Lessons Learned Developing a Visual Analytics Solution for Investigative Analysis of Scamming Activities

Jay Koven *Member, IEEE*, Cristian Felix, Hossein Siadati, Markus Jakobsson and Enrico Bertini

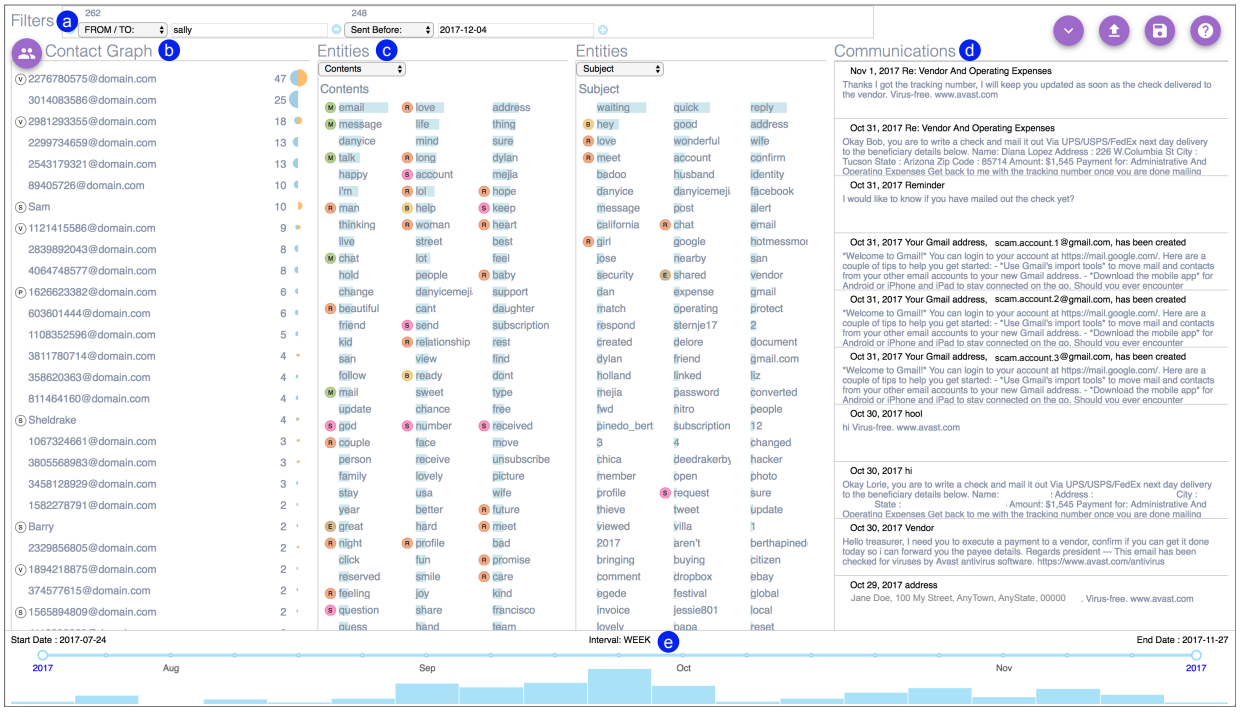


Fig. 1. Beagle has 5 main panels: (a) the *Interactive Query Panel* to specify queries composed of correspondents, keywords and time; (b) the *Correspondents Panel* to list correspondents and their communication frequency; (c) the *Content Panels* to display keywords or entities extracted from the emails; (d) the *Email Panel* to display emails in full detail; (e) the *Timeline Panel* to show the temporal distribution of the emails. All panels display information conditioned to the current query.

Abstract—The forensic investigation of communication datasets which contain unstructured text, social network information, and metadata is a complex task that is becoming more important due to the immense amount of data being collected. Currently there are limited approaches that allow an investigator to explore the network, text and metadata in a unified manner. We developed *Beagle* as a forensic tool for email datasets that allows investigators to flexibly form complex queries in order to discover important information in email data. *Beagle* was successfully deployed at a security firm which had a large email dataset that was difficult to properly investigate. We discuss our experience developing *Beagle* as well as the lessons we learned applying visual analytic techniques to a difficult real-world problem.

Index Terms—Visual Analytics, Email Investigation, Email Forensics

1 Introduction

In this paper we present a case study on the development of a visual analytics email forensics system to help a group of cyber-security experts understand how a network of cyber-scammers interacts among themselves and with their victims. This case study provides information about three main aspects of the project: (a) the iterative development of the visual analytics tool; (b) the data analysis problems investigators face in a complex environment; (c) a series of lessons we have learned while interacting with the investigators and adapting our tool to their needs.

Reporting on complex case studies on the use of visual analytics in business environments gives us an opportunity to better understand which gaps need to be bridged when research ideas are transferred from the lab to real-world environments. These environments often have unexpected constraints and novel challenges rarely encountered in purely academic settings.

In this case study, we focus on the deployment and iterative development of our tool with a security firm that has a large dataset consisting of emails captured from cyber-scammers. The overarching goal of the security firm is to generate a better understanding of how scam-

- *Jay Koven: NYU Tandon School of Engineering. E-mail: jkoven@nyu.edu.*
- *Cristian Felix: NYU Tandon School of Engineering. E-mail: cristian.felix@nyu.edu.*
- *Hossein Siadati: NYU Tandon School of Engineering. E-mail: hossein@nyu.edu.*
- *Markus Jakobsson: Amber Solutions Inc. E-mail: markus@ambersi@com.*
- *Enrico Bertini: NYU Tandon School of Engineering. E-mail: enrico.bertini@nyu.edu.*

mers operate and, through that, generate novel ideas and technologies to protect customers, prevent damage and, whenever possible, disrupt criminal activities.

In the context of this study, we developed a visual analytics system called *Beagle*. *Beagle* aims at making improvements over the often rudimentary yet familiar tools forensic investigators use, such as email clients, text editors, and database applications; which are not designed for the specific purpose of supporting forensic investigations. Our goal was to use *Beagle* as a "probe" to better understand the forensic process and also to produce, as a byproduct, an effective data analytics solution for the forensic team.

The main idea behind *Beagle* is to keep intact the metaphor of searching in a collection of emails, in a fashion similar to email clients, while providing additional capabilities that enhance and support the investigation. More precisely, the enhancements *Beagle* provides over a standard email client include: progressive and reversible data querying; coordinated views to keep results in context; content extraction and summarization of email's data; tagging capabilities to externalize knowledge built by the investigators.

Beagle has been developed through several design iterations during a period of about two years interacting with a number of different actors. In this paper we focus upon our interaction with a team of cyber-analysts from Agari, a security company interested in the analysis of scamming activities. *Beagle* is currently being used in the firm as the main tool for investigation and the version presented in this paper is the latest one we developed in collaboration with the team.

The pivital outcome of this research, in addition to the development of a novel visual analytics system and the description of a relevant application domain, consists of a number of lessons learned which we present at the end of the paper in Section 7. We believe these lessons, while stemming from our specific experience with this project, can provide value to the broader visual analytics community. They have practical value when adopted for development in new applied projects. They also have value for the research community as they provide reflection points on the need to better understand some crucial aspects of the visual analytics process.

The paper is organized as follows. In section 2, we describe domain problem, data, tasks and organizational constraints. In Section 3, we describe the related work and explain how *Beagle* relates to existing approaches. In Section 4 we describe *Beagle* in detail, together with its design rationale, and in Section 5, we describe relevant aspects of the design process. In Section 6, we describe three examples of the use of *Beagle* "in the wild", that is, as documented by our collaborators in the Agari team. Finally, in Section 7, we provide our set of six lessons we have learned during the course of this project.

2 Understanding Scamming and Scammers

The case study we present in this paper stems from a collaboration of our research team with Agari, a private security firm who develops cybersecurity solutions which protect companies from various types of cyber-attacks. Our collaboration focuses on one particular aspect of their work: understanding how scammers operate and ideate solutions to counteract their activities.

Email-based scamming has existed for years, and many of the types of abuse existed before email did. One example of this phenomenon is the so-called *Spanish prisoner* scam, documented as early as 1910 [26], in which a victim is duped to send funds to help release a wealthy prisoner, with the expectation of a spectacular reward. While the exact pitches have evolved over the years, and different versions of the scam have been developed to address different situations, many of the underlying principles have remained the same. The foremost principle is that of establishing or exploiting already-existing trust.

Some common forms of email scamming are Business Email Compromise (BEC)[1], romance scams and rental scams. BEC scams are targeted attacks in which a trusted party is impersonated, and an intended victim is asked to initiate a payment or send sensitive information. While there are consumer-facing instances of this type of fraud – such as real estate purchase scams in which a home buyer is tricked into sending closing funds to the impostor – most of these scams target enterprises or government institutions. In a romance scam, the scammer usually makes initial contact through a dating website and gains the confidence of the victim over time with the goal of either "borrowing" money from the victim or convincing the victim to become a mule or middleman in another scam. Rental scams target people wishing to rent a home, whether for a long term or as a vacation rental. In a rental scam, the scammer typically makes initial contact through an online service such as Craigslist, advertising a property that does not belong to the scammer, asking upfront for an "refundable" deposit before the victim is allowed to see the property.

As part of this research, Agari gathered entire mailboxes from accounts active scammers use for their illegal activities. The resulting email dataset (which we describe in more details below) consists of all the communications happening within and between such accounts, which include: attempted scam conversations; communications between scammers and their victims; communications between scammers; and emails scammers receive from various types of services (e.g., PayPal, Google Voice, Facebook).

Agari's goals for their investigation and analysis are as follows. First, they want to understand what kind of scams are in an email collection and how the scammers conduct these scams. More precisely, Agari wishes to understand how the scammers operate and how they interact with their victims – but also how they cooperate with each other to perpetrate crimes. Second, they want to identify key features of the scam email content and meta-data to devise strategies for the interception of future scam emails. Third, they want to identify bank accounts associated with the criminals, to notify banks and law enforcement before financial damage is incurred. Similarly, Agari also wants to identify the real identity of the scammers, whenever feasible, to aid law enforcement to pursue prosecution.

2.1 Email Dataset

Agari's email dataset consists of emails collected directly from scammers through their *BEC Automated Deception System* [3]. The dataset was acquired using a set of commonly used methods which involve providing (to the criminals) a detailed end user license agreement (EULA) and obtaining the express permission of the email account holders to access their respective accounts. The dataset corresponds to accounts used to mount Business Email Compromise (BEC) attacks against enterprises protected by Agari's "Enterprise Protect" product, which is a service built to detect the use of deception and, in particular, targeted attacks impersonating a trusted user. The dataset includes 59,652 unique emails collected from 78 separate email accounts.

The email data set is structured as follows. At the top level the accounts are organized into criminal organizations; groups of mailboxes Agari found to be part of the same group of scammers. Each criminal organization contains the individual accounts with all the emails found in the three standard Gmail folders: *inbox*, *sent* and *spam*. All the emails collected from these accounts are integrated into a common repository, which tracks which organization, account and mailbox they belong to. Each email is characterized by its textual content and subject line as well as the following meta-data: *from address*, *to addresses*, *date*, *time* (more metadata is available but not used explicitly in the project). Conceptually, the data and analysis space is defined by three main entities: **who**, the email accounts of individuals and services; **what**, the content of the messages exchanged; **when**, the time and date of when the communication takes place. As illustrated in Section 4 this information is then used to create additional derived data which is useful for the analytical tasks identified during our collaboration.

2.2 Analytical Intents and Benefits of Visual Analytics

Our numerous interactions (conference calls, email exchanges and semi-structured interviews) with the Agari team during a period of about four months enabled us to build a picture of their main analytical intents and goals. We organize these into three main categories.

[1] Business Email Compromise is also commonly referred to as "CEO fraud", as it typically involves the impersonation of a CEO to the CEO's colleagues.

(G1) Understanding the dynamics of scamming. Analysts have some basic understanding of how scammers operate but much more is left to be understood – especially as many scams are constantly evolving, and new scams are emerging periodically. With email, they are hoping to shed a light on many of the open questions on the dynamics of scamming. Examples include: Do scammers cooperate with each other and what role do they play in their attacks? How do scammers probe for victims? How do victims respond to these probes and what do scammers say to close a scam? Do scammers specialize in one type of scam? Do scamming activities evolve over time? As investigators find ways of answering questions like this, the answers typically give rise to follow-up questions.

(G2) Identifying victims and perpetrators. In addition to generating a better understanding of how scammers operate, analysts also want to identify both victims and perpetrators to disrupt illegal activities and to facilitate intervention. Pinpointing the actual identity of scammers requires linking several email messages across multiple accounts and also searching for relevant pieces of information such as bank account, social security number and phone numbers. Questions analysts may have include: Are bank account numbers included in the email communications of this set of accounts? Who is communicating with the identified victim? What type of services (e.g., orders on Amazon, Google voice mails) does the account of this scammer use?

(G3) Organizing scams and user accounts into categories. As the analysts perform the type of analyses described above they also want to be able to label/annotate their data according to the information they discover. More precisely, they want to keep track of the types of scams and accounts they discover during the analysis. For example, once they discover a particular victim, scammer or kind of service, they want to be able to keep track of this information and use it in their future investigations. Similarly, they want to annotate terms with specific labels identifying scam types or other useful information. Such activity has two main purposes. The first one is to keep track of information generated during the analytical process. The second one is to use this information as additional metadata to gather statistics about the email collection and to query/filter the email data set.

2.2.1 Benefits of Visual Analytics

As briefly mentioned in the introduction, analysts typically perform this kind of analysis by loading email data into email applications and using their search functionalities to delve into a large collection of messages. In some cases, analysts also use a large variety of text editors, database applications, and scripts to gather documents containing specific combination of words and parameters from the meta-data. While experienced investigators can become extremely proficient with such tools, we, early on, conjectured that visual analytics solutions can potentially reduce the barrier to adoption, speed up the process and enable new types of analyses. There are two main areas of intervention we have identified as potentially benefiting from a visual analytics approach: *query specification and reformulation* [8] and *content summarization and visualization*.

(B1) Query specification and reformulation. An important characteristic of investigative analysis is that it is almost always driven by some pre-existing knowledge and interest investigators have about a specific set of events, individuals, places, etc. In our case these can be email accounts, bank account numbers as well as the names of a specific set of people or locations. An important consequence of this observation is that the analysis is fundamentally driven by a *data querying interaction model*, where the analysts transform their question into a query then inspects, interprets and evaluates the results then produce new questions and queries to narrow the gap between results and the desired outcome. Visual analytics can support this process by making it easier to specify, evaluate and reformulate queries. In particular, interactive/visual query mechanisms can (a) reduce the burden on memory by making existing settings and options visible; and (b) make query reformulation more intuitive by making it easier to change only some elements of an existing query (e.g., add a keyword to the existing query).

(B2) Content summarization and visualization. The biggest limitation of email clients and text editors used for investigative analysis is that these tools force the analyst to sift through a large set of documents at the lowest possible level of abstraction, that is, at the level of each single message. In addition, this type of approach does not make any distinction between content and meta-data, so as a consequence it forces the analyst to evaluate all information using a textual representation. Visual analytics can overcome these issues by providing effective data summarization and visualization methods for the content (that is, email subject and body) as well as appropriate visual representations for the associated meta-data (that is, email addresses and time).

All these observations are the basis of the inception of *Beagle*, an interactive data analysis tool developed to address the problems described above. *Beagle* will be described in detail in Section 4 after providing information about related works in the next section.

3 Related Work

A comprehensive work on text based data for investigative analysis has been done by the Jigsaw Project [14, 18]. While not only directed at emails, this work focuses on supporting the investigative process by creating tools which help the analyst find and map relationships among many data sources found in intelligence datasets. These relationships can be between people, places and things in any combination. These tools help the analyst piece together a coherent story from information contained in a document set typically linking data in many different view types and covering different aspects of the data. With Jigsaw our solution shares the idea of extracting content from unstructured data and using it for data exploration. One major difference is our stronger focus on interactive data querying and decreased emphasis on obtaining overviews of the document collection and its entities. Our approach, other than being motivated by the need of investigators to focus on specific queries, is also justified by the need to work with much larger data sets, typically up to hundreds of thousands of documents.

Haggerty et al. [7] analyzed the problem of forensic analysis of email data and proposed a framework capturing the main needs of such an analysis. One of the main findings is that there is a lack of tools to perform this kind of analysis. Their follow-up paper [6] shows some of the potential of visualizing the relationships of the social network derived from email accounts combined with email content using tag clouds for emails at the folder level. Other works focus entirely on social network graph analysis [4, 24] to explore email data sets with the general goal of discovering hidden connections within data. In our work however we found that analyzing the entire social network derived from the connections between email accounts is rarely useful and investigators prefer to issue specific queries and then connect the evidence they discover.

Martin et al. [19] explored large collections of emails in order to discover spam. Their methods focused on analyzing features of emails such as number of attachments, embedded images and attachment types. While they were not analyzing the content of messages, their work shows that other email features such as choice of punctuation, or number and type of attachments can yield important information about the documents, such as whether or not an email is spam. *Beagle* does not deal with spam but focuses entirely on scamming, an activity that requires a much deeper look at the content of the emails exchanged.

Li et al. [17] explored automated clustering of emails by feeding information derived from semantic analysis of email subject lines into an SVM classifier that was used for topic analysis. They determined the success of their analysis by how closely it clustered the emails according to the existing folders used to organize the collection. Kulkarni and Pedersen [16] similarly explored the content of email clusters in order to assign relevant labels to groups of emails. Thematic clustering as a method to improve forensic search results was also demonstrated by Beebe et al. in a series of works [2, 1]. One problem we noticed with these approaches is that investigators are normally not interested in obtaining summaries and overviews of an entire collection (especially when it is particularly large and heterogeneous). There-

fore, even though automated summarization methods can have some value, in practice it does not seem to match with the particular needs of investigators. However, as we will see in the later sections of the paper, using machine learning methods to extract entities and important keywords seems to be a particularly effective method to support investigative analysis, especially when paired up with querying functionalities.

In EmailTime, Joorabchi et al. [13] explored techniques for the visualization of temporal relationships of emails. Again these techniques show interesting characteristics of a data set but they are limited to one of the interesting aspects an investigators may be interested in. While temporal relationships are an important aspect of data they need to be combined with more features to be a useful part of the investigative process.

Kerr [15] explored the relationships between senders and receivers in email threads using a unique arc visualization which displays connections between senders and receivers in an email thread using a series of arcing arrows to show the connections. This work led us to thinking about the importance of tracing the sender/receiver relationships in search results in addition to threads. *Beagle* expands on this by showing the related correspondents, entities and emails related to the current query. This broadens the investigators' understanding of the relationships between the correspondents and content of the emails.

In summary, there is only a small set of solutions for forensic analysis of email data despite a growing need for this kind of activity. Existing approaches are either too broad or focus on aspects that do not seem particularly relevant for real-world investigations. In particular, approaches that aim at visualizing the entire collection all at once do not seem to solve the problem. Relatedly, approaches that focus exclusively on visualizing and analyzing email data as a social network do not seem to match the real needs of investigators.

4 BEAGLE

Beagle is the result of two years of iterative development of prototypes for the email investigation problem. In Section 5, we provide details of how our thinking about the design evolved over time and how we arrived at the final version we present in this paper. In this section we first present this final version, starting from the design rationale and following with a detailed explanation of its computational, interactive and graphical capabilities. The design rationale derives directly from the observations we provided in Section 2 regarding *analytical intents* and *major limitations* of existing approaches. More specifically, we designed *Beagle* using the following design principles.

(P1) Maintain similarity to email clients interaction. We want our users to feel comfortable switching from an email client (often the preferred tool) to *Beagle*. For this purpose it is important to maintain some common user interfaces elements found in emails clients. These include, search functionality and access to email content (including subject, to, from and cc addresses as well as the body).

(P2) Make data queries the main driver for exploration. As noted above, analysts think in terms of specific leads they have in mind involving combinations of email addresses, keywords, and times/dates. For this reason everything in *Beagle* is driven by user specified queries. Each view (which we describe in detail below) is always "conditioned" to the currently specified query except when no query is specified, in which case the view displays a summary of the whole collection.

(P3) Present results in context. Every time a query is issued we summarize the results in a combination of three main facets; which constitute the *context* of the current result set. These facets correspond to the three main type of logical entities found in emails (and in many other investigative scenarios), namely: the *who* (email addresses), the *what* (the email content and subject) and the *when* (time and date). In *Beagle*, each of these three entities has a separate user interface component and each employs a visual representation specific to the information it displays.

(P4) Make queries visible and easy to extend or modify. Analysts often start with a partially specified idea of what to look for and refine their search as more of the problem is understood. For instance, one may start by looking for emails sent and received with a specific address, then realize there is a group of two or three people on which to focus, and then a specific set of keywords that may lead to the evidence of interest. For this reason providing query mechanisms that permit easily extending or modifying an existing query is a crucial need for investigative analysis. Similarly, since all results shown in the interface are conditioned to the current query, it is important to make sure that parameters of the query are visible and easy to access.

(P5) Provide access to content at multiple levels of granularity. Standard email clients return results at the highest level of granularity possible, that is, the actual subject and text of each email. This is an inefficient and potentially hindering discovery. A better approach is to provide access to content at multiple levels of aggregation so that the user can get a sense of what the content or results set is and more easily jump to elements containing information of interest. For this reason in *Beagle* we employ content reduction and extraction processing methods that enable the analysts to gain an overview of content without having to scroll through the whole result set. More details on these processing steps are provided in the sections below.

(P6) Enable knowledge externalization. As noted before, analysts start their investigations with some pre-existing knowledge of the subject matter. Furthermore, as they develop their analysis they learn new information that is progressively added to their knowledge structure. Providing a method to "externalize" such knowledge and make it explicitly available in the interface permits the reduction of the cognitive burden on memory and also of using such knowledge structure as additional data to support data querying, summarization and visual comparison. For this reason, we equipped *Beagle* with tagging functionalities which we will describe in more details in Section 4.1.5.

These proposed design principles support the achievement of the goals stated in Section 2.2 as follows. Understanding the dynamics of scamming (G1) requires identifying interactions between scammers, victims, collaborators, and services over time. To this end, it is crucial to have a flexible querying mechanism (P2, P4) as well as track how unknown accounts relate to known actors and accounts (P6). Identifying victims and perpetrators (G2) requires identifying information in large sets of text. To this end, it is crucial to provide summaries of content that both readily react to changes of context (P3) and present data at multiple levels of granularity (P5). Finally, organizing scams and user accounts into categories (G3) is enabled by interactive tagging mechanisms (P6).

4.1 User Interface

As shown in Figure 1, *Beagle*'s user interface is organized around four main panels: interactive query, correspondents, content and time. The query panel on top is where queries are specified and edited interactively. The correspondents panel on the left is where email addresses are shown. The content panel at the center and left is where content is displayed at different levels of granularity, and the time panel at the bottom is where frequency of emails over time is shown. All of these panels are always synchronized and conditioned to the current set of filters specified through the query panel.

4.1.1 Interactive Query Panel

In order to make the current query always visible the query panel is located on top in its own area. After analyzing which types of queries investigators want to execute on the email database we realized they always involved a combination of keyword searches (on email subjects, body or both), specification of accounts involved (including whether they are sending, receiving or both), and time (typically before or after a date or between two dates). Therefore, we implemented a very simple query model in which users can combine these elements in a chain of smaller sub-queries connected by "AND and "OR" operators. Each sub-query is composed of a field type and values one can specify as filters. For instance, if one wants to retrieve emails received or sent by *jay@example.com* containing the term "vacation", the two sub-queries chained in "AND" would be: `to/from = "jay@example.com"` and `content = "vacation"`. *Beagle* has three main types of sub-queries, namely queries involving correspondents, content and

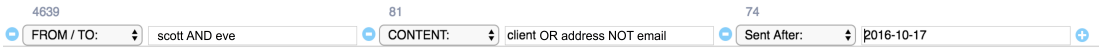


Fig. 2. Example of interactive query in *Beagle*. Each query is a combination of sub-queries, each one with a specific type and set of values. Queries can be extended by clicking on the plus sign and edited by removing sub-queries (with the minus sign) or by changing values in the open text fields.

dates. For each one there are a number of variants to accommodate specific search needs. For instance, in correspondents one can specify whether the address should be in the "from" or "to" field or both. In content one can specify if the search should be done using the subject line, the email body or both. And in the date field one can specify whether the email should be before or after a date or between two dates. While query languages can of course cover many more complex combinations than the one provided, in our experience we observed these satisfy the large majority of investigators search needs.

The query is presented to the user in the form of a simple list across the top panel of the display. A query is specified by adding or removing sub-queries and their values to this panel. The addition of sub-queries can be done directly by the user by clicking on a plus sign, specifying the type and typing the desired values to use for filtering. Sub-queries can also be initiated by interacting with elements of the interface that display information. For instance, users can double-click an address in the correspondents view and have a sub-query automatically added to the existing query (by default in an AND chain). As shown in Figure 2 a query is represented by a chain of sub-queries, each one with its own content type and values. The values can be joined by conjunction, union or negation. Each of these sub-queries can be removed with a simple click on the minus sign or edited by typing new values in the textual fields. Each sub-query also shows frequency numbers on top indicating the number of emails that are progressively reduced by applying filters which are always joined by conjunction.

4.1.2 Correspondent's Panel

The correspondent panel on the left side of the display is a list of all the senders and receivers in the current result set, that is, for any active query, the list shows only the email addresses contained in the current set of emails. The list is sorted by frequency of emails in the results. The email account for each correspondent is on the left side of the column and its frequency is shown on the right side of the column with an icon showing the amount of sent or received emails. The icon is a pair of semi-circles with the blue representing sent emails and the yellow representing received emails. The size of the semi-circles are relative to the largest number of sent or received emails. In addition when the investigator mouses over the correspondent a tool tip gives more detailed information about emails sent and received.

Every time a new query is issued or edited the list of correspondents is updated accordingly. One subtle but important behavior of the list is how it adapts to queries that contain one or more correspondent. In those cases, the selected correspondents are removed from the list. This is because when a given correspondent is part of a query it is disproportionately frequent in the results set and as such it skews the distribution of frequencies among other correspondents in the list. By removing such selected correspondents we make the frequencies of the connected addresses easier to evaluate and compare.

There is another important aspect of this interaction that it is worth highlighting. Early on in the project we discussed at length the trade-off of representing what is effectively a network structure (i.e., the connection between to and from addresses) with a node-link diagram versus a simple list such the one we use in *Beagle*. It turns out that paring up query functionalities with a simple list like the one we have just described, is a very powerful mechanism to make sense of relevant aspect of this network structure. In particular, in the large majority of cases that which an investigator needs to understand first is which accounts are connected to one or more account of interest; which is exactly what is possible to achieve by issuing a query and observing the connected "nodes" in the list. As we will discuss in more depth in Section 7, observing the power of a simple model based on queries and lists as a mechanism to explore a network structure, is one of the most interesting lessons we have learned in this visual analytics project.

Once an investigator has obtained an overview of which accounts are involved in a given set of emails it is still relevant at times to obtain an overview of who is connected to whom. The main limitation of the list method is that it is not possible to observe how the subjects in the correspondents list are connected among them. For this reason, we also provide a graph visualization which a user can open on-demand and display in a pop-up window. The graph arranges the nodes around a circle (using a cross-minimization algorithm to optimize readability[5]) and connects them with lines whose thickness is proportional to the frequency of emails exchanged (only considering those in the result set). By default, the graph shows only the top twenty most frequent notes. The user can then increase or decrease such value by clicking a plus/minus button available in the interface. In our experience, most queries produce result sets in which the frequency of connections is highly skewed. For this reason, in most cases displaying just a handful of nodes is sufficient to obtain the most relevant information.

4.1.3 Content Panels (Summaries, Entities and Full Details)

The main goal of the content panel is to provide access to the actual content of the emails returned by the query. As mentioned above, we designed *Beagle* following the principle that content should be accessed hierarchically. To this end, the content area is split into two parts. On the right-end side we have what we call "content summaries", sets of keywords that summarize the content and on the left we have the actual emails with both their subjects and bodies.

The content summary part is based on keyword extraction algorithms that process the current result set and extract the most relevant keywords according to some user-selected criteria. By default, the main criterion is the classic *term frequency over document frequency (TF/IDF)* strategy, which extracts terms that are specific to the currently selected set of emails (that is, terms that have high local and low global frequency). In addition to this default summary, other criteria can be used to extract the content of a specific type. To this end in *Beagle* we also include entity extraction methods [12] to pull keywords from the result set that may be of particular interest for a given type of investigation.

One crucial requirement we discovered during our interactions with the Agari team is that some type of entities are relevant and that it is important to provide flexible mechanisms to specify new types of entities when necessary. For this reason, we included mechanisms to define new entities in the content summary. At the present moment, *Beagle* provides classic entity extraction methods as well as pattern specifications implemented with regular expressions. This last one is particularly useful to identify textual elements of importance such as bank accounts, social security numbers, and passwords. The user can select multiple types of summaries to display and if necessary can select more than one at once. Each summary is displayed as a list of terms organized in rows and columns together with a small bar showing the frequency of the term in the current result set. Terms can also be double-clicked to add them as conditions to the current query.

The email panel on the right side displays the list of emails returned by the current query. The list of emails is displayed in a fashion similar to many email clients: the date and subject are displayed at the top of each email and the first few lines of each email are displayed beneath. Email results are sorted in date order with the most recent date first. The investigator can examine the contents of the entire email by clicking on the date/subject line to expand it in place so that the full email can be read. In addition to the compact list of emails described above, it became clear while working with Agari investigators that at

times the ability to read through the entire list of email results is important. To support this, the investigator can click on a toggle button at the top of the email panel to expand all emails in the result set. Finally, whenever a keyword is included in the query as part of a content filter, the corresponding terms are highlighted in the subject and body of the email.

4.1.4 Timeline Panel

The timeline panel presented at the bottom of the display shows the temporal distribution of the current result set, with bars proportional to the number of emails found at each time interval. Depending on the time span represented, the timeline units can be years, quarters, months, weeks or days so that that the number of divisions in the timeline does not get too small for the investigator to comprehend. The user can use handles on the ends of the timeline to add temporal constraints to the filters which are automatically added and reflected in the query panel above.

4.1.5 Tagging

One of the significant discoveries within our design process with Agari has been about the need for tagging capabilities. Early in our collaboration, it became clear that *Beagle* needed to provide methods to keep track of useful information generated by the investigation (a process often called "knowledge externalization" [21]). More precisely, the team voiced the need to tag email accounts to keep track of information about whether an account corresponds to a scammer, victim, service, or any other other categories. A similar function was requested to tag specific terms found during the investigation; typically terms that signal some particular type of scamming activity.

There are several reasons why such tagging activity is crucial for the investigative process. First, because it is possible to recognize tagged objects when they show up in new contexts (e.g., when the account of a victim shows up in the context of a new query). Second, because the tags manually provided by the investigators can be used as filters for new types of searches. For example, if investigators have tagged scammers and victims in the correspondents panel then they would be able to filter emails sent from scammers to victims or scammers to other scammers rather than only being able to select one account at a time as part of the filter. Finally, tags can be used as additional metadata to calculate statistics about scammers and scamming; an activity of great interest for Agari, as shown in more details in Section 6.

In order to support the activities outlined above, we equipped *Beagle* with a tagging functionality. Investigators can select one or more accounts in the correspondents panel and assign a tag to them. The tag can be a previously defined one or a new one defined at the time of tagging. Similarly, investigators can also tag individual terms in the content panel using the same mechanism. In order to make the tagging of accounts and terms visible, we manipulate the visual appearance of individual account names and terms as follows: Tagged account names contain a small symbol next to their name (using the first one or two letters of the tag) (Figure 1b). Tagged terms are also equipped with a set of symbols but are colored according to a predefined color palette (Figure 1c). The addition of color for tagged text was deemed important to more easily segment the terms into separate clusters when many of them are present at once.

In addition to the visual identification of objects, tagging also has an effect on the query panel on top. Once one or more tags have been defined, such tags can also be used as search filters to use in a new query. For example, once a tag has been defined to label all terms that correspond to a romance scam, it is possible to issue a query containing all words related to that type of scam just by using that single corresponding tag.

One final functionality we added during our collaboration is to set the tags of a given investigation as shared among a group of investigators. When such functionality is active, users can still conduct their independent research while also sharing the tags and their assignments. This functionality turned out to be extremely helpful to enable collaborative efforts; especially when the Agari team needed to generate as many tags as possible for further processing in external tools.

5 Development Process and Insights

As previously mentioned, *Beagle* is the result of about two years of explorations on how to develop effective interactive data analytics tools for email forensic analysis. In the following, we recount some of the main steps of the development process followed during this period. Our main intent is to highlight some relevant problems we faced during this time as well as mistakes we made. These will be the basis for some of the lessons learned which we will present later on in Section 7.

Although we did not explicitly model our study on the nine stage process espoused by Sedlmair et al. [23], we naturally progressed through most of the stages of that process. We met with potential collaborators, *winnowed* and *cast* several during our early stages, which led to our initial designs. This was followed in our later stages with new collaborators who helped us refine, test and evaluate our design. We break down the development timeline into two main phases: shallow collaborations and deep collaborations based on the level of cooperation and interaction we had with our collaborators at each phase. The first is characterized by our initial struggle to find domain experts willing to deeply engage in a mutually fruitful collaboration. The second is characterized by our deep engagement with the Agari team and the positive outcomes that stemmed from that. The shallow collaboration encompassed the precondition and core stages of the design process [23] while our deep collaboration overlapped with the core stages and covered the analytical stages of the process. The final set of guidelines we propose in Section 7 correspond to the *reflect* and *write* stages and constitute the main contribution of this work.

5.1 Phase 1: Shallow Collaboration

The project that led to the development of *Beagle* started from our desire to explore opportunities for visual analytics in the context of existing graduate-level work on digital forensics taking place in our department. As the first step in this project, we sought to find contacts with groups of lawyers and investigators which confirmed the need for such advanced tools and gave us hints on how interactive interfaces could be helpful in improving current state of the art processes. Our interactions confirmed that most investigators used email clients and/or text processing systems to analyze large sets of emails.

This initial phase was followed by tighter interactions with a group of investigators whose work involved analyzing large sets of emails. Unfortunately, for logistical reasons, we never managed to engage in an in-depth collaboration. This period, however, was characterized by intense tinkering with design ideas and useful interaction, though sparse and rare, allowed us to better understand both precisely what investigators needed and which kind of visual analytics approaches did not work. From this intense phase of experimentation, we learned a few precious lessons which we will discuss in more depth in Section 7. In summary, we learned that an excessive focus on extravagant visual representations can often hinder rather than help adoption of visual analytics solutions. Further, it is important to be wary of making data the main driver behind choosing the appropriateness of visual representation rather than choosing goals and tasks. More specifically, in our case, we found that framing the problem concerning social network analysis and trying to make network visualization the primary goal of our visualizations greatly hindered progress and ultimately failed to solve the investigators' main problems. Some examples of visualizations we created and discarded were: node-link graphs showing connections between accounts in terms of volume of emails exchanged, circular versions of the same graph, and linked lists to depict from/to relationships between users. These attempts seem to reflect one of the major visualization design threats mentioned by Munzner in her popular *nested models for visualization design and validation* [20], namely the *abstraction threat* in which: "*operations and data types do not solve the characterized problems*". We conjecture that such threat is amplified when we pretend to design visual representations and operations according to the metaphor that seems more natural for the available data (e.g., since the data is mostly a network the problem must be about network analysis).

5.2 Phase 2: Deep Collaboration

The second phase of our project started when we began collaborating with Agari. Agari had just obtained a substantial email dataset from scammers who had attacked or attempted to attack their clients, and they were having trouble investigating the dataset. After initial interactions, the Agari team voiced its interest in having us install *Beagle* on their servers to help them understand their dataset and in return they were willing to share their experiences with us and keep logs of their interactions. This started a period of intense collaboration which is the basis of the case study presented in this paper.

During this period of intense collaboration, *Beagle* went through several transformations. Here we highlight only two main relevant stages: the first one aimed at gathering feedback from the Agari's after they had the opportunity to try *Beagle* in their environment and the second aimed at developing a new version of the tool, which is the one presented in this paper.

During the first phase, we installed *Beagle* in Agari's environment and processed their emails so that they could be analyzed. In this phase, we instructed the team (a total of five individuals) to keep track of the information they gathered and to write notes about what worked well and what could be improved. After this phase, all team members voiced their excitement with the tool and its capabilities and confirmed that it had great potential for speeding up the analysis process as well as enabling the new type of analyses not attempted before. This phase resulted in a number of improvements to the interface and functionality of *Beagle* through a constant stream of feedback from the collaborators.

In this phase, we also gathered all the issues and limitations expressed by the team and used them to develop a new version of *Beagle*. Here we summarize the most relevant hurdles found in the tool at this stage. The first limitation was the lack of a tagging capability to enable annotation of terms and email accounts. The second one was the lack of methods to single out specific types of textual content/patterns, such as account numbers, email accounts and social security numbers. The third was about the need to scroll through the whole set of emails in full details once a sufficiently small set of emails has been singled out. While *Beagle* allowed the user to scroll through the entire set of snippets, it did not offer functionality to expand all of them at once. The final limitation was related to the need to automatically classify emails according to a specific type of scam found during the analysis. All of these issues, except the last one, were fixed to develop the final version of *Beagle* as described in Section 4.1. After several discussions, we agreed upon excluding the classification capability and keeping that step external since it could be solved more efficiently with existing tools outside *Beagle*.

6 Examples of Real-World Investigations

In this section, we present an in-depth investigation by one of Agari's investigators into the makeup of a criminal network. We then discuss two additional tasks in which Agari applied *Beagle*, the discovery of new types of email scams in the dataset and the understanding the business model of scamming.

6.1 Discovering a Criminal Network

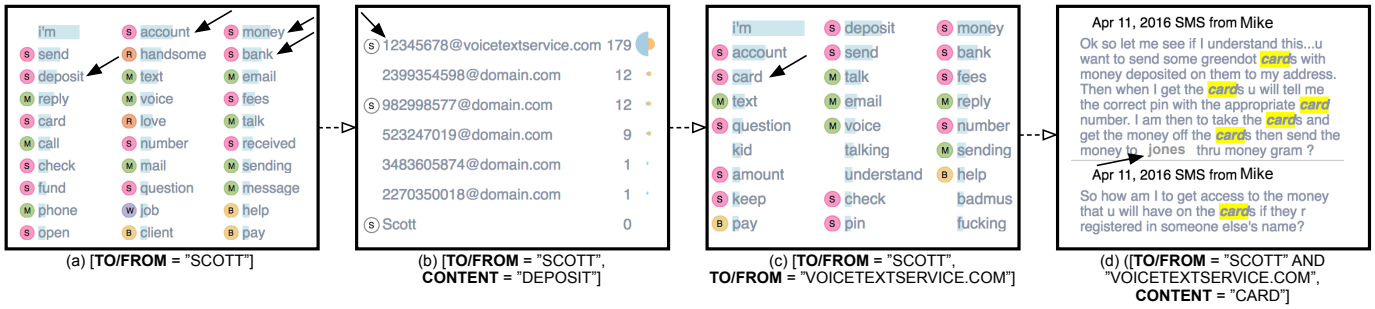
Here we describe an overview of an actual series of investigative steps using *Beagle* as presented to us by Alice, an Agari investigator, it is a demonstration of how she used *Beagle* to gain insight into how one of the criminal networks in the dataset was connected and how it functioned. All of the names used in this example are fictitious, including that of the investigator in order to protect their privacy. However, the analysis we report corresponds to the steps followed by Alice as presented to us after being requested to document a whole investigative session. It is important to note that when Alice started the investigation, the accounts of some scammers were known because they were the starting points for the data collection. Alice's goal for this investigation was to understand the criminal network which included the known scammer, whose account we will call Scott, from whom they had collected the largest number of emails.

Some steps of the analysis are depicted in Figure 3, below, with the intent of giving a sense of how information flows and evolves in a data analysis session with *Beagle*. The images show *exclusively small portions* of the interface; namely only those containing the specific information that was useful at that particular moment in time in the investigative process. In order to help the reader keep track of the current status of the system we also added, below each image, text depicting the query currently set at the time when such information was displayed.

Alice started his investigation with a to/from filter set to "Scott" ([TO/FROM = "SCOTT"]). In the Content Panel, under the keywords, he noted that the terms "account, money, bank and deposit" were all very near the top of the list and he associated these words with relevant scamming activity (Figure 3a). To investigate further, he added the content term "deposit" to his query list (a term Alice knows to be common in scamming activities) ([TO/FROM = "SCOTT", CONTENT = "DEPOSIT"]). At this point, he observed that one account dominated the communications with "Scott" containing such terms (Figure 3b). The name of the account indicated that it was a text and voice messaging service, i.e., "voicetextservice.com"). He then added the account to the query list (using to/from) which limited the results to emails between Scott and the service account ([TO/FROM = "SCOTT" AND "VOICETEXTSERVICE.COM", CONTENT = "DEPOSIT"]). Immediately it became clear from reading the subject lines and snippets in the email display that the emails were text messages from a single phone number with a name associated with them. A separate search on "www.usphonebook.com" (a web service to find owners of phone accounts) quickly revealed the probable identity and location of the person corresponding with Scott and he filed this information away for later action. At this point, Alice decided to remove the term "deposit" from the query to get a better overview of all the communications between Scott and this actor ([TO/FROM = "SCOTT", TO/FROM = "VOICETEXTSERVICE.COM"]). He then noted the term "card" was near the top of the Contents Panel(Figure 3c). Since credit and debit cards are a common method scammers use to move money around, Alice decided to add a content filter with the term "card" ([TO/FROM = "SCOTT", TO/FROM = "VOICETEXTSERVICE.COM", CONTENT = "CARD"]). The resulting list of emails (figure 3d) made it very clear that his new correspondent was a mule (an intermediary used to transfer money between accounts). By reading the emails, it was possible to understand that the mule, who we will call Mike, had deposited funds from prepaid credit cards into bank accounts. Mike had then forwarded the money using Moneygram (a money transfer company), as directed by Scott, to a new actor, Jones, whose name was in the content of one of the emails and whose role was not completely clear other than that the funds seemed to be directed to him.

Alice now wanted to learn more about Jones, so he removed the term "card" from the query and replaced it with "Jones" ([TO/FROM = "SCOTT", TO/FROM = "VOICETEXTSERVICE.COM", CONTENT = "JONES"]). By reading the resulting list of email subject lines and snippets, Alice was now able to give Jones a first name and noted dates and reference numbers which, given the previous instructions, were likely to be Moneygram reference numbers. Alice now wanted to identify Jones' role in the organization so he removed the to/from filters ([CONTENT = "JONES"]) and noticed Jones was mentioned in conversations other than those between Scott and Mike. From the list, he decided to add "paypalcustomerservicecentre@mail.am", which seemed like a scam since PayPal does not use this address ([CONTENT = "JONES", TO/FROM = "PAYPALCUSTOMERSERVICECENTRE@MAIL.AM"]). This resulted in three emails. By expanding the first email in the list to read its full contents, he found a mailing address for Jones in Nigeria. Jones appeared to be a receiver of funds and goods for the organization. Alice also found the email addresses of three potential victims who could be contacted to see whether actual losses had been incurred.

After going through a series of investigative steps like those described above Alice managed to reconstruct a whole network of actors and victims. To summarize, Alice started with one known scammer, Scott, and in a short time, using *Beagle*, identified Mike, Jones and



(a) [**TO/FROM** = "SCOTT"] (b) [**TO/FROM** = "SCOTT", **CONTENT** = "DEPOSIT"] (c) [**TO/FROM** = "SCOTT", **TO/FROM** = "VOICETEXTSERVICE.COM"] (d) ([**TO/FROM** = "SCOTT" AND "VOICETEXTSERVICE.COM", **CONTENT** = "CARD"]

Fig. 3. Sequence of knowledge from Scott to Jones. a) Content terms in Scott's emails. b) Correspondent's list for Scott's emails with content filter term deposit. c) Content terms in Scott's emails with the voice service account d) Partial list of Scott's emails with the voice service account and with the content filter card. Note: Information has been anonymized where necessary.

many other members of a criminal network that were carrying out various types of email scams. Alice could then use the the information he collected to do more in-depth investigations.

As a result of the knowledge gained from this and similar investigations, Agari took several actions. When they were positively able to identify individuals engaged in fraud, they reported them to appropriate enforcement agencies along with their supporting data for followup. Perhaps most importantly Agari disrupted the scammers' money chain by reporting bank accounts being used by the scammers and mules to a clearinghouse organization specifically set up to deal with this information. It not only allowed the bank holding the account to take action by shutting it down but also notified other banks about the account so that they could stop money from leaving their respective banks. Furthermore, Agari also shared data with Google, which enabled them to take action on these accounts. This way, Agari helped Google gain access to large numbers of accounts used for crime and to potentially improve their security controls given the additional knowledge gained.

6.2 Discovering New Types of Scam

Our investigator, Alice, had collected 78 mailboxes belonging to ten scamming groups, amounting to a total of 59,652 emails overall. One of his goals was to discover the types of scam methods that the scamming groups use to create revenue, and how these methods differed between the groups. Moreover, these discoveries would enable Alice to identify building blocks of each scam and target groups. The former would help Alice to cripple scamming activities by reporting to financial services and authorities, and the latter would help rescue victims in ongoing frauds by alerting and training them.

To discover types of scams in this big data set, Alice started by looking for the known class of scams such as "romance", "business email compromise", and "rental", by guessing or reusing a subset of very common keywords that identify some of the emails in those categories. For example, he guessed that "love", "baby", and "babe" might help to list a set of romance scams. Alice tested his hypothesis by forming a query and searching each of these keywords separately using *Beagle*'s filters and examining the content of the resulting emails in the communications panel. During the examination for each type of scam, he encountered some related keywords that struck him as good candidates to capture a broader set of those scams. For example, the "charming" keyword showed up in several romance scam emails and appeared to help to identify romance scam emails that the initial keywords (i.e., "love", "baby", "babe") could not.

By querying the new keywords, Alice observed several instances of romance scams that he did not encounter before. He also found emails that included the keyword "charming" but were not typical romance scams. More specifically, Alice noticed a new type of scam in which escorts were the target of an advance-fee fraud. Alice named this new method a *escort girl scam*. Using a similar process, Alice identified novel scams including *babysitter* (a scam that targets babysitters) and *mystery shopper* (a scam that targets job seekers) scams. These were tracked over time. Those that were not long-lived must not have been successful, but those that were performed over a long period of time must have been more successful. Additionally, actual profits were estimated by performing custom searches in *Beagle* .

Another new type of scam that was discovered by a similar process was the *real estate man-in-the-middle* attack. In this scam, the scammer infects the computer of a real estate agent and adds a forward rule to the email account of a victim with the goal of sniffing all email communications. From that moment, the scammer would be able to listen to all the conversations between the real estate agent and his clients. The real victims of this scam are the clients who get fake *closing instructions* to wire money to the scammer.

As a result of identifying this type of scam, all four real-estate agents who were under this attack were notified and instructed how to remove the malware from their computers as well as the forwarding rules from their email accounts. Since the knowledge of Alice and other investigators is now embedded into *Beagle* using the tagging mechanism, the system will continually identify the example of emails that are related to the discovered scams and enables investigators to take appropriate actions.

6.3 Understanding the Scamming Business Models

Understating the business model of online crimes is a prerequisite to identifying their bottlenecks as well taking actions to cripple the criminal organizations. The investigators came up with some relevant questions about whether scammers specialize in one type of scam or their activities evolve over time. Also on whether they run one kind of scam or several types at the same time. Initially, Alice and three other investigators approached the problem using a process similar to what we described in Section 6.2. However, after some time they realized that tagging emails and scammers would enable them to contribute necessary information to answer their main questions. More precisely, after having labelled a sufficiently large number of objects they would be able to use the newly collected information to develop useful statistics about the evolution of scamming activities.

This is the motivation for adding the tagging feature to *Beagle*, as described in section 4.1.5. After the tagging feature was added, the investigators stored it and were able to share the pieces of information they discovered while using this information to build a consensus. More specifically, by having enriched data with their tags, the investigators managed to export the data and calculate statistics about the temporal evolution of scamming activities which were analyzed to better understand the scammers' financial models.

Understanding the business model of the scammers is key to fighting back. Agari used the knowledge they gained to strengthen their security controls to better protect their clients and, by collaborating with the financial community, to shut down as much of the money network as possible in an effort to hurt the scammers.

7 Lessons Learned

During this project, we learned many lessons we believe are potentially useful to a large number of visual analytics projects. These lessons come from our struggle with developing the right tool for the

right people as well as from surprising suggestions and comments we received from our collaborators. The lessons we have learned include both new insights we deem useful to share with the visual analytics community (L2, L3, L4, L6) as well as some established knowledge we intend to corroborate with additional details and further support for their value (L1, L5).

L1: Analysis goals and tasks should have primacy over data types. One of the major mistakes we made during the early phases of our project was to insist on framing our problem as a social network analysis problem due to the network nature of the data. In retrospect, it may seem obvious that analyzing the actual network of all accounts in an email data set is not particularly useful: most nodes are irrelevant; the main task is not to find communities of actors; and investigators are not interested in an overview of the network. Rather, they want to start from a specific query. From our experience, this problem is more insidious than it may seem at first. The issue of finding an appropriate visualization for a given problem is often framed as one of finding the right representation for the data at hand. For instance, Shneiderman's seminal paper "A Task by Data Type Taxonomy for Information Visualizations" [25] proposes to choose visualization according to data type which agrees with much of the existing educational material. We conjecture that an excessive focus on thinking about visualization design as the problem of finding the right representation for the data at hand can lead designers astray. A much better mindset is one in which analysis goals and tasks have primacy over data formats. Framing our problem as a network analysis problem disregarded the fact that analysts are only interested in specific subsets of data and that they are only moderately interested in who is connected to whom. Similar problems can happen in a myriad of cases in visualization, e.g., geographical data where spatial information is not relevant for the task or temporal data in which time is not crucial for the analysis.

L2: Familiarity and simplicity should be given more relevance. Another problem we have encountered in our project is the way in which people respond to complex interfaces and visualizations. Our initial design explorations contained many complex solutions, mostly trying to compress as much data as possible in a single interface. As a consequence, they often led to disorientation. As our project unfolded, we progressively simplified our user interface and the visualizations it contained and subsequently noted that we started receiving better feedback from our collaborators. There seems to be a significant trade-off between the power of a visualization to handle complex information and the need for simplicity and familiarity. Visual analytics systems often contain numerous interactive views in a single interface and many of these views make use of complex visual representations. We conjecture that it is not evident that more complex solutions, through powerful, ultimately benefit the target users of a visual analytics system. In particular, when users are accustomed to using a particular solution it seems advisable to assume there is a high cost to designing solutions completely different from those that already exist. A more conservative approach may lead to adoption while still providing significant improvements in performance. We propose that familiarity and simplicity should be given more relevance in the design of visual analytics systems and, similarly, that complexity should be viewed as a potentially high cost to take into account.

L3: Query-first can trump overview-first. One of the breakthroughs we had in this project was realizing the importance to the investigators of starting their analysis from a specific set of queries. As mentioned before, all analyses begin from some pre-existing knowledge which analysts use to leverage and bootstrap the analysis. This is another area where common guidelines used in visualization may not apply. In particular, the *overview-first* rule, often used as a significant inspiration for visualization design, does not seem appropriate for our case. Investigators do not want to obtain an overview of the whole dataset, they want to have their questions answered and be able to move from one question to the next effortlessly. It is also fair to say, however, that gaining an overview of the results obtained *after* having issued a query, turned out to be a compelling strategy. Similar to the concerns raised by Hornbaek and Hertzum [10], there seems to be a need to better characterize the *overview-first* notion and also to better understand within which circumstances it is an advisable solution. If such a guideline is interpreted as always striving to visualize as much as possible of a given data set, it does not seem to be particularly useful in cases where gaining an overview is not relevant to accomplish the stated goal of a data analysis problem.

L4: Visual analytics can be used to generate (rather than analyze) data. The most surprising idea we have encountered in this project is the one that visual analytics tools can, in some cases, be used to generate generate and analyze data. When our collaborators suggested that we implement a function to help them tag individual entities in the data we realized that certain interactions lead to data enrichment. In turn data enrichment can amplify the analytical capabilities of a team. This is precisely what happened in *Beagle*: the act of annotating email accounts and terms enabled the team to use the annotations for data querying and further analysis. We believe this is a somewhat under-explored direction for visual analytics research which should probably be investigated further through novel exploration and systematic analysis of its capabilities.

L5: It is important not to overlook opportunities for collaborative data analysis. While developing *Beagle* we never considered the idea of creating functions that would enable collaborative data analysis. However, once our collaborators suggested we should implement a method to permit them to share the codes they created, it seemed obvious that such functionality would be of great help. Visual analytics is often seen as providing a tool for isolated users when in fact they tend to be used and adopted in complex data ecosystems where people rarely work in isolation. While many efforts exist to make visual analytics more collaborative [9, 11] we deem that it is important to explicitly consider possible untapped opportunities in applied visual analytics projects. In addition, it seems important to better understand how such types of collaborative environments should work. In our case, collaboration happens only by sharing codes generated by all participants, not by making the whole application a shared environment. More research on understanding how collaboration can and should happen may help design more effective visual analytics solutions.

L6: Goals are more important than insights. One final lesson we learned in this project is the relevance of goals over insights. In business settings such as the one we considered in this project we learned that not all insights are relevant. Of more relevance ultimately is whether a given insight leads someone to take a specific action or make an important decision. Visual analytics has a long tradition of using *insight* [22] as the main factor to judge the quality of a solution, but insights do not seem to be particularly valuable if they are not connected to an important goal. Our collaboration with Agari enabled us to experience this difference first-hand. Data explorations were deemed relevant only when instrumental to the stated goals, not as an intrinsically valuable activity. Furthermore, insights are highly dependent on domain knowledge and expertise. One important observation we drew from observing how analysts analyze their data is that most of the knowledge necessary to produce useful information is in the analyst's head, not in the data. In light of this observation, it seems important to develop models that explicitly consider the role of existing knowledge and mental models in the data analysis process.

8 Conclusion

We presented a case study about a visual analytics solution for investigative analysis of scams. The project enabled us to present unique challenges faced in this context as well as a series of lessons we have learned in this process. We believe these lessons as well as the tool to be generalizable to several other situations. In particular, *Beagle* can be used in any other investigative data analysis problem involving time-stamped, text-based, communication data involving human actors (e.g., sms and chats). At the time of writing *Beagle* is being adopted by another group of investigators in a government-based fraud detection agency and there are plans to expand its use to several other organizations.

REFERENCES

[1] N. L. Beebe, J. G. Clark, G. B. Dietrich, M. S. Ko, and D. Ko. Post-retrieval search hit clustering to improve information retrieval effectiveness: Two digital forensics case studies. *Decision Support Systems*, 51(4):732–744, 2011. 3

[2] N. L. Beebe and L. Liu. Clustering digital forensic string search output. *Digital Investigation*, 11(4):314–322, 12 2014. 3

[3] D. E. Denning. Framework and principles for active cyber defense. *Computers & Security*, 40:108–113, 2014. 2.1

[4] J. Diesner and K. M. Carley. Exploration of communication networks from the enron email corpus. In *SIAM International Conference on Data Mining: Workshop on Link Analysis, Counterterrorism and Security, Newport Beach, CA*. Citeseer, 2005. 3

[5] E. R. Gansner and Y. Koren. Improved circular layouts. In M. Kaufmann and D. Wagner, editors, *Graph Drawing*, pages 386–398, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 4.1.2

[6] J. Haggerty, S. Haggerty, and M. Taylor. Forensic triage of email network narratives through visualisation. *Info Mngmnt & Comp Security*, 22(4):358–370, 10 2014. 3

[7] J. Haggerty, A. J. Karran, D. J. Lamb, and M. Taylor. A framework for the forensic investigation of unstructured email relationship data. *International Journal of Digital Crime and Forensics*, 3(3):1–18, 2011. 3

[8] M. A. Hearst. *Search User Interfaces*. Cambridge University Press, first edition, 2009. 2.2.1

[9] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. *Information visualization*, 7(1):49–62, 2008. 7

[10] K. Hornbæk and M. Hertzum. The notion of overview in information visualization. *International Journal of Human-Computer Studies*, 69(7-8):509–525, 2011. 7

[11] P. Isenberg and D. Fisher. Collaborative brushing and linking for co-located visual analytics of document collections. In *Computer Graphics Forum*, volume 28, pages 1031–1038. Wiley Online Library, 2009. 7

[12] T. G. Jenny Rose Finkel and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, 2005. 4.1.3

[13] M. E. Joorabchi, J.-D. Yim, and C. D. Shaw. EmailTime: Visual Analytics of Emails. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 233–234. IEEE, 2010. 3

[14] Y.-A. Kang, C. Gorg, and J. Stasko. How can visual analytics assist investigative analysis? design implications from an evaluation. *Visualization and Computer Graphics, IEEE Transactions on*, 17(5):570–583, 2011. 3

[15] B. Kerr. Thread arcs: An email thread visualization. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 211–218. IEEE, 2003. 3

[16] A. Kulkarni and T. Pedersen. Name discrimination and email clustering using unsupervised clustering and labeling of similar contexts. In *IICAI*, pages 703–722, 2005. 3

[17] H. Li, D. Shen, B. Zhang, Z. Chen, and Q. Yang. Adding semantics to email clustering. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 938–942. IEEE, 2006. 3

[18] Z. Liu, J. Kihm, J. Choo, H. Park, and J. Stasko. Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw. *Transactions on VisualizationN and Computer Graphics*, X(Y), 2013. 3

[19] S. Martin, B. Nelson, A. Sewani, K. Chen, and A. D. Joseph. Analyzing behavioral features for email classification. In *CEAS*, 2005. 3

[20] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 2009. 5.1

[21] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1604–1613, 2014. 4.1.5

[22] P. Saraiya, C. North, V. Lam, and K. A. Duca. An insight-based longitudinal study of visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1511–1522, 2006. 7

[23] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, Dec 2012. 5

[24] J. Shetty and J. Adibi. Discovering important nodes through graph entropy the case of enron email database. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 74–81, New York, NY, USA, 2005. ACM. 3

[25] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The Craft of Information Visualization*, pages 364–371. Elsevier, 2003. 7

[26] A. Train. The Spanish Prisoner. `http://www.hidden-knowledge.com/funstuff/spanishprisoner/spanishprisoner1.html`, 1910. [Online; accessed 08-Feb-2018]. 2