of higher complexity of both task allocation and communication, we design the combination of human and artificial intelligence at two levels: *the functional level* that determines who does what, i.e., the task allocation between human and machine and *the communication level* that determines what and how is communicated between them.

The task selected for our work is text classification applied to the cyber security domain, where in a "sea of data", practitioners and researchers are interested in detecting suspicious communication that may indicate, for example, fraud, drug related transactions, or terror acts. Once detected, the suspicious message triggers an action in real-time to further investigate and possibly prevent the crime (the action is beyond the scope of our research). Our working assumption is that an ongoing interaction between human intelligence (an expert analyst) and artificial intelligence (ML classification models), should be designed not only to jointly classify texts accurately, but also to learn to cope with the dynamic nature of human communication. We believe that task allocation in text classification should not be determined only according to performance (accuracy of ML classification), but must also consider the expert's learning and mutual intelligibility between the expert and ML models.

The prototype we developed, Fusion, supports and combines both contemporary ML text classification methods and human (qualitative) text analysis methods to detect suspicious communication. While we necessarily adapt extant ML methods and qualitative methods to fit the classification problem, our contribution is in fusing the human and the machine learning. To ground our design of Fusion, we used two case studies of identifying suspicious communication in Darknet forums. The first case study aimed to identify illegal drug activities (drug usage, solicitation, selling, and purchasing) within "Hidden Answers", a general purpose Darknet Q&A site. The second case study aimed to identify expert hackers on the cybercriminal Darknet forum "BitsHacking". We use both case studies and dedicated domain experts throughout our iterative design process of Fusion, to improve the collaborative interaction between these domain experts and the software system, and to evaluate our prototype.

Our design science research (DSR) produces two artifacts that are described, respectively, in the paper's two parts: a theory-based conceptual artifact [3] referred to as *the HML configuration* (in sections 2 and 4), and a technical artifact called *Fusion* (in sections 5-6). The first artifact is the HML configuration that offers a novel paradigm for continuous reciprocal human-machine learning. The second artifact, a component of the HML configuration, exemplifies the design for supporting reciprocal learning. Sections 3-6 describe the methodology and case studies for developing Fusion, the functionality, architecture, human computer interaction of Fusion, and its evaluation with domain experts. In the final sections, we discuss the validity of the HML configuration in practice, the way HML can change the practice of combining human and artificial intelligences, and the revealed challenges faced by the domain experts, which help plot directions for future research. We conclude the paper by providing researchers and practitioners design principles for building HML support systems.

We thus offer **two key contributions**: (1) an HML configuration in which both human and machine learn reciprocally, which we believe can impact the practice of combining human and machine intelligences, and (2) lessons learned from the design experience and its resulting prototype to support reciprocal learning, which are presented in the form of synthesized design principles.

2 BACKGROUND

To better understand systems that combine human and artificial intelligence for text classification, we begin by reviewing AI-based text classification and its challenges, especially the challenge of leveraging context. Classifying a text message relies on *determining the meaning* of the content in light of its context and accordingly *deciding on the appropriate classification*. Humans and machines use different methods for determining context. Therefore, HML configurations must allocate the specific methods for human or machine to implement, and must facilitate the communication process between them. In this section, we review these methods at two levels: (1) at the *functional level* tasks are allocated between human and machine, and (2) at the *communication level* reciprocal feedback enables coordination and learning between human and machine. Section 4 later builds on these aspects to propose a human-machine framework that is the basis for designing Fusion (as described in Sections 5 and 7.5).

2.1 AI-based Text Classification of Messages in Context

In recent years, the need for and use of AI methods to accurately classify text has grown significantly. Despite the advances made in language models for text classification [37] and the progress in machine learning methods to leverage context for improving classification [41], the process of ML-based text classification has remained relatively stable. Its main steps include: data preprocessing, feature engineering and selection, model selection and fitting, and model evaluation [37]. We depict this process in Figure 1. Kowsari *et al.* [37] have recently outlined a variety of techniques that help accomplish each of these steps, e.g., preprocessing with term based tokenization, feature engineering with bag-of-words or word embedding techniques, building a model with ML or deep learning, and model evaluation based on accuracy or other criteria. Each technique has its advantages and disadvantages, therefore choosing between them depends on the goal and the characteristics of the corpus of interest.

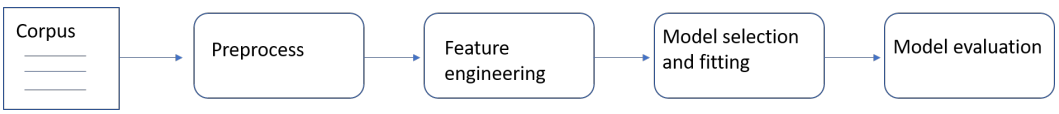


Fig. 1. The main steps of AI-based text classification (adapted from Kowsari *et al.* [37]).

What is often overlooked in this process is the context. Context is what gives meaning to data, i.e., textual data we wish to be classified. Priss [47] formally defines context as the set of objects and attributes together with their relation to each other. In text classification, context is used regularly for expanding the meaning of words (or structures of words). Mathematically, context can be built from the corpus being classified or from an external (enhancement) corpus, a technique called word embedding [42].

Word embedding techniques are considered context-aware, in the sense that terms are modeled as an integral part of their linguistic surrounding [18, 36, 47]. Though addressing the context in the local sense is important, it is sometimes necessary to understand the broader context of a text, which manifests a hidden *"internal conceptualization in the human mind"* called a *"meaning*

space" [36]. In contrast to the meaning space, which is free of noise and uncertainty, the linguistic space of the textual corpus (on which embedding techniques operate) is noisy and incomplete. The meaning space is often needed to interpret the message correctly. One challenge to ML in a noisy space is to generate features that represent the meaning space. Another challenge is the ability to learn from rare cases and imbalanced data [13].

The challenges of automated methods in modeling a broader context are even more pronounced when the data have some behavioral aspects, in the sense that they contain elements of intention, deception, emotion, reciprocation, herding, or other forms of human behavior [56], as in the case of online discourse [1, 2, 47, 57, 73]. This is particularly true for Darknet communication, where people who use it usually have something to hide or prefer to communicate anonymously [70], requiring common data-science tools to be tailored to mining and analyzing messages in the Darknet [14].

Recognizing these challenges, recent attempts began examining ways to automatically incorporate layers of context to language models. Common examples are the use of internal knowledge base with semantic NLP algorithms (e.g., Gardner *et al.* [24]), and the use of an external knowledge base (e.g., Wang *et al.* [67]). Alongside these attempts, in the general AI research community [31, 58], there is growing agreement that not all decision-related tasks can be delegated to AI. We believe that this applies to the text classification field as well, and that different configurations of human and artificial intelligence are therefore needed. This is particularly true when the text classification task occurs in dynamic environments, in which the context changes frequently, like in the case of online discourse. Human communication changes as its context changes and therefore classification systems must learn and adapt continuously. Leveraging context may be one of the advantages humans bring to automatic text classification, however, when applying context to HML configurations it will be necessary to consider how context is used differentially by humans and machines.

2.2 Human Text Classification

The human approach to text analysis differs from the AI-based classification described above, particularly in the way humans use context when interpreting messages. In an HML configuration, and unlike extant AI research on human augmented classification, humans are not only seen as a means for improving the machine's learning, but they themselves aim to learn. Learning in practice has been likened to a *spiral of knowledge creation* that pauses to take action and resumes to upgrade itself continuously [45, 69]. The knowledge creation spiral grows to include more dialog, more information, more interpretations and perspectives, and more confidence in putting the knowledge to action. People derive concepts and relationships from holistic observations, imagination, and rare nuances, in a process called *sense-making*. The process of sense-making involves incremental learning and testing old beliefs and interpretations in light of new data, new perspectives, and new contexts. Sense-making results in some abstract view of the concrete data, i.e., a new *"presumptive understanding through progressive approximations"* [69, p.412]. Moreover, the process iterates, often in a trial-and-error manner, between a highly contextualized interpretation of a message to a more abstract, and necessarily decontextualized, view of the message [50, 61, 74]. The result of sense-making, which we refer to as *'conceptualization'*, may take the form of a list of concepts (themes, constructs, abstractions) and possibly the relationships between the concepts, as well as, classification criteria (rules). The conceptualization is the basis for deciding on classifications of objects, which either observed directly (e.g., an image of a face) or understood from words in a text [27]. Humans exhibit a distinct capacity for leveraging context when making sense of human communication [19]. We employ this capacity in our proposed HML configuration (described in Section 4), e.g., when the domain expert interprets the machine's false-classifications or observes unexpected behavior.

Understanding the broader context enables a more effective processing of the data [36]. Additionally, some consider context as going beyond the text, explicitly or implicitly; context reflected by the entire corpus can be used to form a perspective from which the analyst builds interpretations. E.g., analysts who come from different professional backgrounds or having different native-languages may hold different perspectives that lead to different interpretations and conflicting classifications [19]. Thus, broader contexts give rise to the opportunity to entertain more perspectives that enrich the processes of sense-making and improve learning [69], a process referred to as perspective taking [9], which is another central mechanism of reciprocal learning [34].

Humans use context naturally, as context is central to human perception, categorization, reasoning, and communication [26]. Domain experts, especially, are capable of seeing the big picture and using it to focus their judgment of the more specific messages, their distinctions and commonalities [72], offering more opportunities for humans to complement computers. Nevertheless, combining human and artificial intelligences for decision making requires attention and careful application [5, 68]. In some cases, human judgment has been found to add accuracy to algorithmic techniques, such as forecasting, but not in others. Therefore, human experts should focus only on cases where contextual information supports more accurate interpretations. The contingent effect of context on performance has been documented in several areas. Bernardy *et al.* [7] found that context improves acceptability ratings for ill-formed sentences, not for well-formed. They also found that context helps unsupervised systems to model acceptability. Katz and Te'eni [35] found that adding context to messages was effective only when their complexity was high. It follows that allocating to the human the task of leveraging context for better classification should be managed for contingencies.

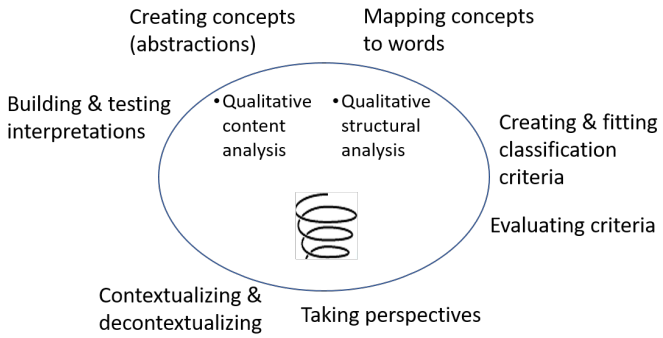


Fig. 2. Activities in the *'spiral-like'* human sense-making process for text classification.

Figure 2 depicts a set of unordered activities associated with sense making. The apparent contrast between the AI-based text classification (shown in Fig. 1) and human sense-making process raises the question of compatibility, i.e., the need for structuring the sense-making process to ensure an effective and consistent output that can be fed into machine classification models. Several techniques that allow this are used in qualitative research methods for analyzing text. Two such techniques that can be used sequentially are: (1) *qualitative content analysis* which combines concept-driven and data-driven categorization to guide classification [52]. The resulting categories form part of the expert's conceptualization. And, (2) *structural analysis* in which words and other symbols are extracted to relate the conceptualization to the text [28]. Using these qualitative analysis techniques, the domain experts employed for our case studies, generated a lexicon of categories and related terms. These lexicons are similar in structure to the renown LIWC lexicon [64], yet differ in the

fashion they are generated in the HML configuration—instead of using general categories (e.g., universal emotions), categories rooted in the specific corpus are generated by domain experts to correspond to the specific ML classification task.

2.3 Human in The Learning Loop - The Functional Level

As noted in the introduction, the original designs of man-machine systems sought a division of labor that capitalized on the relative advantages of human versus machine in performing a task, e.g., human creativity vs. computerized calculations [22]. The criterion for allocating tasks to either human or machine was their *relative advantage* in performing a particular function. The same line of thought has been extended in several ways and applied in various domains over the years. For example, Ip *et al.* [33] produced 'Task Allocation Charts' that first decompose a task into subtasks and then allocate subtasks to the machine or (possibly multiple) operators. For example, given a high volume or high velocity of data to be classified, it may be infeasible to allocate the task to a human expert but the task can be broken into subtasks, e.g., the expert classifies only a selected subset of the data manually. Indeed, common task allocations in text classification are: (1) for data labeling, humans assign a class to texts in a training set, and then provide feedback to the machine by re-labeling false ML based classifications [39]; and (2) for feature engineering, humans generate or select the features with high prediction power, commonly in a trial-and-error fashion [72].

Classification accuracy can no longer be the sole criterion for task allocation. Indeed, several recent studies have examined possible human-machine configurations that take into account additional considerations, such as organizational implications [31], the degree of automation versus augmentation [48], the contingencies of configurations based on parameters like task complexity or ambiguity [58], and ethical issues such as discretion [15]. Keeping the human in the loop can therefore be argued on grounds other than relative advantages in task performance. For instance, the human remains in the loop to stay in control and take responsibility that comes with control, or to supervise the machine performance to assure quality and learning.

One additional consideration for task allocation that is most relevant to HML configurations is interpretability [43], which is the ability to interpret machine algorithms by the user. In fact, there may be a tradeoff between interpretability and accuracy when choosing a classification method. A loss in model interpretability may limit the effectiveness of human learning. For instance, when learning is important, an interpretable model such as *bag-of-words* that trains the model using words directly as features may be preferred over a word embedding model such as *Word2Vec* which offers higher accuracy but loses interpretability when transforming words into vectors.

Our proposed HML configuration, described in Section 4, uses the criteria above to allocate tasks of the machine's activities associated with feature engineering and classification modeling (shown in Fig. 1), and of the human activities associated with sense-making (shown in Fig. 2).

2.4 Human-Computer Interaction – The Communication Level

Our perspective of human-computer interaction is one of collaboration between partners [63], according to the allocation of tasks between them. Effective communication between human and computer to support coordinated performance and mutual learning is therefore critical for a successful HML configuration. In particular, dialog is an essential part of reciprocal learning and human sense making [69]. Human-machine communication is expected to be of high complexity because the issues communicated will generally be complex (e.g., explaining the rationale for classification), and the gap between human mental models and machine models may be substantial. For these reasons, the loss of interpretability discussed above is a critical constraint on effective communication in HML configurations. We discuss two aspects of effective communication, namely feedback and context, both related to interpretability.

Effective feedback from the machine to the human is essential for increasing human control over communication and ensuring its quality [55], and it is particularly significant when communication complexity is high [65]. The design of feedback depends on the role and context of the communication, and on the source of communication complexity too. Indeed, in HML configurations, the bi-directional communication between human and machine enabled by the feedback is essential for sense making [6]. In other words, an analyst working with a computer trying to make sense of texts will not be effective without appropriate feedback during the sense-making process. If the source of complexity is the gap between the classification model used by the machine and the user's mental model, feedback to increase understanding must be designed to consider that gap—i.e., interpret the machine's output in terms of the user's meaning space and its terminology.

Feedback in HML configurations can further be articulated in terms of context, namely, *Outcome feedback* versus *Explanatory feedback* that includes contextual information. In HML configurations, the communication between human and machine, operationalized with the bi-directional feedback, is difficult because of the communication gap between them. This gap is caused by the different classification models they hold and the different formalism they use. To ensure mutual understanding and overcome communication gaps, communicators rely on context around the core message, and the greater the gap between communicators (i.e., less common ground), the more layers of context are needed [65]. *Outcome feedback* indicates whether a prediction was correct or not (in the case of multiple predictions, it may indicate their overall accuracy). *Explanatory feedback* uses "problem-space" context to explain why a prediction is correct or incorrect (in the case of multiple predictions, it may explain low accuracy due to some bias in estimation). Explanatory feedback, compared with the higher-level outcome feedback, includes more concrete and detailed context around a core message (e.g., specific examples of true or false classifications). Research on learning with multimedia has shown that combinations of outcome (corrective) feedback and explanatory feedback ensure deeper learning [44]. More generally, computer support can combine multiple layers of context and deliver it interactively so that only relevant feedback is given at the right time in a manner that is easy to understand and use effectively [35].

3 DESIGN SCIENCE RESEARCH METHODOLOGY

Our research methodology builds on previous recommendations for design science research (DSR) [30, 54], which is aimed at studying the design and development of new artifacts as a means for gaining knowledge. Our DSR produces a conceptual artifact, namely *the HML configuration*, and a technical artifact, namely *the Fusion prototype*. The HML configuration offers a novel paradigm for continuous reciprocal human-machine learning. The second artifact, a component of the HML configuration, is a prototype we developed. It exemplifies the design for supporting reciprocal learning. The remaining sections of our paper are organized around the DSR stages shown in Table 1, which although presented sequentially, iterate in learning loops as described below.

The idea of a human-machine learning system, applied to a domain expert working with ML text classification algorithms, began as part of a research proposal submitted in 2017 and continued into the first few months of the project. We had assumed, correctly in retrospect, that the HML configuration would require software to support the challenges of reciprocal learning. We therefore developed the HML configuration and Fusion in parallel.

We formed an interdisciplinary team of nine researchers composed of: (I) the data science sub-team responsible for developing ML algorithms; (II) the domain experts who developed the explicit representation of their knowledge and participated in user studies; (III) a qualitative research-methods expert; and (IV) the design sub-team who formalized the human-machine processes and

Table 1. Our iterative research process outlined as four stages.

	DSR Stage \ Artifact	HML configuration	Fusion	Section
1.	Formulate research problem	Formulate HML with reciprocal learning	Specify design to support HML configuration	Sec. 4
2.	(a) Build		Develop the software	Sec. 5
	(b) Use		Users (domain experts) use Fusion to detect suspicious messages	
	(c) Evaluate	Measure accuracy and conceptualization	Test functionality and usability	Sec. 6
3.	Reiterate and redesign reciprocal learning	Augment HML configuration	Adjust Fusion	
4.	Generalize	Set boundaries on HML configuration	Formulate design principles	Sec. 7

oversaw Fusion's development. We also collaborated with a cybersecurity software company that specializes in data mining and analysis of communication in hidden social media (e.g., Darknet).

For research stage 1, the team reviewed several streams of research in AI-based text classification, intelligence analysis in the domains of crime and terror, design of human-computer intelligent collaboration, and mixed-methods analysis of unstructured conversational data (Section 2 summarizes this effort). We then spent several months of field work, team brainstorming, and experimentation with various options to arrive at an initial HML configuration, which also served to specify the initial functionality of Fusion.

Research stage 1 surfaced the challenge of interdisciplinary team work, particularly, the need for procedures, tools, and terminology to enable effective dialog. We built a common-ground dictionary representing multiple perspectives, formalized the workflows of each sub-team and the integration of individual efforts, and conducted team tutorials to obtain the minimal working knowledge required for collaboration. For example, in order to augment the features used for classification, the domain expert needed to understand which features affect classification and how, in the different types of ML models used.

Research stage 2 is a Build-Use-Evaluate sequence of steps constituting one iteration in developing Fusion and gaining knowledge for the next iteration. We worked on two corpora of data taken from two different forums on the Darknet, each corpus with its corresponding domain expert (subsection 3 describes these forums). We began by applying the Build-Use-Evaluate sequence to one corpus, and augmented the HML configuration and adjusted Fusion according to the lessons learned. We performed a second iteration on the first corpus to improve the interaction between human and machine. Only then, we added another corpus to the study and followed the same Build-Use-Evaluate sequence on it. In essence, we worked on a cascade of corpora, learning and interchanging between them as we proceeded, to develop our design knowledge and incorporate it into Fusion.

The *Build-Use-Evaluate* process was as follows:

- The development of Fusion began with gathering requirements from all stakeholders: the data scientists, the qualitative-methods expert, and the domain experts. We used Figure 3 as a map of the interaction and flow between human and machine. We also defined the data flow and interface between the domain expert and the system being developed (e.g., the structure of lexicon file). Fusion was developed in an agile-manner, beginning from a small

single-scenario prototype, and then iterating with short rapid iterations. Fusion is written in R and uses the Shiny framework[1] for R.

- The process of using Fusion consisted of multiple iterations between the domain expert and Fusion. These iterations were geared towards increasing the classification accuracy, which Fusion communicated to the domain expert along with additional explanatory feedback. The domain expert had the ability to select from several types of ML classification models offered by Fusion. During this process, the qualitative-methods expert guided the domain expert in developing the knowledge representation, which was part of the input into Fusion. A data scientist would only intervene to change and extend the ML models within Fusion, or to explore other external corpora.
- The design sub-team conducted ongoing evaluations of the HML configuration and the use of Fusion. Classification accuracy measures of different models were compared to a baseline model (several indices were used, such AUC, precision, and recall). Additionally, the domain expert assessed the quality of the conceptualization as it progressed through the iterations (each domain expert for the corresponding corpus). Finally, the design sub-team conducted user studies to evaluate the combined work, looking at how the human benefits from the machine and vice versa, exploring the challenges involved and the necessary functionality, and examining the usability of Fusion. We performed five design iterations across two case studies. Section 6 details the evaluation of the two case studies.

Research stage 3 initiates a deeper learning loop in which the lessons learned in the first *Build-Use-Evaluate* iterations inform not only the design of Fusion but also the HML configuration. In particular, we adjusted the work of the domain expert and the selection of qualitative techniques used. Moreover, additional types of feedback from the machine to the human were designed (the need for some of them came up only later in the process). At this stage, we also designed more functionality in Fusion, expanded the ML modules and external corpora, and improved usability.

In the final research stage 4, we generalized the design knowledge we gained about HML configurations and Fusion. The *lessons learned* through the DSR iterations about the functional level in Fusion are detailed in the Technical Artifact section (Section 5), and lessons about usability and the communicational level are summarized for the user studies in the Evaluation section (Section 6.1). In our discussion section, we synthesize four of these lessons learned as design principles that apply to our human actors [29].

Case Studies

To ground our work and guide the design, we used two case studies of text classification applied to the cyber security domain with the goal of detecting suspicious communication. These case studies were chosen at the beginning of the project and were motivated by our industry partner[2] specializing in Darknet communication analysis and threat detection. We used their API to collect the data. The data we used involved several inherent challenges—including strong data imbalance towards non-suspect messages, hard to distinguish language-use in both suspect and non-suspect groups, the use of slang, and the unstructured nature of the data. Each case study had an assigned dedicated domain expert throughout the iterative design process of Fusion (referred to as DE1 and DE2 respectively). These domain experts also participated in periodic user study sessions as part of our evaluation.

[1] https://shiny.rstudio.com/
[2] https://www.cybersixgill.com/

Case 1: Identifying Illicit Drug Transaction Messages in a Darknet Forum. Hidden Answers[3] is the Darknet version of Q&A sites such as Stack Exchange, Quora, Yahoo Answers, and Reddit, where users can post questions about effectively any topic (without censorship). This is a diverse forum with very broad themes, that operates in English, Spanish, Portuguese, and Russian. Allegedly, users can also ask crime related questions, such as *"Where can I buy drugs?"*, *"Which site is legitimate or not?"*, and *"Where to buy guns and fake ID's on the Deep Web?"* The interface is similar to Reddit's and Stack Exchange's interfaces, centered around questions organized by tags, and offers a search functionality. The classification goal in this case was to classify as 'suspect' messages that talk about usage, solicitation, selling, and purchasing of illegal drugs. We collected a total of 5,337 messages (containing questions, answers, and comments) for this case study from March 8th 2018 until April 25th 2018. At this point, the domain expert (DE1) built an initial conceptualization based on 513 messages: inspecting the messages, assigning categories to words and phrases, and generating a categorization. To avoid over-fitting, only when this process was done, the rest of the data (4,824 messages) were classified to serve as ground truth during the iterative process within Fusion. After removing messages that couldn't be classified, we ended up with 5,285 messages. Table 2 shows descriptive statistics of the collected data.

Case 2: Identifying Expert Hackers in a Cybercriminal Forum 'BitsHacking'. BitsHacking is an English-language cybercriminal Darknet forum operating since 2012. It is known as a one of the most popular carding sites, however, it also includes hacking and security, cracking, dump sharing, tutorials on various topics, and hacking competitions. The functionality offered by both Hidden Answers and BitsHacking is quite similar, however, whereas Hidden Answer's UI is similar to Reddit's interface, BitsHacking is more of a traditional forum. The classification goal in this case was to differentiate between expert hackers and amateur hackers. Specifically, to classify as 'suspect' messages indicating an expert hacker, i.e., a person with the knowledge and means for conducting a harmful attack. We collected a total of 3,242 messages (containing questions, answers, and comments) for this case study from December 28th 2019 until January 26th 2020.

Similarly to the previous case study, the domain expert (DE2) built an initial conceptualization based on 543 messages: inspecting the messages, assigning categories to words and phrases, and generating a categorization. However, due to data imbalance reflected by a low number of suspect messages, the domain expert increased the initial conceptualization data set to 1,575 messages (containing 59 suspect messages). To avoid over-fitting, the rest of the data (1,668 messages) were classified by an external domain expert (DE3) specializing in cyber-crime, to serve as ground truth during the iterative process within Fusion. During this process, both the domain expert assigned to this case and the external expert discussed the classification process, and discussed which examples they consider as suspect or not (as part of an inter-rater reliability protocol established ahead of time).

Table 2. Descriptive statistics of the collected data. Data size indicates the total number of messages in the corpus and the number of messages used to build the initial conceptualization. The class distribution indicates how many messages were *'true suspects'* out of the whole data set. Word count indicates the average number of words per message.

Case Study	Data Size	Class Distribution	Unique Users	Word Count
Hidden Answers	5,285 messages (513 initial conceptualization)	196 suspect (3.85%)	1,166	57.43 avg.
Bits Hacking	3,242 messages (1,575 initial conceptualization)	78 suspect (2.46%)	344	198.59 avg.

[3] Accessible through Onion or i2p link (e.g., hiddenanswers.i2p).

Domain Expert Descriptions: DE1 is a researcher specializing in qualitative data analysis and mixed methods research, with more than five years of experience in social discourse analysis and cyber ethnography. DE2 is a cybersecurity and web intelligence expert, specializing in identifying cyber terrorism and the use of Dark web and social media by non state actors. He is consulted by industry B2B cyber intelligence firms and previously served as an expert consultant on online violent extremism and radicalization for the Organization for Security and Co-operation in Europe (OSCE). The external domain expert (DE3), who assisted in case study 2, is a senior law enforcement officer and cyber intelligence analyst who specializes in analysis and monitoring of the Dark web.

4 THE HML CONFIGURATION

We propose a general HML configuration that is demonstrated through text classification aimed at suspect communication detection. The HML configuration is novel in integrating ML classification with human text analysis in such way that *allows reciprocal learning*. We first formulate it and later protoype and evaluate it in sections 3-6. The premise of our work is that a configuration of human and artificial intelligence will lead to better classification than AI alone. The goal of HML configuration that we study in this project is a combination of high classification accuracy and effective learning. In effect, learning is by doing, i.e., the classifier extends their classification knowledge in the act of attempting to classify new materials accurately. We characterize the configuration using four components: processes, task allocation and control, data, and classification knowledge. We further assume that the proposed configuration incorporates software to support the collaboration between human and artificial intelligences.

4.1 HML configuration processes

Figure 3 depicts an abstract view of the HML configuration processes that rely on the data and conceptualization to enable classification tasks. The processes are shown as a sequence of five interdependent steps. The steps are performed by one of the actors (a human domain expert) and a machine operating with a set of ML classification models. These ML models are designed and redesigned by a data scientist. The implementational aspects of the processes, the allocation and control of classification tasks, data models, and conceptualization, are further detailed in Section 5.

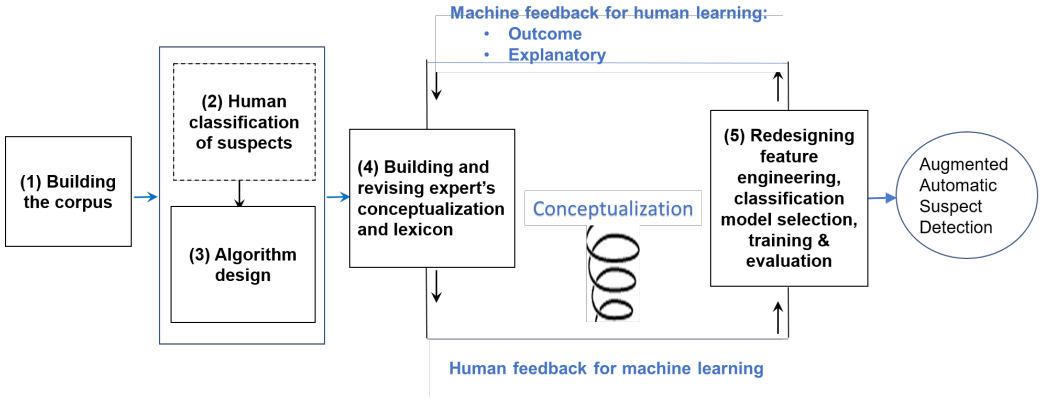


Fig. 3. A process diagram of the proposed HML configuration (adapted for suspect communication detection).

The left-hand side of Fig. 3 is an *initialization phase* and the right-hand side is the *continuous learning-feedback loop*. The initialization phase (steps 1-3) includes processes needed to set up the

HML configuration. The learning-feedback loop (steps 4 and 5) is ongoing, but pauses at some satisficing point to enable automatic suspect detection, and reactivates periodically to control quality and improve if need be. Initialization begins with setting up the corpus of textual communication, cleaning, and organizing the data, as input for human and machine processing. In our case, it involved mining data from the Darknet. In step 2, domain experts label the texts as either suspect or not. This process is necessary for most ML algorithms. In step 3, a set of ML models is created. Specifically, we considered two types of algorithms: supervised models, that use the labeling from step 2, and sentiment-scoring based models which we refer to as unsupervised [38, 46], that do not require labeling. Both types of algorithms are designed to be able to account for context in the learning loop.

The reciprocal-learning feedback loop is the centerpiece of the HML configuration. It helps to imagine two parallel learning processes occurring simultaneously: the machine learning and the human learning. Both rely on the same data but interpret and use context in different ways to reach a shared conceptualization of the corpus. In step 4, the domain expert creates a conceptual view of the human classification process, including a statement of criteria by which to classify (e.g., suspect vs. non-suspect). Conceptualization, as a process, entails making sense of the individual messages in the context of other messages and the corpus as a whole (as depicted in Fig. 2). In the process of sense-making, categories of content are formed and provide a stable structure to capture the constant stream of new messages yet sufficiently flexible to be adapted to new circumstances. Sense making entails not only integrating more information in context, but also examining alternative perspectives, which includes alternative classification models. The resulting conceptualization can therefore be seen, from the human perspective, as a temporary and presumptive view that can lead to action, and although organized, is sufficiently adaptable to enable learning.

In Step 5, the selected ML model is trained on the corpus, with possibly additional inputs (lexicon, external corpus). If necessary (e.g., unforeseen changes in the conceptualization), the algorithm is re-designed by the data scientist. The ML models are evaluated for their predictive accuracy, and when accuracy is satisfactory, the HML configuration will move to an automatic mode of detection with the appropriate classification model and given conceptualization. Generally, we envision an initial stage of intensive sessions in which a domain expert and the machine learn together how to classify, and then, once the system begins classifying new data automatically, more HML sessions will be initiated periodically to ensure continuous learning and assure classification quality.

The ML models generate feedback, both outcome and explanatory feedback, from which the domain expert can learn and improve the conceptualization. The feedback should therefore describe not only the classification accuracy (outcome feedback) but also the context that may explain or suggest why specific messages were (machine) classified falsely or truly (explanatory feedback). Armed with this information, the domain expert re-enters step 4 to revise the conceptualization. The revised conceptualization becomes input to ML models in step 5, to implement the new knowledge gained, and to retrain/redesign the ML models accordingly. In other words, the input from step 4 to step 5 can also be seen as feedback from the human to the machine. This loop demonstrates, for our case, how in the HML configuration the quality of classification depends on the reciprocal learning between human and machine. It can be seen as reinforcement of both machine learning and human learning, recognizing that machines and humans learn differently.

4.2 Task Allocation and control

Steps 4 (Human process) and 5 (Machine process) enable the required classification tasks as shown in Figures 1 and 2. In the proposed configuration, not only do the human and the machine perform the tasks allocated to them according to their relative advantages in attaining classification accuracy, but they also learn from each other to perform better in subsequent assignments so that high

accuracy should not compromise learning. Furthermore, the human-machine collaboration requires additional control tasks to ensure coordination and reciprocal learning. Table 3 allocates the classification tasks between human and artificial, and, for clarity, distinguishes the control tasks allocated to the human in a separate column. We also included tasks needed for the initialization (Steps 1 - 3). The tasks allocated to the human can be performed by domain experts or data scientists. In some cases, humans perform the task of labeling that serves as an input to ML supervised algorithms, and as ground truth for generating feedback, but we concentrated here on the generic tasks only.

Table 3. Initial task allocation in the proposed HML configuration.

Classification Tasks Allocated to Machine	**Classification Tasks Allocated to Human**	**Control Tasks Allocated to Human**
Organize corpus for efficient feature extraction	Build conceptualization	Control the initiation and flow of operations
Extend features with external corpora	Sense making, contextualization, perspective taking	Control when to pause (and resume) learning and classify automatically
Extract features	Qualitative content analysis to produce concepts	Labeling for ML training and validation
Operate classification models	Qualitative structural analysis to produce lexicon	Select and design ML classification models [Data Scientist]
Generate feedback to support human learning	Analyze false classifications and rare cases	

4.3 Data

The data (not shown in Fig. 3) include the content and context of messages organized as a corpus of human communications (e.g., text messages and posts) and external corpora that expand the context (e.g., Wikipedia).

4.4 Classification Knowledge

The third component is the conceptualization, which is central to the HML configuration and represents the classification knowledge learned by the human and machine. The conceptualization is, at an abstract level, shared by human and machine but, at a concrete level, represented in different forms with different formalisms. Furthermore, the choice of knowledge representations, such as decision trees or lexicons, may differ between human experts and machine classifications. When the domain expert uses qualitative content analysis, the emergent categories form part of the conceptualization. They can be represented to fit the particular ML classification models, e.g., as a hierarchy of categories, and separately, represented to fit the human, e.g., as a network of people or objects, an algorithm, or a decision tree. Ideally, while not always feasible, machine representations should fit human representations. In what follows, we limit the examples and implementation to a lexicon based representation of the conceptualization. We use (human) structural analysis to relate the conceptualization to the lexicon, which is then input to the ML classification models. We revisit conceptualization representation in the discussion.

The conceptualization serves the machine in two ways. First, it organizes the context for word embedding (metaphorically, the conceptualization "draws the machine's attention" to where it should seek related words). Second, the conceptualization defines new features (or relationships) that the ML model should consider.

To sum up, the HML configuration shown in Figure 3 is a combination of human intelligence and artificial intelligence in which each performs and learns classification tasks and, moreover,

learns from each other. The HML configuration is characterized by its processes, its specific task allocation, the data sets it processes, and the evolving conceptualization by which it classifies. *Fusion* is designed to cope with the volume and complexity of operating the HML configuration. The next three sections describe how we developed Fusion and used it to test the power of the HML configuration.

5 TECHNICAL ARTIFACT: FUSION

Fusion is a working prototype designed to support HML configured for classifying text. Other text classification systems exist, however, unlike Fusion they do not put human learning as a primary (business) goal, nor have an automatic feedback loop between the user and machine. These alternative systems can be categorized by their core capabilities: (1) *GUI-based model construction*—a user-centric approach in which the user outlines the data analysis process. Such tools essentially follow the state-of-the-art ML process (examples: deepcognition.ai, orangedatamining.com, rapidminer.com, dataiku.com); (2) *Data visualization (BI) capabilities* (examples: bigml.com, dataiku.com); and (3) *Automatic data-driven ML*—machine-centric approach with minimal user intervention, aimed at lower-tech users (examples: bigml.com, datarobot.com);

Fusion was developed following an agile methodology as described in Section 3, beginning from a small single-scenario prototype, then, iteratively, used, tested, and evaluated. We started with Case Study 1 (detection of suspected illegal drug activity messages) and expanded to Case Study 2 (identifying expert hackers).

In this section, we focus only on the key functionality of Fusion and four lessons learned for reciprocal learning. The first lesson is about the need to support the setup and manage the entire learning process shown in the HML-configuration process diagram (Fig. 3) according to the initial task allocation (Table 3). The other three lessons refer to the key components and mechanisms of reciprocal learning. In Section 7, we synthesize these lessons into generalizable design principles.

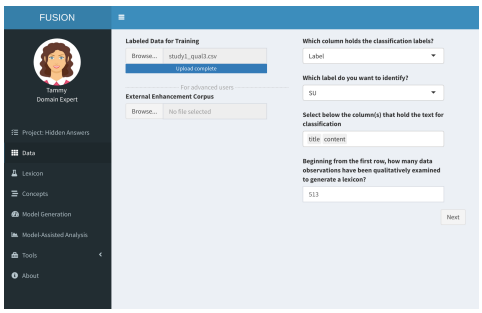
5.1 Facilitating the Process and Managing Task Allocation

Fusion's menu items seen on the left-side of the screenshots in Fig. 4 reflect the key functionality dictated by the HML-configuration processes (Section 4.1). The HML-configuration setup operations (steps 1-3) are enabled through menu item *'Data'* or performed offline and then imported into Fusion. *'Lexicon'* and *'Concepts'* menu items enable human operations of step 4, and *'Model Generation'* menu item facilitates the control task for selecting and designing ML classification for step 5. Lastly, *'Model-Assisted Analysis'* supports sense making, contextualization, and perspective taking based on machine feedback for human learning. These processes are best demonstrated by the screenshots from Case Study 1 (shown in Fig. 4). The screenshots exhibit the overall user interface and the general flow of the interaction.

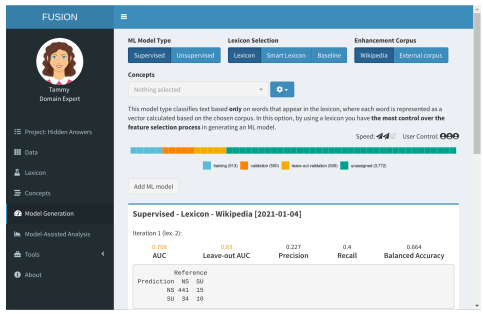
The users of Fusion are domain experts, who are unfamiliar with ML classification. They, as users, require support and guidance throughout the interaction with the system. Fusion provides a notification feature and a tabbed navigation menu, which is always visible, allowing users to navigate between steps and orient themselves quickly within the process. The user is further guided by Fusion when confirming all the previous steps were done successfully before moving to the next step. For example, (1) informing the user if no lexicon was uploaded when trying to proceed and in generating a lexicon-based model; and (2) checking and informing the user if the lexicon includes invalid entries, such as assigning the same term to multiple classes, which may hinder the next steps. Additional functionality to support and guide users included tip-tools to explain technical terms.

We attempted to allocate all data management tasks to Fusion. In particular, iterative classification requires the division of data into iteration-chunks and allocating the data to model training and

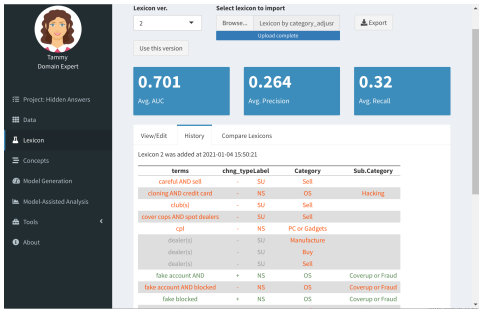
validation. For instance, if the domain expert inputs 5,285 data observations, and indicates that 513 of these were used for conceptualization and initial lexicon creation, these 513 will be assigned as the *training* subset. The *validation* and *leave-out* test sets will get 500 proportionally-randomly-chosen data points each. The leave-out test set stays constant throughout the iterations to help recognize situations of data over-fitting. The rest, 3,772 data points, will be temporarily put aside as *unassigned*. At the second iteration, the training set will grow to 1,013 data points (the original 513 + the 500 taken from the previous validation set), while a new validation set will be created by



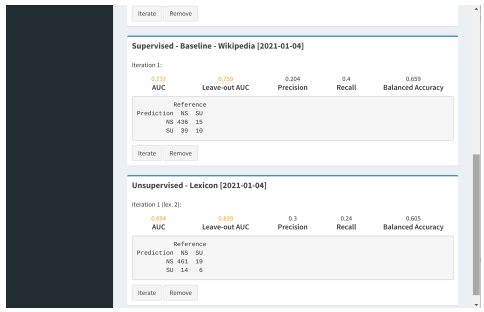
(a) Data input screen where the user configures the data input.



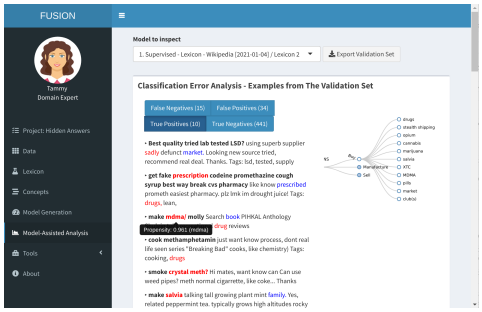
(b) Model selection and generation screen, providing a high level view of the existing models.



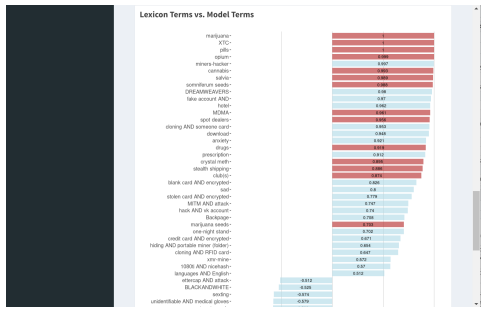
(c) Lexicon version management screen providing a 'diff' functionality between versions 1 and 2.



(d) The high level view allows easy comparison between models.



(e) Drill-down analysis of concrete text messages. On mouse hover, the users is shown the term propensity and the context of the word.



(f) A bar-chart visualizing how the domain expert assigned terms to classes vs. how the model assigned them, ordered by propensity.

Fig. 4. Screenshots of Fusion's user interface (based on data from case study 1).

taking another 500 proportionally-randomly-chosen data points. This process will repeat itself in each iteration step, as long as there is enough data in the unassigned subset or if new data is added. Fusion automates this necessary control process in a way that is transparent to the user.

5.2 The Need for Conceptualization That Serves Human and Machine Learning

Conceptualization (shown as a spiral in Fig. 3) serves as the knowledge repository needed in reciprocal learning for storing the accumulative knowledge, and at the same time, structures the learning. Fusion, at this stage of its development, represents the conceptualization of the HML configuration primarily as a lexicon that includes the domain expert's hierarchy of concepts (categories) and its mapping to word strings in the corpus. In essence, a lexicon is the mapping we use between the domain expert's conceptualization and a machine readable (and testable) input. Lexicon terms can be a word, a phrase of consecutive words, or patterns of words, such as *"X AND Y"*. Figure 5 shows an example of a lexicon (from Case Study 1), and Figure 4c shows the screen for managing lexicons, in which Fusion shows a history of changes (i.e., *diff*), for example, the pattern "careful AND sell" that had been part of the category 'Sell' in Lexicon version 1, was deleted from Lexicon version 2.

Labeling	Category	Sub-Category	Words	Phrases	Patterns
SU	Manufacture	*NA*	drugs; opium; cannabis; marijuana; salvia; XTC; MD...	papaver somniferum seeds; opium poppy seeds; m...	grow room AND hide
SU	Manufacture	Poison	cyanide	*NA*	*NA*
SU	Buy/Consume	*NA*	drugs; snort; opium; cannabis; hash; dabs; edibles; ...	stealth shipping; green lollipop; get high;	*NA*
SU	Buy/Consume	Fake Prescription	codeine; promethazine; oxy; xans	fake prescription	*NA*
SU	Sell	*NA*	drugs; opium; cannabis; marijuana; salvia; XTC; MD...	leave immediately; liquid L; crystalized L	[careful AND sell]; [cover cops AND spot dealers]; dr...
NS	OS	Hacking	computer; pc; fee; vulns; botnet; hacking;	Revenge Rat; text messages; magnetic card; MITM-...	[to rat x; x= people, phone]; [x comput...
NS	OS	Ransomware	RaaS	*NA*	*NA*
NS	OS	Counterfeit money	print; bills; note(s); supernotes	ATM notes; ATM skimming (device); black dollar	counterfeit; fake AND money
NS	OS	Coverup or Fraud	*NA*	*NA*	[fake x; x= id, numbers, account AND, blocked]; [x ...
NS	OS	Money laundering	*NA*	dirty money	*NA*
NS	OS	Self-harm	*NA*	*NA*	[Kill; hang AND yourself; myself; physical pain]
NS	Web	*NA*	steam; google; gmail; youtube; facebook; netflix; hb...	*NA*	[web; site AND account];
NS	Intimacy or Sex	*NA*	sex; sexting; relationship; girl(s); virgin; romance; h...	dating websites; one-night stand; casual encounters	*NA*
NS	PC or Gadgets	*NA*	programming; Python; javascript; cpl; iphone; andro...	imsi catcher; open source	*NA*
NS	Money making or Investing	*NA*	regulation; database; forex; transfer; trading	*NA*	[1080ti AND nicehash; mining]; [crypto {0.01} AND ...
NS	Purchasing	*NA*	hotel; book; worried; trading; stock;	weed edibles	*NA*
NS	Education	*NA*	SAT; homework;	*NA*	*NA*
NS	Public commentsphere	*NA*	atheist; God; sacred; Krishna; aliens; prison; love; tr...	*NA*	[languages AND Swedish; English; Danish; Norwegia...
NS	Personal issues	*NA*	lie; truth; family	*NA*	*NA*
NS	Sleeping disorder	*NA*	prescription; meditation	*NA*	*NA*
NS	Mental disorder	*NA*	depressed; anxiety; fear; afraid; prescribed; sad; vit...	anti-depressant;	*NA*

Fig. 5. An example of a lexicon from Case Study 1 (not shown are optional columns such as website names, website links, etc.)

5.3 The Need for Multiple Perspectives in Reciprocal Learning

Offering multiple perspectives, even when marginal change is introduced in each loop, is a central mechanism of reciprocal learning. When generating an ML model, Fusion gives the user a choice in classification methods and guides the user in this process. On the model generation screen, the user is shown model-selection buttons (see Fig. 6), each one allows to choose a characteristic of the desired model.

Fig. 6. Model selection characteristics offered in Fusion.

The possible ML model characteristics offered in Fusion are as follows:

Machine Learning Model Type: Supervised or Unsupervised.

- Supervised model types are machine-learning based models, that use a portion of the labeled data for training (learning), and then based on that learning, these models apply it to generate classifications on new data. The ML models offered in Fusion employ word2vec and lasso regression via `glmnet`[4] to generate the model.
- Unsupervised model types in Fusion refer to sentiment scoring method [46], implemented as follows: classify a message by examining how many terms (words, phrases) from the message are labeled 'suspect' vs. 'non-suspect', normalized by the size of the lexicon. These unsupervised models don't use or require labeled data for training.

Lexicon Selection: Lexicon, Smart-Lexicon, or Baseline.

- The baseline model type doesn't rely on a lexicon file. It classifies text by considering all the words from the text (excluding stop words), where each word is represented as an embedding vector calculated based on the chosen enhancement corpus. In this option, the user has no control over the "feature selection" in generating an ML model.
- Lexicon and Smart-lexicon options use a lexicon file for feature selection. In essence, they "focus the machine's attention" to important words and filter out the noise. As opposed to a non-lexicon based type (baseline), using a lexicon gives the user control over the feature selection process, more so in the Lexicon type vs. the Smart-lexicon type. The difference between Lexicon and Smart-lexicon options, is in the preprocessing of the lexicon. The Lexicon model types use the lexicon 'as given', while the Smart-lexicon models first enhance the lexicon with semantically-similar words (based on their embeddings) and then generate the model.

Enhancement Corpus: Wikipedia or Other.

- Fusion uses a built-in enhancement corpus based on a portion of the English version of Wikipedia, it is small (at the size of 100MB), but surprisingly is quite comprehensive, and works well in many cases. For specialized cases where specific terminology is used, for instance, if the text comes from online forums prevalent with slang, it may be better for the user to provide a custom enhancement corpus (e.g., built by collecting texts that use a similar slang). In the two case studies described in this paper, we did not use a custom enhancement corpus, but the option to do so exists within Fusion.

Together, these characteristic selections indicate the type of model to be generated. In total, there are currently five possible model types, each with their own characteristics and unique approaches for classifying text: (1) *unsupervised-lexicon*, (2) *unsupervised-smart-lexicon*, (3) *supervised-lexicon*, (4) *supervised-smart-lexicon*, and (5) *supervised-baseline*. In selecting certain classification model characteristics, the user controls the **effect of human conceptualization on the machine's feature engineering**, and the **perspective it grants the user**. For instance, an *unsupervised-lexicon* model type relies completely on the domain expert's conceptualization for deciding the classification, therefore, it helps examine the current human conceptualization 'as-is', i.e., examine whether it captures the text as intended or not. On the other hand, a *supervised-smart-lexicon* model type uses machine learning to both expand the lexicon with semantically-similar new terms and to decide how the lexicon terms affect the classification. This grants a way for providing the domain expert with recommendations on ways to expand their conceptualization. By taking advantage of alternative model types the user can learn different lessons to improve their own conceptualization.

[4] https://glmnet.stanford.edu/articles/glmnet.html

5.4 The Importance of Explainability for Contextualization and Explanatory Feedback

In Section 2, we talked about the importance of interpretability, which is the ability to interpret machine algorithms in an understandable way to humans, and the tradeoff between model interpretability and accuracy. More recently, researchers began exploring explainability and explainable AI (XAI) techniques [8, 11, 20, 51, 60]. For us, explainability support was critical in Fusion. We explored existing methods and initially implemented it by using the Local Interpretable Model-Agnostic Explanations (LIME[5]) software package [49]. However, as Fusion evolved, we faced a challenge relating to using LIME—the software package was impractical for the large data sets we used, consisting of 500 text messages at a time (using LIME for more than 10 messages would work extremely slowly or would crash due to high memory-consumption, hindering interactivity).

For this reason, we developed our own explainability component, that in essence uses the generated model to predict term propensity (for N-grams and skip-grams), and then another component that colors the words (unigrams), phrases (N-grams), and patterns (skip-grams), that the model used for classification, based on term propensity (coloring in red for higher propensity towards 'suspect', and in blue for higher propensity towards 'non-suspect'). This solved the aforementioned issue we had, and allowed us to control and customize the feedback provided to the user. We found that information overload was a concern—communicating too many details about how propensity was calculated or the inner-workings of the model, confused the user. To reach the right level of details, we began by only showing colored words (either in red or blue). Then, based on observations from the user studies, we gradually added relative propensity scores per word, which helped communicate how the model reached the final classification, and added context to the coloring, showing which phrases or patterns (skip-grams) the colored word is part of. This is an example of explanatory feedback.

More generally, explainability is essential for feedback from the machine to the human expert in order to enable the understanding necessary for effective reciprocal learning, as described in Section 2. Feedback is effective when it is given and understood *within context* and *can be used* in the context to improve the learning process and its outcomes over time. Fusion provides either outcome or explanatory feedback depending on the user's motivation and goal. Explanatory ("drill-down") feedback is characterized by detailed contextual information. Outcome ("birds eye") feedback is in the form of performance metrics (e.g., AUC, recall, precision, accuracy) and is used when managing lexicons (shown in Fig. 4c) or when creating models (shown in Fig. 4b). This allows to quickly evaluate and make decisions about the performance of the model and to compare with other co-existing models (Fig. 4d).

The explanatory feedback is used in the sense-making process and allows to 'drill-down' on specific cases and terms, given within context. One example of explanatory feedback provided by Fusion is colorized terms within a text message based on how they affected the model's classification. Fusion does this within the data being analyzed (not in an isolated view of terms alone), which gives the domain expert the opportunity to reflect on and analyze terms and categories while preserving the context these terms were used in. Moreover, as part of the low level analysis and the ongoing conceptualization process, Fusion also provides domain experts with an interactive visual representation of their categorization, as a way to increase context and provide cognitive support. For example, Figure 4e demonstrates three instances of context provided to the user. The marked word *'mdma'* is shown in its linguistic context, i.e., the words around it in all the messages it was used to classify. Additionally, the user can see (by hovering over *'mdma'*) the term's propensity to classify the message and the other lexicon terms that affected this propensity score. Finally, the interactive category-tree, shown on the right-hand side of Figure 4e, shows the term as part of the

[5] https://cran.r-project.org/web/packages/lime/index.html

user's mental model of their conceptualization of the meaning space, as opposed to the linguistic space of the corpus.

This low level analysis is further enhanced by giving the user the ability to filter and focus on specific groups of terms or messages (e.g., only messages that were false negative classifications), helping to gain new insights that were otherwise hard to see within the high volume of raw data. We also take advantage of the 'linking and brushing' interaction technique [10] commonly used in the information visualization field, allowing users to select lexicon terms in the bar-chart visualization, and have it filter the drill-down text analysis component, to only show text messages containing those terms. For effective communication between the expert and the machine, the differences between the user's assignments of words in the lexicon and the machine's use of the same words in the classification model must be made explicit. This comparison is presented in Figure 4f. For example, the term 'miners-hacker' (5th from the top in the figure) receives a positive propensity towards 'suspect' from the supervised ML model, while its blue color indicates that the domain expert assigned it as a non-suspect. Lastly, the provision of effective feedback is not enough, the user needs to act on it and enact a positive change in classification performance. This meant that when designing Fusion and the feedback it provides, we needed to also consider how to instill the user with trust in the system. We approached this by communicating transparently about the process and giving the user a sense of involvement in the process.

The features described above evolved as a result of functional-level lessons learned from the iterative DSR process we followed. In the next section, we share lessons about usability and the communicational-level as part of a formative evaluation.

6 EVALUATION

We evaluated the application of Fusion in two case studies with their respective domain experts. For clarity, we distinguish here between two types of evaluations: Formative evaluation that provides feedback on functionality and usability intended for Fusion's developers (stage 2c in Table 1); and Summative evaluation that assesses the accuracy of classification models, and assesses the quality of the conceptualization. For each case, we conducted two rounds of user studies, five months apart. Participants were instructed to use Fusion to classify text and improve their conceptualization. These sessions, mainly by Zoom, lasted 2-5 hours with each domain expert. Audio and screen-capture were recorded, and participants were asked to think-aloud. We then conducted a semi-structured interview around the classification task and the benefits of and difficulties in using Fusion. The interview scripts for both rounds are shown in Appendices B and C.

6.1 Formative evaluation of functionality and usability

The first round of user studies focused on usability difficulties and the second on the expert's actions in reciprocal learning, working with the lexicon, and processing the feedback generated by the ML model's output. For simplicity, we combine the lessons learned from both rounds.

Control, guidance, terminology: Overall, the user study participants were satisfied that Fusion supports the entire workflow of a single iteration, from data input, lexicon input, and model generation, to a detailed analysis of lexicon terms and misclassified messages. Both participants encountered difficulties in using the various functionalities, indicating a need for improvements in guidance, better error checking, and overall better feedback on their input and better explanations on the computer output (e.g., notifying the user of words that were assigned to both classes in the lexicon file, suspect and non-suspect). DE2 wanted a step-by-step guidance indicating how far you are in the process and wanted to better understand the process flow (e.g., why lexicon input was not part of data input). DE1 wanted to know which parts of the data input were considered by the

ML models, and wanted better explanations on the performance metrics provided by the model. To overcome some of these issues, DE1 made use of visual cues from Fusion, such as the color coding of AUC values and the four-colored bar data visualization (shown in Fig. 4b).

Feedback from ML models: The most pressing set of observed difficulties were tied to the inadequate explainability of the feedback, both high level feedback and the low-level, specific, feedback. DE1 found it useful to see *concrete* text messages analyzed at the word level (see Fig. 4e). Moreover, DE1 was impressed that he can use "brushing" (i.e., selection and filtering) on the bar chart which affected the low-level analysis as well. However, at the same time, both DE1 and DE2 needed time to adjust to the color coding of words, red as being more indicative of 'suspect' and blue as more indicative of 'non-suspect', specifically what each color represented and why it was colored that way. At the time, we only colored the words, without providing any context to the coloring—something we addressed by the second round of user studies. Interestingly, when asked to find improvements to the classification of messages, DE1 focused mainly on words that were colored, i.e., words that already appeared in the lexicon, while DE2 first explored the whole text message, and then examined the colored words.

Another key challenge the domain experts faced was the dissonance between how they assigned a term in their lexicon and how the machine used it to classify. For example, DE1 used a supervised lexicon-based model and saw that some terms such as "Marijuana seeds" were given a high propensity score for 'non-suspect' by the model, while they marked this term as 'suspect' in their lexicon. This confused DE1, as they wanted to know how this "mistake" had happened. DE2 also encountered this dissonance, and described it as a *"nice to have feature"*—explaining that by looking at unexpected propensity scores, it may trigger them to examine how and why the machine assigned these scores, either by examining all text messages that contained the specific term, or by using the term-frequency aggregated table.

We learned from the domain expert feedback and addressed some of the issues between user study rounds. We added components that provide the necessary context for low level analysis and improved cognitive support. Two examples of this are: (1) adding an interactive category-tree (shown on the right side of Fig. 4e), that is as close as possible to the mental model the domain expert built his conceptualization around; and (2) by hovering on colored terms, Fusion shows which lexicon-terms affected the propensity score of the selected term, thus helping domain experts see the rationale behind it.

Working with the data and lexicon: Aside from the difficulties in processing the machine feedback, participants also showed difficulty when working on subsequent iterations of conceptualization: examining new data and adjusting the lexicon. For instance, for the domain experts it was difficult to find (from the corpus) the context in which specific classifications were originally made, or the context that led to assigning certain terms to specific sub-categories. Similarly, DE2 wanted an easier way to see under which sub-category in the lexicon the currently selected term is, without manually opening the interactive category tree (which can be large). DE2 also suggested additional new features that could be helpful, e.g., *"I'm interested to take all the messages we classified as 'suspect', and by using the sub-categories we have in the lexicon, to see cross-patterns between the 'suspect' messages"*.

Additionally, we gained a better understanding of the low level iterative analysis step. We now see that this step requires more time for rethinking and working on the conceptualization. The domain expert needs time to reflect, re-examine, and think things over. Sometimes the domain expert may need to go back and re-examine previous categories or terms in the lexicon, and remind themselves of the reasoning behind it. This is a comprehensive process, that requires both, the software support from Fusion, and (parts of) the original conceptualization process. We also saw

that there is a need to provide the domain expert with additional context about why they labeled certain words in a certain way. Lastly, we believe that there is a need to instill the user with trust in the machine, as we observed from DE1's statements in the user studies.

Overall, the interaction and workflow between the domain experts and AI seems to have been successful. *"I better understand now where Fusion can help me, to simplify things for me, to shine a light to places I'm not sure I would have looked at. I think I know now better where I need to look, beyond the confusion matrix values we see on screen. I understand better the importance of model-assisted analysis. I see it as a tool that improves the lexicon, from the perspective of someone who builds the lexicon"* [DE2].

6.2 Evaluating Classification Accuracy

Table 4 shows the performance metrics of both case studies across different model types, organized by iteration number. Note, typically a domain expert will not run all these model types within a single iteration in Fusion, but rather choose a model based on *utility*. Between iterations, the lexicon is revisited and improved by the domain expert and a new validation set is generated. However, the data set used for calculating *'leave-out AUC'* does not change between iterations. To make the results shown in in Table 4 even more comparable, we set a lower bound of 0.4 on the recall value, which allows readers to compare models and iterations more easily using the precision value. In two instances, there was no feasible solution found for this lower-bounded recall value. The domain expert, as a user of Fusion, can also set a lower boundary for recall, and then using precision compare and consider the "cost" of reaching this recall value. All the supervised models shown, including the baseline model, used 10-fold cross validation to fit the model and synthetic minority oversampling technique (SMOTE) [13] to help overcome class imbalance. Moreover, while Fusion's code and algorithms evolved over time during the project, the values shown in Table 4 were generated on the same version of code for comparison reasons (thus, not identical to values generated during user studies which were conducted at various points in time). From the results, we can see that (1) there is a clear ability to improve between iterations, (2) there is no a priori way to know which model will perform best, and (3) our approach revealed that the data is not stationary in case study 2 (seen by the values of 'unsupervised'), yet, the problem posed by this phenomenon is mitigated when we use lexicon-based approaches.

6.3 Evaluating The Conceptualization

In order to evaluate the conceptualization, we need to isolate and examine the improvement in lexicon performance over time. A lexicon is the mapping we use between the domain expert's conceptualization and a machine readable (and testable) input (an example is shown in Fig. 5). Conceptualization evaluation can be achieved by examining the *unsupervised-lexicon* model's *leave-out AUC* values across different iterations. An unsupervised-lexicon model within Fusion is a deterministic model that classifies text purely based on lexicon terms. The leave-out data set stays constant between iterations. Thus, by combining both, we control for all the variables except the lexicon. By looking at Table 4, we can see the *leave-out AUC* values from the unsupervised-lexicon models are increasing across iterations: 0.767⇒0.767⇒0.896 in case study 1, and 0.755⇒0.785 in case study 2.

To further investigate whether the conceptualization is improving over time, we approached it from a qualitative perspective, and focused on it during user study sessions, especially during the second round of user studies. While we directly asked participants *"Do you feel that your conceptualization is better or worse than two iterations ago? How? and Why?"*, participants also shared their thoughts throughout the user studies. DE1 replied that *"Yes, of course. As the sampling increases (through iterations), a qualitative domain expert, enriches the knowledge accumulated from*

Table 4. Performance metrics across different model types and iterations. Highest values within an iteration are highlighted in bold font. Two instances where no feasible solution was found (for lower-bounded recall value) are marked with *'nfs'*. In one instance, the Lasso regression did not converge [23]. It is based on the Newton algorithm, which computes the gradient descent of the loss function at each iteration. When the loss function has no derivative (usually in cases of data-overfitting, where loss = 0), the algorithm cannot converge.

Case Study	Iteration	Model	AUC	Leave-out AUC	Precision	Recall	Balanced Accuracy
Hidden Answers	1	Baseline	0.767	**0.874**	0.068	0.4	0.615
		Unsupervised Lexicon	0.807	0.767	0.345	**0.667**	**0.814**
		Unsupervised Smart Lexicon	0.795	0.779	0.333	0.4	0.688
		Supervised Lexicon	0.754	0.793	0.571	0.533	0.76
		Supervised Smart Lexicon	**0.879**	0.869	**0.6**	0.4	0.696
	2	Baseline	**0.806**	0.833	0.19	0.421	0.675
		Unsupervised Lexicon	0.754	0.767	0.25	**0.526**	**0.732**
		Unsupervised Smart Lexicon	0.744	0.793	0.167	0.158^{nfs}	0.563
		Supervised Lexicon	0.764	0.812	0.308	0.421	0.692
		Supervised Smart Lexicon	0.729	**0.865**	**0.364**	0.421	0.696
	3	Baseline	0.824	0.795	0.4	0.421	0.698
		Unsupervised Lexicon	**0.901**	**0.896**	**0.5**	**0.737**	**0.854**
		Unsupervised Smart Lexicon	0.841	0.856	0.085	0.211^{nfs}	0.561
		Supervised Lexicon	0.857	0.862	**0.5**	0.421	0.702
		Supervised Smart Lexicon	0.879	0.864	0.471	0.421	0.701
BitsHacking	1	Baseline	**0.946**	0.776	**0.103**	0.429	**0.688**
		Unsupervised Lexicon	0.799	0.755	0.042	0.429	0.645
		Unsupervised Smart Lexicon	0.82	**0.932**	0.065	0.429	0.671
		Supervised Lexicon	0.901	0.858	0.047	0.429	0.652
		Supervised Smart Lexicon	0.819	0.802	0.064	0.429	0.67
	2	Baseline	0.478	0.727	0.008	0.5	0.473
		Unsupervised Lexicon	0.633	0.785	0.011	0.5	0.541
		Unsupervised Smart Lexicon	**0.786**	0.928	0.05	0.5	0.707
		Supervised Lexicon	0.63	**0.989**	**0.15**	0.5	**0.737**
		Supervised Smart Lexicon	(Lasso regression did not converge on this data set)				

the coded data. So, it is likely that both the lexicon will grow, as happened to me, and the likelihood of developing 'concepts' for certain data will increase." Similarly, DE2 agreed by saying *"Of course the lexicon has improved. The fact that I added words and phrases (to the lexicon). There were posts that revealed new dimensions that weren't there before. Which makes sense, forums are dynamic."* DE2 further elaborated on how and why they felt their conceptualization has improved, *" It also gave me a lot of confidence by the way, I think the results of the lexicon were nice in all performance metrics we examined. This proves the iterative method, the learning loop. The more iterations I do, more likely I'll enrich the lexicon more."*

7 DISCUSSION

In this section, we first discuss human-machine learning configurations and raise questions on the role of humans versus machines in reciprocal learning. Next, we examine the challenge of capturing and representing the conceptualization knowledge, an area we feel has great potential. We also discuss methodological limitations that may affect the generalizability of our results. Finally, we provide design principles aimed at researchers and practitioners. These principles, which emerged from our theoretical framework as well as our experience in developing Fusion, provide guidelines for designing computerized systems to *support* collaboration for intelligent tasks, such as text classification, decision making and design.

7.1 Human-Machine Learning Configuration

We started out with a vision of human-machine collaboration that keeps the human in the learning loop [59]. In the HML configuration, both the human and the machine, not only combine their

intelligent advantages to perform better, but also learn from each other to improve future performance. Thus, the loop with reciprocal learning in Figure 3 is the centerpiece of the configuration. In reciprocal learning, the dialog between partners leads to shared meaning and sense making, particularly through contextualization and perspective taking, and, at the same time, acts as a self-reinforcing mechanism [34]. Interestingly, in our second case study, DE2 commented that the interaction with the machine built confidence in his classification. Additionally, in the HML configuration, the allocation of tasks is meant to capitalize on the relative advantages of human versus machine in performing and learning classification tasks so that the reinforcing feedback from machine to human complements human cognitive abilities (and vice versa).

Our evaluations of using Fusion in the two forums, Hidden Answers and BitsHacking, demonstrate the effect of learning on classification accuracy and conceptualization quality. In both forums, the general increase in classification accuracy of the supervised lexicon-based models shown in Table 4 indicates that learning improved accuracy from one iteration to another. This can be seen by contrasting the lexicon-based models (Leave-out AUC column) with the baseline ML-only approach as commonly implemented [42]. Furthermore, the accuracy of the unsupervised-lexicon (a proxy for conceptualization quality) also increased with iterations. In parallel, the domain experts reported (in the user studies) their subjective assessment of improvement. Interestingly, the experts used different sources of feedback from the machine to decide how to improve the lexicon. DE1 chose to attend to feedback on words already in the lexicon, while DE2 first explored the whole text message for possible words that he had neglected. Regardless of the particular learning style, both experts received reinforcement from the machine that would be infeasible (or extremely difficult) for them to generate. It may be beneficial in future to explore HML configurations that adapt to the user's learning style.

In past research, the idea of a configuration of human and machine to perform jointly, has been advanced at a *macro level* that addresses organizing, control, responsibility, and work implications [63]. We on the other hand, take a more *micro view* of how the configuration operates, i.e., the allocation of tasks and the communication within the configuration—a necessary level for making these configurations operational. We concentrated on learning and sense-making, but we believe our conclusions also apply to other forms of intelligent collaboration, such as joint decision making and design.

Several researchers [31, 58] have argued that the combination of human and artificial intelligence requires a *distinct analysis*. For example, Marcellino *et al.* [39] recently included human judgment to complement advanced automatic text classification to detect online interference on Twitter. We agree with them and offer a new approach to developing such collaborations that include a concurrent analysis at both the functional and the communication levels to achieve a productive collaboration, as we demonstrated in this paper.

Inspired by other researchers looking into how human-centered approaches affect machine learning [25, 59], we re-iterate the following two questions: (1) What role do humans play in HML configurations? and (2) Do HML configurations change the way in which machine learning is being done, if so, how? We believe that our work helps answer the first question and takes a first step towards answering the second question. By defining and evaluating the allocation of tasks within an HML configuration, we can better understand and define the role of human experts working collaboratively with, and learning from, artificial intelligence.

Our vision for the HML configuration as a conceptual framing for any human-artificial collaboration, in which both human and machine can learn from each other, goes beyond communication analysis. We have provided examples of applying HML in text classification, however this approach can be generalized to other cases and domains, such as the detection of bullying and intimidation online, an analysis of political messaging or fake news, and the analysis of medical imaging. For

instance, screening mammography is a well established mechanism used globally for early detection of breast cancer, but due to the vast number of performed mammography studies and insufficient number of trained radiologists, there is a lack of experts to handle the increasing workloads. AI based screening systems can achieve higher accuracy in interpreting mammograms than trained radiologists [40]. More importantly, an HML configuration would enable periodic learning sessions in which radiologist and system update their diagnostic knowledge.

7.2 Forms of Conceptualization

Reciprocal learning both builds and draws from the classification knowledge represented by the conceptualization. Conceptualization acts as the memory necessary for learning, not only to store knowledge, but also to structure the acquisition of new knowledge. At an abstract level, it is shared by human and machine, but at a concrete level, it is represented in different forms with different formalisms. It can therefore be seen as memory that supports distributed cognition [9]. A major design challenge is to provide the functionality of a generative memory for effective cognition distributed between human and machine. Boland *et al.* [9] see conceptualization as a temporary and presumptive view that can lead to action, and although organized, is sufficiently adaptable to enable learning.

Currently in Fusion, the classification-knowledge is represented primarily as a hierarchy of concepts (categories), derived from the process of qualitative content analysis and mapped to the corpus words through a lexicon. This lexicon becomes an input to the ML classification models that use it to classify new text messages (step 5 in the learning loop), and then feedback on it is provided back to the human expert, who can further augment the lexicon (step 4). However, this may not necessarily capture or map all of the human expert's conceptualization knowledge—other mappings could be possible or needed. In fact, we have began to experiment with another form of mapping the human expert's conceptualization, represented as a set of rules that capture the expert's classification criteria, e.g., when looking for expert hackers, a rule to disregard messages in the forum that in some way disclose the sender's identity. These rules do not fit into a category-based lexicon, and therefore require alternative structures to feed into the ML models.

We encountered this when the domain expert from case study 2, described his decision process and mental model, and represented part of his classification knowledge as a decision tree. Further development is needed in order to represent the richer forms of knowledge and their mapping (richer than a lexicon), to serve both as input to the ML models and as a working memory for the human expert to incorporate the feedback from the machine. In sum, there is a design challenge in providing the functionality for a generative memory, much like the multiple modes of information processing available in human memory [26]. Side by side with the new functionality, we must provide and support effective communication between human and machine so that new forms of generative memory processed by the machine must be explainable to the human. We believe that visualization of the conceptualization will also become a significant area of research for HML configurations [60].

7.3 Research Contributions

Several human-AI configurations that keep the human in the loop beyond the stage of machine training have been proposed to ensure the successful operation of intelligent systems [31, 55]. Our research takes one step further by analyzing how such configurations should be designed, conceptually and operationally. Concentrating on keeping the human in the *learning* loop [59], we first specified a theory-based conceptual artifact [3], *the HML configuration*, for continuous reciprocal learning. This contribution enabled us to then develop a technical (information systems) artifact. *Fusion*, which supports the human-machine communication and visualization necessary for

reciprocal learning in practice, implements the specification of the HML configuration based on the functional and communicational analyses that explain how the configuration can be adapted effectively to specific situations. The HML configuration can be seen as an extension of Suchman's [63] idea of human-machine configurations and a new general solution to how humans and machines can learn reciprocally, posing two measurable targets, machine learning and human learning. As a result, two new streams of research are currently building on and utilizing our HML configuration to guide human-machine collaborations, one for examining the phenomenon of online religious influencers and another for identifying expert hackers within dedicated online hacker communities on Darknet.

A second, more practical, contribution is the lessons learnt from building the technical artifact. Based on these lessons, we formulated design principles that can be useful when designing systems that support reciprocal learning in order to ease the human-machine communication and facilitate an effective allocation of tasks. Together with the HML configuration, these design principles and lessons learned form a *roadmap* for others engaged in designing information systems to support reciprocal learning, outlining pitfalls and suggested strategies to overcome them.

7.4 Limitations and Future Work

Several methodological limitations may affect the generalizability of our results. As we worked on text classification related to cybersecurity of data taken from the Darknet, our proposed HML configuration may need to be adapted to other joint-task contexts, particularly contexts in which human communication is open among known friends or colleagues. Furthermore, each domain requires its expert classifiers, who may differ one from another. Indeed, we relied, for each case study, on specialized experts to provide their subjective judgment during the initial data classification (labeling) and conceptualization phases. Researchers in augmented text classification have noted the threats to validity in employing human judgment due to duplication of bias and low inter-rater reliability when establishing ground truth [21]. We used recommended methods of qualitative analysis [12, 16, 17] to overcome inconsistencies and biases in creating the conceptualization of classification knowledge, and we added an external expert to increase reliability and minimize bias. Nevertheless, we are seeking ways to follow up on a sample of the suspects to compare expert predictions with cases that were subsequently identified in reality.

Our use of machine learning algorithms was not exhaustive, we explored the most suitable approaches (bag-of-words, word2vec). We are currently exploring the transition to Google's BERT, and, similarly, plan to experiment with others in the future. Additionally, we are developing new forms of conceptualization (beyond the lexicon and rule based). We believe explainability of the conceptualization and machine learning algorithms is a promising future avenue for advancements. Finally, we proposed a general HML configuration that we hope others will apply in other domains in which human experts learn from machines and vice versa.

7.5 Design Principles for Supporting Human-Machine Reciprocal Learning

We generalize our insights from the design-research process as principles for designing systems that support human-machine reciprocal learning. We chose four challenges that were tackled in the development of Fusion, which we believe are useful lessons for the design of human-machine learning systems generally. We decided to concentrate only on design challenges that involved human actors because they are different from the more algorithmic designs for non-human actors [59], and because we feel we have more insights to contribute. The design principles to overcome the challenges are: (1) iterative reciprocal learning with a shared conceptualization, (2) feedback showing changing perspectives, (3) feedback showing changing contexts, and (4) explainability of feedback.

Table 5. Proposed design principles for developing HML support systems.

Design principle	Explanation	Operationalization
1. System should support **iterative**, **continuous**, **accumulative** learning represented in a conceptualization accessible by machine and by human	HML configuration requires learning by machine and human. Supporting human learning requires iterations with limited new information at each stage, the learning should be able to continue as long as the human is in the learning loop. Similarly, machine learning should be iterative and accumulative. The accumulated knowledge should be represented in a memory, namely the conceptualization, that is accessible to both human and machine.	**Functional** (a) Knowledge accumulates in a visible conceptualization; (b) Incremental learning adds small chunks of information per iteration; (c) Human-computer interaction feedback directs and restricts action at each iteration; (d) Accessible logs of progress across iterations; (e) Ability to navigate across iterations. **Communication** (f) Conceptualization presented to fit the human expert's mental model
2. System should support **examining alternative perspectives in feedback**	Taking and examining **alternative perspectives** is essential for effective sense making. In the HML configuration, each ML model generates a perspective that is communicated through the machine feedback to the human with its unique added value. Perspective taking encourages exploration of new contexts through a new lens.	**Functional** (a) Fusion allows the user to add new model types at any point of the process and iteration, subject only to ML rules for avoiding overfitting. This functionality enables differential learning in supervised vs. unsupervised models; (b) Presenting the differences between models in terms of features used and accuracy achieved. **Communication** (c) Feedback on alternative models should fit the user's mental model.
3. System should support **examining changing contexts with bi-directional feedback**	In HML configurations, reciprocal learning between human and machine is supported with bi-directional feedback that shifts attention to alternative and changing contexts. **Explanatory feedback to the human expert** guides the expert to parts of the conceptualization that were more or less effective, directing the next learning iteration. Additionally, the outcome feedback assesses the individual's own learning. **Feedback to the machine** (qualitative reinforcement) is provided via the expert's revised conceptualization that (a) points the machine to promising parts of context at which text should be analyzed, suggesting, possibly, new features or relationships that should be considered in ML classification, and (b) expands the linguistic context of words.	**Functional** (a) Outcome (high level) feedback in the form of classification accuracy measures; (b) Explanatory (specific) feedback at the *'message level'*; (c) Explanatory feedback organized by alternative dimensions (e.g., false vs. true classifications) **Communication** (d) Feedback to human indicates the mapping between human and machine conceptualization; (e) Feedback to human indicates how a change in human conceptualization is modeled by the machine
4. System should provide **explainability** in feedback from machine to human.	**Explainability ensures effective communication** so that the human expert understands the feedback and the reasoning behind it. This principle singles out the need to understand the machine in order to learn from it.	**Communication** (a) Coloring the words in a message that impact the classification, showing their propensity towards the assigned classification as 'suspect' (red) or 'non-suspect' (blue).

As we wish to generalize our insights beyond our experience with Fusion, we define the overarching design goal for all four principles as facilitating effective reciprocal learning. The common context is the HML configuration. Detailed explanations of how these challenges surfaced and of the design solutions in Fusion are provided in Sections 5.2-5.4. Fusion is best viewed here as an exemplar showing the context and rationale of the more general principles. Each solution in Fusion corresponds to a learning mechanism in our HML configuration but adapted according to the requirements made by the experts, who wished to learn and perform better. Table 5 explains each principle and its application to Fusion at the functional and communication levels. As noted above, the functional level refers to what functionality should be provided by human or by machine, and the communication level refers to how information should be presented to ensure effective communication. We established the latter mainly from miscommunications we observed in the user studies. As Fusion is still under development, additional functionality will be developed to fully materialize the four design principles.

We see these design principles as a major contribution of our research. In a way, the table can be read from right to left, starting with the concrete functionality supplied according to the domain experts' requirements, and then generalizing to abstract principles that capture the essence of supporting learning, regardless of the specific task of text classification. As suggested, this is not a

complete list of lessons learned. Other design principles will be added as we gain more experience and expand the support for the range of tasks and corresponding task allocations, shown in Table 3, to include for example, control transfer from human to machine.

We believe the four principles can be generalized to other systems that leverage the combination of human intelligence and artificial intelligence by boosting reciprocal learning. A word of caution, however. Computerized support should be designed to enhance learning at both the functional level and the communication level, which is often downplayed.

8 CONCLUSIONS

In a world increasingly reliant on AI, we wish to advance a new configuration of human and artificial intelligence, one that keeps the human in the loop for ongoing efforts requiring continuous learning. In particular, the use of automatic text classification and other problems that demand context-aware intelligence have led to an explosion of research and development of AI-based classification. Although, human-in-the-loop classifications are a long-standing goal, little progress has been made at the micro-level determining how such systems should be implemented and used. Our HML configurational view of human-machine learning has the potential to introduce a game-changing paradigm to the design and use of machine intelligence. Through a design-science research process, we have demonstrated both the feasibility and usefulness of an HML configuration as an information system artifact, which we implemented and evaluated with Fusion. With two distinct case studies, we demonstrated our new approach to reciprocal learning, creating an ongoing loop in which human experts and machine models incrementally increase each other's understanding. Both studies showed: improvements in classifier accuracy over standalone ML models; higher levels of explainability and domain experts' perceptions of a productive interaction with the machine; and the ability to effectively improve conceptualization and understanding.

The four design principles that we have derived from this work will help developers to design, implement, and improve systems that support reciprocal learning between human and artificial intelligence. The improved effectiveness of ML systems, the acceptance of such systems by domain experts and society alike, and our ability as humans to remain in the loop, are all essential in a world swept by AI applications. Our vision of keeping the human in the learning loop may be seen in contrast with a trend towards complete automation, notably in the promise of a Deep-learning systems that aim at automatic ML (AutoML) [32]. We believe, however, that there is room for complementary visions in the foreseeable future, one that automates the learning of new but structured tasks, and the other that supports the learning of new unstructured tasks.

ACKNOWLEDGMENTS

This project was supported by the Blavatnik Interdisciplinary Cyber Research Center at Tel-Aviv University and Israel National Cyber Directorate (INCD). We also acknowledge the fruitful collaboration with Sixgill.

REFERENCES

[1] Ahmed Abbasi and Hsinchun Chen. 2005. Applying authorship analysis to extremist-group Web forum messages. *IEEE Intelligent Systems* 20, 5 (2005), 67–75. https://doi.org/10.1109/MIS.2005.81

[2] Ahmed Abbasi and Hsinchun Chen. 2008. CyberGate: a design framework and system for text analysis of computer-mediated communication. *MIS Quarterly* (2008), 811–837.

[3] Steven Alter. 2017. Nothing is more practical than a good conceptual artifact… which may be a theory, framework, model, metaphor, paradigm or perhaps some other abstraction. *Information Systems Journal* 27, 5 (2017), 671–693. https://doi.org/10.1111/isj.12116 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/isj.12116

[4] Ofra Amir, Finale Doshi-Velez, and David Sarne. 2019. Summarizing agent strategies. *Autonomous Agents and Multi-Agent Systems* 33, 5 (2019), 628–644.

[5] Philippe Baecke, Shari De Baets, and Karlien Vanderheyden. 2017. Investigating the added value of integrating human judgement into statistical demand forecasting systems. *International Journal of Production Economics* 191 (2017), 85–96.

[6] Evangelia Baralou and Haridimos Tsoukas. 2015. How is New organizational knowledge created in a virtual context? An ethnographic study. *Organization Studies* 36, 5 (2015), 593–620.

[7] Jean-Philippe Bernardy, Shalom Lappin, and Jey Han Lau. 2018. The influence of context on sentence acceptability judgements. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 456–461.

[8] Przemyslaw Biecek and Tomasz Burzykowski. 2021. *Explanatory Model Analysis*. Chapman and Hall/CRC, New York. https://pbiecek.github.io/ema/

[9] Richard J Boland Jr, Ramkrishnan V Tenkasi, and Dov Te'eni. 1994. Designing information technology to support distributed cognition. *Organization science* 5, 3 (1994), 456–475.

[10] Andreas Buja, John Alan McDonald, John Michalak, and Werner Stuetzle. 1991. Interactive data visualization using focusing and linking.. In *IEEE Visualization*, Vol. 91. 156–163.

[11] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8, 8 (2019), 832.

[12] Kathy Charmaz. 2006. *Constructing grounded theory: A practical guide through qualitative analysis*. sage.

[13] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[14] Hsinchun Chen, Wingyan Chung, Jialun Qin, Edna Reid, Marc Sageman, and Gabriel Weimann. 2008. Uncovering the dark Web: A case study of Jihad on the Web. *Journal of the American society for information science and technology* 59, 8 (2008), 1347–1359.

[15] Danielle Keats Citron and Frank Pasquale. 2014. The scored society: Due process for automated predictions. *Wash. L. Rev.* 89 (2014), 1.

[16] John W Creswell and J David Creswell. 2017. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

[17] John W Creswell and Cheryl N Poth. 2016. *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[19] Andreas Diedrich, Ulla Eriksson-Zetterquist, and Alexander Styhre. 2011. Sorting people out: The uses of one-dimensional classificatory schemes in a multi-dimensional world. *Culture and Organization* 17, 4 (2011), 271–292.

[20] Jonathan Dinu, Jeffrey Bigham, and J. Zico Kolter. 2020. Challenging common interpretability assumptions in feature attribution explanations. arXiv:2012.02748 [cs.LG]

[21] Natasha Duarte, Emma Llanso, and Anna C Loup. 2018. Mixed Messages? The Limits of Automated Social Media Content Analysis.. In *FAT*. 106.

[22] Paul M Fitts. 1951. Human engineering for an effective air-navigation and traffic-control system. (1951).

[23] Jerome H. Friedman, Trevor Hastie, and Rob Tibshirani. 2010. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software, Articles* 33, 1 (2010), 1–22. https://doi.org/10.18637/jss.v033.i01

[24] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. arXiv:1803.07640 [cs.CL]

[25] Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, Nicolas d'Alessandro, Joëlle Tilmanne, Todd Kulesza, and Baptiste Caramiaux. 2016. Human-Centred Machine Learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 3558–3565. https://doi.org/10.1145/2851581.2856492

[26] E Bruce Goldstein. 2014. *Cognitive psychology: Connecting mind, research and everyday experience*. Nelson Education.

[27] Charles Goodwin. 2000. Practices of color classification. *Mind, culture, and activity* 7, 1-2 (2000), 19–36.

[28] Carol Grbich. 2012. *Qualitative data analysis: An introduction*. Sage.

[29] Shirley Gregor, Leona Chandra Kruse, and Stefan Seidel. 2020. Research Perspectives: The Anatomy of a Design Principle. *Journal of the Association for Information Systems* 21, 6 (2020), 2.

[30] Shirley Gregor and Alan R Hevner. 2013. Positioning and presenting design science research for maximum impact. *MIS quarterly* (2013), 337–355.

[31] Tor Grønsund and Margunn Aanestad. 2020. Augmenting the algorithm: Emerging human-in-the-loop work configurations. *The Journal of Strategic Information Systems* 29, 2 (2020), 101614.

[32] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. 2018. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 784–800.

[33] WK Ip, Leela Damodaran, C Wendy Olphert, and Martin C Maguire. 1990. The use of task allocation charts in system design: a critical appraisal. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction*. 289–294.

[34] Ton Jörg. 2004. A theory of reciprocal learning in dyads. *Cognitive systems* 6, 2 (2004), 3.

[35] Adi Katz and Dov Te'eni. 2007. The contingent impact of contextualization on computer-mediated collaboration. *Organization Science* 18, 2 (2007), 261–279.

[36] Daniel Khashabi, Erfan Sadeqi Azer, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2020. On the Possibilities and Limitations of Multi-hop Reasoning Under Linguistic Imperfections. arXiv:1901.02522 [cs.CL]

[37] Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, and Brown. 2019. Text Classification Algorithms: A Survey. *Information* 10, 4 (Apr 2019), 150. https://doi.org/10.3390/info10040150

[38] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.

[39] William Marcellino, Christian Johnson, Marek N Posard, and Todd C Helmus. 2020. Foreign Interference in the 2020 Election. (2020).

[40] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg S Corrado, Ara Darzi, et al. 2020. International evaluation of an AI system for breast cancer screening. *Nature* 577, 7788 (2020), 89–94.

[41] Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*. 51–61.

[42] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[43] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1 – 38. https://doi.org/10.1016/j.artint.2018.07.007

[44] Roxana Moreno. 2004. Decreasing cognitive load for novice students: Effects of explanatory versus corrective feedback in discovery-based multimedia. *Instructional science* 32, 1 (2004), 99–113.

[45] Ikujiro Nonaka and Hirotaka Takeuchi. 1995. *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford university press.

[46] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058* (2004).

[47] Uta Priss. 2006. Formal concept analysis in information science. *Arist* 40, 1 (2006), 521–543.

[48] Sebastian Raisch and Sebastian Krakowski. 2020. Artificial Intelligence and Management: The Automation-Augmentation Paradox. *Academy of Management Review* ja (2020).

[49] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv:1602.04938 [cs.LG]

[50] Wolff-Michael Roth. 2005. Making Classifications (at) Work: Ordering Practices in Science. *Social Studies of Science* 35, 4 (2005), 581–621.

[51] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.

[52] Margrit Schreier. 2012. *Qualitative content analysis in practice*. Sage publications.

[53] David G Schwartz. 1995. Cooperating Heterogeneous Systems.

[54] Maung K Sein, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. 2011. Action design research. *MIS quarterly* (2011), 37–56.

[55] Thomas B Sheridan. 1995. Human centered automation: oxymoron or common sense?. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, Vol. 1. IEEE, 823–828.

[56] Galit Shmueli. 2017. Analyzing behavioral big data: methodological, practical, ethical, and moral issues. *Quality Engineering* 29, 1 (2017), 57–74.

[57] Galit Shmueli. 2017. Research dilemmas with behavioral big data. *Big data* 5, 2 (2017), 98–119.

[58] Yash Raj Shrestha, Shiko M Ben-Menahem, and Georg Von Krogh. 2019. Organizational decision-making structures in the age of artificial intelligence. *California Management Review* 61, 4 (2019), 66–83.

[59] Chaehan So. 2020. Human-in-the-Loop Design Cycles–A Process Framework that Integrates Design Sprints, Agile Processes, and Machine Learning with Humans. *Lecture Notes in Artificial Intelligence, 1st International Conference on Artificial Intelligence in HCI, AI-HCI, Held as Part of HCI International 2020, Kopenhagen, Denmark* (19-24 July 2020).

[60] Chaehan So. 2020. Understanding the Prediction Mechanism of Sentiments by XAI Visualization. *arXiv preprint arXiv:2003.01425* (2020).

[61] Ji Y Son and Robert L Goldstone. 2009. Contextualization in perspective. *Cognition and Instruction* 27, 1 (2009), 51–89.

[62] Lucy A Suchman. 1987. *Plans and situated actions: The problem of human-machine communication*. Cambridge university press.

[63] Lucy A Suchman. 2007. *Human-machine reconfigurations: Plans and situated actions*. Cambridge university press.
[64] Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of language and social psychology* 29, 1 (2010), 24–54.
[65] Dov Te'eni. 2001. A cognitive-affective model of organizational communication for designing IT. *MIS quarterly* 25, 2 (2001), 251–312.
[66] Lev S Vygotsky. 1978. Mind in society: The development of higher mental processes (E. Rice, Ed. & Trans.).
[67] Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. 2019. Improving Natural Language Inference Using External Knowledge in the Science Questions Domain. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 7208–7215. https://doi.org/10.1609/aaai.v33i01.33017208
[68] Richard Webby and Marcus O'Connor. 1996. Judgemental and statistical time series forecasting: a review of the literature. *International Journal of forecasting* 12, 1 (1996), 91–118.
[69] Karl E Weick, Kathleen M Sutcliffe, and David Obstfeld. 2005. Organizing and the process of sensemaking. *Organization science* 16, 4 (2005), 409–421.
[70] Gabriel Weimann. 2016. Going dark: Terrorism on the dark web. *Studies in Conflict & Terrorism* 39, 3 (2016), 195–206.
[71] David D Woods and Erik Hollnagel. 2006. *Joint cognitive systems: Patterns in cognitive systems engineering*. CRC Press.
[72] Doris Xin, Litian Ma, Jialin Liu, Stephen Macke, Shuchen Song, and Aditya Parameswaran. 2018. Accelerating human-in-the-loop machine learning: Challenges and opportunities. In *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning*. 1–4.
[73] Inbal Yahav, Onn Shehory, and David Schwartz. 2018. Comments mining with TF-IDF: the inherent bias and its removal. *IEEE Transactions on Knowledge and Data Engineering* 31, 3 (2018), 437–450.
[74] Efpraxia D Zamani, Anastasia Griva, Konstantina Spanaki, Paidi O'Raghallaigh, and David Sammon. 2021. Making sense of business analytics in project selection and prioritisation: insights from the start-up trenches. *Information Technology & People* (2021).

A FUSION'S ARCHITECTURE

This section provides an overview of Fusion's underlying architecture, as shown in Figure 7. This allows to see how we modeled the different aspects and components of our prototype, and its extensible and modular capability. For instance, the ML algorithm modules (shown on the top-right side of Fig. 7) are all managed through a shared interface controlled by the server. This way, additional ML model types can be easily added to the system. Similarly, the contextualization mapping components (shown on the top-left side) which are responsible for *mapping the domain expert's knowledge into a machine-compatible input* are designed and built in a modular fashion. Lexicon is one such mapping which we used in our work so far, however, other alternative mappings are possible and we began exploring these options. It is important to note that some components of Fusion are not fully developed yet, such as user management, however we are gradually adding and extending them. Overall, we envision Fusion becoming a fully-fledged framework for HML support systems in the future, and it is reflected in Fusion's architecture design.

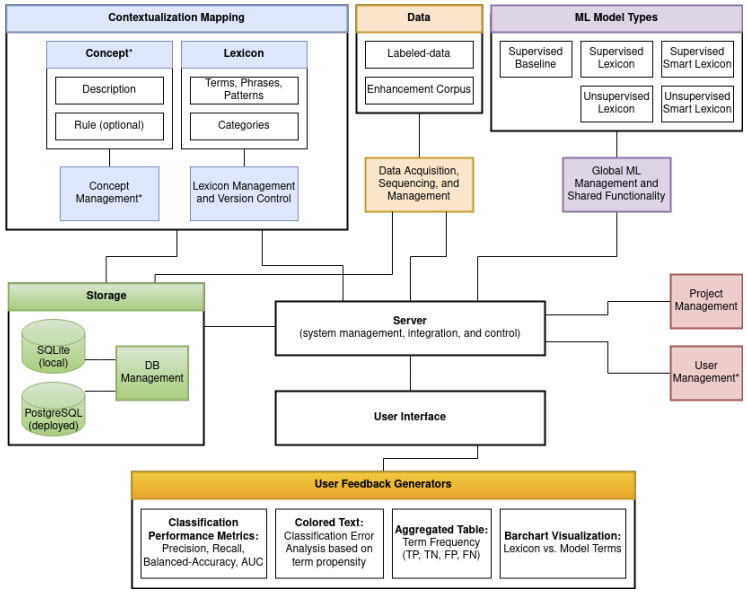


Fig. 7. An overview of Fusion's underlying architecture. Components marked with a star (*) are still under development and may not be fully functional yet.

B USER STUDY GUIDE SCRIPT - ROUND 1

B.1 Goal & Method

User tests can be run for many different reasons: (1) to validate a prototype; (2) to find issues with complex flows; (3) to gather unbiased user opinions; and (4) to get the insights that help create a better overall user experience. In our case, we have a computer-based software system that aims to guide and support the iterative process of machine-learning (ML) model generation for the purpose of text classification. The system aims to support the user's goals: The qualitative researcher aims to improve and evolve the lexicon as much as possible. Moreover, the system serves as a bridge between the quantitative and qualitative worlds, bridging the terminological and knowledge gaps.

For these reasons, we aim to test how effective and usable the system is in the use case of a qualitative user creating a lexicon and generating ML models to evaluate and improve the lexicon, as part of an iterative process.

B.1.1 Method. We will conduct a qualitative user study session, in which we will record the session and interview the participants. The interview will consist of semi-structured interview (with open ended questions). There will be tasks given to the user, based on the expected scenarios designed for Fusion.

B.1.2 What is being captured?

- Videos (screen capture) and audio of users completing tasks and thinking aloud
- Verbal answers to the qualitative semi-structured interview questions

B.1.3 Equipment.

- Laptop with Fusion already set up on it (R-Studio is needed at the moment)
- OBS software to record the screen capture and audio
- Microphone
- Input files (labeled data + lexicon), properly organized for the session
- Notebook for taking notes
- Print out of this guide

B.2 Pre-test instructions to participant(s)

- It is important to emphasize we are not testing you the participant, we are testing the system. Please don't worry if something seems confusing or doesn't work as you expected – do let us know though.
- Please provide open honest feedback.
- Think-aloud: please verbally communicate your thought process and anything else that goes through your mind when using the system. It is extremely helpful to us. We may remind you to do so as we go, as some people tend to forget to think-aloud after a couple of minutes.
- There is no time limit, so don't worry about it and no need to hurry up.
- During the test we are not allowed to interfere/comment/suggest things. We may ask specific questions, but we will refrain from answering questions during the tasks (or touching the laptop/fusion app).
- There might be errors or system crashing mid-testing (e.g., lack of memory). Fusion is still a prototype and a work in progress, plus we have a recording system in the background which is resource-heavy (CPU mainly). Please don't worry if these issues happen, we'll reboot the Fusion system and continue the user study from there.
- We are recording the session: both the screen and the audio of the conversation. This is to help us better analyze the user study results. We will also take some notes during the test, these are notes about the system, please don't worry about it.
- Risks to participants: NONE!
- Expected duration: about an hour.

B.3 Pre-Test Questions

- You have done the process of creating a text-classification model generation manually so far (not through a supporting software system). What challenges or specific pain points did you have in doing so?

B.4 Test

Note: Fusion is a prototype and is still a work in progress. For this reason, some things are currently hard coded. I've already gone ahead and created a user and a project for you for the purpose of this user study session. The project is called "Hidden Answers". You won't need to select or change it during the session.

Task 1: Upload labeled data into system. A labeled data file is given to you, with 5,337 data rows and 17 columns, each data row is labeled (as one of: SU, NS, OS, N/A) – this labeling is the ground truth for the Fusion system. The first 513 rows in this file were used by you to generate a lexicon. [Moderator shows file in Excel]

Task: You need to upload the labeled data into the Fusion system.

Pre-task questions:

- How easy do you think it is going to be? (on a scale of 1 to 5 where 1 is very easy, and 5 is very difficult)

Post-task questions:

- How many rows are in the uploaded data inside Fusion? Why 5,285 and not 5,337?
- Was the data split into several smaller sets? Which sets?
- Fusion is a system that supports an iterative process. How many iterations can we do with this given data?
- Out of the 17 columns in the data, which columns are going to be used for the text classification?
- How easy was it for you to accomplish the task? (1 to 5, 1 very easy... 5 very difficult) Why?
- How confident are you with your task's outcome? Why?

Success condition: successfully uploaded the data into the system 5,285 rows, and it has split that data into 4 sets: 513 training, 500 validation, 500 leave-out validation, 3,772 unassigned.

Task 2: Run a baseline model. As our first model, lets run a supervised baseline model with the data we uploaded. This will help you get a general idea of how a standard approach would perform. We'll use it as a reference point later when we begin using a lexicon.

Task: Run a baseline model and report on the performance metrics.

Pre-task questions:

- How easy do you think it is going to be? (on a scale of 1 to 5 where 1 is very easy, and 5 is very difficult)

Post-task questions:

- How well do you think the baseline model has performed? Why?
- How many SU were identified by the model? Out of how many SUs in total in this validation set?
- How many false negatives were there? (i.e., posts classified as NS by model but were SU in reality)
- What is the AUC score for the model?
- Let's look at the "Model-Assisted Analysis" tab, the top part only – this tab shows classification examples from the validation set; It shows the words the model has used for classifying each post. Blue words are words the model believes indicate NS, and red are words for indicating SU). Can you briefly explain **in your own words** how the baseline model works?
- How easy was it for you to accomplish the task? (1 to 5, 1 very easy... 5 very difficult) Why?
- How confident are you with your task's outcome? Why?

Success condition: successfully run a baseline model, and examine performance metrics. Correctly identify *true positive* classifications from the confusion metrics. Have a general idea of how baseline model works, with emphasis on understanding that it will examine all words from the data documents (i.e., all words from the posts and their titles).

Task 3: Upload lexicon. You previously generated a lexicon file on your own. It has 19 rows and 8 columns. These include: **words**, **phrases**, **patterns**, and **website links** — each one of these has a **label** (SU/NS), a **category** (e.g., Manufacture, Buy, Sell), and in some cases a **sub-category**. We provide this file to you now, and we'll use it as a lexicon for the ML model generation. [Moderator opens file in excel]

Task: You need to import this lexicon into the system.

Pre-task questions:

- How easy do you think it is going to be? (on a scale of 1 to 5 where 1 is very easy, and 5 is very difficult)

Post-task questions:

- Notice that the lexicon structure has changed slightly, it was "flattened" by Fusion to make it machine readable, but all the same content was kept. How many rows does it have?
- Please run an "unsupervised-lexicon" model. This model type simply counts the number of lexicon-terms from each post and calculates the resulting percentage as follows: the percentage of words labeled with SU divided by percentage of words labeled as NS. Go back to the lexicon tab afterwards. On average, how well did the lexicon we imported performed?
- How easy was it for you to accomplish the task? (1 to 5, 1 very easy... 5 very difficult) Why?
- How confident are you with your task's outcome? Why?

Success condition: successfully import a lexicon into the system with 229 entries. Successfully run an unsupervised-lexicon model and by using it, see the average metrics of the lexicon (average across all model runs with this specific lexicon version).

Task 4: Run a supervised lexicon model. Now, lets run a supervised lexicon model with the data and lexicon we uploaded. As opposed to a baseline model (which is also a supervised model), this way we have much more control over how the model will classify the posts – only terms that appear in the lexicon will be examined by the model.

Task: Run a supervised lexicon model and compare it to the baseline model we run earlier.

Pre-task questions:

- How easy do you think it is going to be? (on a scale of 1 to 5 where 1 is very easy, and 5 is very difficult)

Post-task questions:

- How well did the supervised-lexicon model compare to the baseline model? Which one had a better Recall value?
- We ran three models so far. Did all of them use the same validation data set?
- Which of the three models do you believe is the one you'd choose to continue working with when continuing to develop the lexicon further?
- If you wanted to run another iteration of a specific model, how would you go about doing in Fusion?
- Is there anything that is unclear to you about the model you generated or its performance metrics?
- How easy was it for you to accomplish the task? (1 to 5, 1 very easy... 5 very difficult) Why?
- How confident are you with your task's outcome? Why?

Success condition: successfully run a supervised-lexicon-wikipedia model, and examine performance metrics. Correctly compare this model type to previous models (e.g., correctly comparing true classifications, etc). Have a general idea of how a supervised models works, with emphasis on understanding that **a lexicon is the control mechanism the qualitative-researcher has over the ML system**.

Task 5: Analyze results and come-up with improvements to lexicon. At this point, we would like to analyze the results of the last model we ran – the supervised lexicon model (wikipedia based). We would like to better understand why the performance was as it was, and why the machine learning model classified (or misclassified) as it did. For this purpose, we'll use the "Model-Assisted Analysis" tab. [Moderator please notice if user switches to the correct model id]

Task A (no guidance): By using this tab, come up with **two improvements** you can do to the lexicon.

Pre-task questions:

- How easy do you think it is going to be? (on a scale of 1 to 5 where 1 is very easy, and 5 is very difficult)

Task B (specific instructions):

(1) Drill down to specific examples: examine all the false negatives the model has generated (i.e., model classified as NS while in reality they should have been SU). Extract terms that can be added to the lexicon in order to improve it.
(2) Look at the high level aggregated table and bar chart. Which terms were the best predictors in your opinion? Which terms were best predictors of NS posts?

Post-task questions:

- Let's select a group of words on the bar chart to limit the analysis. Select a group of five words of interest (e.g., blue words that have high positive probability). What can you learn from this?
- Based on this in-depth analysis, how well do you think the model performed? Why?
- Do you believe you can improve the lexicon? How?
- What unexpected things did you find here?
- How easy was it for you to accomplish the task? (1 to 5, 1 very easy... 5 very difficult) Why?
- How confident are you with your task's outcome? Why?

Success condition: successfully use the drill down view to examine examples from the validation set and identify terms that can improve the lexicon (either by adding or removing them from lexicon). Successfully sort the aggregated table based on TP, FN, etc... to get a better view on which words were best predictors per case.

B.5 Post-test Open Questions

(1) What do you think of this user study session?
(2) What is your overall experience with Fusion?
(3) What difficulties did you have during the whole process?
(4) What was easy for you?
(5) If you created a lexicon, how would you know it is good?
(6) Was it easy or difficult for you to create a machine learning model and gain insights from it? Why?
(7) Would you like to add something that you did not mention? (share insights)

C USER STUDY GUIDE SCRIPT - ROUND 2

C.1 Pre-test instructions to participant(s)

- It is important to emphasize we are not testing you the participant, we are testing the system. Please don't worry if something seems confusing or doesn't work as you expected – do let us know though.
- Please provide open honest feedback.
- Think-aloud: please verbally communicate your thought process and anything else that goes through your mind when using the system. It is extremely helpful to us. We may remind you to do so as we go, as some people tend to forget to think-aloud after a couple of minutes.
- There is no time limit, so don't worry about it and no need to hurry up.
- You'll be operating Fusion remotely, from your own laptop.
- There might be errors or system crashing mid-testing (e.g., lack of memory). Fusion is still a prototype and a work in progress, plus we have a recording system in the background which is resource-heavy (CPU mainly). Please don't worry if these issues happen, we'll reboot the Fusion system and continue the user study from there.
- We are recording the session: both the screen and the audio of the conversation. This is to help us better analyze the user study results. We will also take some notes during the test, these are notes about the system, please don't worry about it.
- Risks to participants: NONE!
- Expected duration: about two hours.

Task 1: Run an Unsupervised-Lexicon model.

- Let's focus on posts that the system didn't capture (i.e., that it didn't classify correctly). Open the "Model-assisted Analysis" tab and look at the examples from the validation set—focus on the **false-negatives**.
- Examine the false-negative posts.
- Do you understand the feedback you are shown by Fusion?
- Why do you think the system misclassified them? Focus on specific posts and point out specific reasons you believe cause the misclassification.
- Choose 2-3 posts and explain to us **how does your categorization applies to them**. Notice that we are also showing you your existing categorization tree on the right side of the screen.
- What **NEW insights** did you gain by these actions?
- Considering that you are working with an unsupervised model, **what changes can you make now to the lexicon** to improve it? Lets do (some of) them now live. Feel free to do these changes in Excel and upload a new version of the lexicon.

Run another, parallel, Unsupervised-Lexicon model – this way we run the model on the exact same data. Let's look again at the false-negatives.

- Was there an improvement in the classification? Why?
- Do you feel your lexicon is improving? Why?
- Are there any posts that don't have any colored words? Should your lexicon capture these as well?
- To better cover posts from this domain, how can you improve your conceptualization and lexicon?

Task 2: Now, let's move to using a Supervised model. This time, the model will apply *"his own judgment"* in the classifications. This gives you a great opportunity to consider new aspects and terms you haven't included in your lexicon. It gives a different perspective that allows you to better learn as part of the iteration process.

- Run a Supervised-Lexicon model (first iteration).
- Does it **perform better (or worse)** than the unsupervised ones?
- Does it **capture better (or worse)** the posts in this domain? (for instance, posts that were not colored before)
- What can you learn from this model's results?
- Can you use this feedback to improve your lexicon? How? (give specific examples)

Make any improvements you identified and let's run a second iteration for the supervised-lexicon model. This will apply your improvements to a new validation data subset – we will examine this new data subset now.

- We now see **new posts** that you haven't examined previously (i.e., you did not examine these during your conceptualization phase). Let's continue focusing on false-negatives (if none, focus on false-positives).
- Can you walk us through on how you'd continue building your conceptualization by using these new posts?
- What new insights did you gain this time?
- Explainability: do you understand the difference in feedback when using supervised vs. unsupervised models? What changes can you make now?
- Do you feel that your conceptualization is better or worse than two iterations ago? How? Why?

Received January 2021; revised April 2021; accepted July 2021