Volume 7, Issue 1
February, 2014
ISSN: 1946-1836

Journal of
Information Systems Applied Research

In this issue:

The **Journal of Information Systems Applied Research** (JISAR) is a double-blind peer-reviewed academic journal published by **EDSIG,** the Education Special Interest Group of AITP, the Association of Information Technology Professionals (Chicago, Illinois). Publishing frequency is currently quarterly. The first date of publication is December 1, 2008.

JISAR is published online (http://jisar.org) in connection with CONISAR, the Conference on Information Systems Applied Research, which is also double-blind peer reviewed. Our sister publication, the Proceedings of CONISAR, features all papers, panels, workshops, and presentations from the conference. (http://conisar.org)

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the JISAR journal. Currently the target acceptance rate for the journal is about 40%.

Questions should be addressed to the editor at editor@jisar.org or the publisher at publisher@jisar.org.

Journal of Information Systems Applied Research

The De-Escalation of the DHS HSIN Next Gen Project

Alan F. Rosenhauer
Afr2k@mtmail.mtsu.edu

Melinda Korzaan
Melinda.Korzaan@mtsu.edu

Computer Information Systems
Middle Tennessee State University
Murfreesboro, TN 37132, USA

Abstract

In the eight years since its creation, the US Department of Homeland Security (DHS) had tried to provide a platform for the federal government to share sensitive but unclassified (SBU) information among its varied mission partners. These partners include federal agencies and state and local public safety and law enforcement officials. Its third iteration was under development and was behind schedule, over budget, and was not garnering the support from either management or the user community. The US Office of Management and Budget (OMB) had halted any additional spending on the project. The existing course of action was not acceptable and de-escalation was required. A review of the project led DHS to cancel the project, re-scope the work, and start over. This case study examines the process of de-escalating the project by mapping the de-escalation phases of DHS Homeland Security Information Network (HSIN) Next Gen into an established research framework (Keil & Montealegre, 2000). The study confirms the practical application of Keil and Montealgre's de-escalation framework and provides insights for practitioners from the case's lessons learned.

Keywords: DHS HSIN Next Gen, project management, failed projects, de-escalation.

1. FAILED PROJECTS

Project Failure

Imagine dreaming of a new house, taking the time to draw up blueprints, buy the land, and hire a contractor. You expend time and money digging the foundation, framing the structure, finishing the interior, and landscaping the outside. Then, after all of that work, you decide that you really do not want a new house so you tear down the new house and leave an eye sore of a broken foundation behind for all of the neighbors to see. This may seem like a silly example in the context of building houses but it happens all too frequently when building software applications.

The list of software projects that never reach production is staggering. The US Federal government spent $4 billion for a new IRS computer system and never used it (Charette, 2005). The US Federal Aviation Administration spent $2.6 billion on a new air traffic control (ATC) system and cancelled the project before it went to production (Charette, 2005). The FBI had the Virtual Case File (VCF) system built for a total project cost of $581 million and never used it. The VCF contained over $105 million in unusable code (Goldstein, 2005).

Project failure is not limited to the federal government. After seven years of development, the state of Tennessee cancelled a new health department system. The project had cost nearly $20 million (Gonzalez, 2013). After spending $200 million on a new purchasing system, Ford Motor Company terminated its Everest system (Sherriff, 2004).

Unlike a spectacular failure in the civil or mechanical engineering realm, failures in the computer software discipline often go nearly unnoticed. On July 1, 1940, the Tacoma Narrows Bridge opened near Tacoma, Washington. At that time, the bridge was the third longest suspension bridge in the world. A little over four months later, on November 7, 1940, a 42 mile per hour wind caused the bridge to oscillate and collapse (Billah & Scanlan, 1991). News reporters captured the dramatic collapse on film. For the past 70 years, educators have used the film as a teaching tool and shown it in nearly every high school and college physics class. In 2013 dollars, the bridge would cost approximately the same as the unused FBI VCF code ($106 million). The resulting analysis of the bridge failure showed the root cause of the problem. The solid steel beams did not allow the proper flow of air around or through the structure. The state of Washington devised a solution to the problem and built a new bridge at the same location. The new bridge has been standing and handling vehicular traffic for over 60 years.

As the software industry matures, it must examine the failures and identify the root cause or causes of the failures. A survey of IT projects shows that 18% of the projects failed to deliver the desired outcome or the organization terminated the project before release (PMSolutions, 2011). The same survey showed that 25% of the projects were at risk but the organization was able to recover the project.

Sometimes an organization can recognize the indicators of pending failure. During the construction of the Tacoma Narrows Bridge, the workers noticed that the road surface moved on windy days. The workers affectionately called the bridge “Galloping Gertie” (WS DOT, 2005). Engineers were working on a fix for the “fluttering” of the bridge and on the morning of the collapse were working on obtaining quotes for a solution for the instability. Instead of realizing that the unusual bridge movement was a warning sign for bridge failure, the engineers thought was a minor issue and that they could fix the problem after the bridge was in use. They simply ran out of time.

In the computer world, potential problems are often visible long before the project fails. These warning signs can prompt the project manager to take action before the project comes crashing down around them.

The De-Escalation Process

When a project exhibits warning signs, like a missed deadline or insufficient stakeholder involvement, the project manager must decide between two paths: escalation or de-escalation. Escalation is defined as “continued commitment to a previously chosen course of action in spite of negative feedback” (Keil, Mann, & Rai, 2000). In contrast to escalation, de-escalation is a “reduced commitment to a failing course of action” (Montealegre & Keil, 2000). The goal of de-escalation is to rescue the project and to produce a viable and useful product. The rescue may include a radical re-scoping or a redefinition of the project itself (Montealegre & Keil, 2000). However, sometimes the project manager is unable to salvage the project and must terminate the project. Previous research discusses the process to decide between escalation and de-escalation (Staw, 1976; Keil, 1995; Keil, Mann, & Rai, 2000; Lunenburg, 2010). Studies have also emphasized the importance of organizations being aware of effective de-escalation strategies to prevent future projects from escalating out of control and unnecessarily wasting valuable resources (Pan, Pan, & Flynn, 2004).

Similar to the series of steps during the start of a project, projects often follow a defined series of steps during de-escalation. Research by Keil and Montealegre (2000) proposed a four-phase process for de-escalating a project: 1) problem recognition, 2) reexamination of prior course of action, 3) search for alternate courses of action, and 4) implementation of an exit strategy.

The remainder of this paper presents a case study into the de-escalation of the DHS HSIN Next Gen project. The researchers gathered the information for this case study through interviews with current and former DHS employees, email correspondence from the project, and a review of public records available on the Internet. See Appendix 1 for a list of the interview questions.

The information presented will identify some of the project's escalation warning signs, detail the actual de-escalation process following the four phases of the Keil and Montealegre (2000) model, and provide insights into specific lessons learned from the project.

2. BACKGROUND

The United States federal government created the Department of Homeland Security in response to attacks of September 11, 2001. One of the stated functions of DHS was to provide a mechanism for the federal government to share information with state and local authorities. (Homeland Security Act of 2002) Many of the existing agencies that merged into DHS were already sharing information with state and local officials but these different sharing capabilities led to silos of information. It was this compartmentalization of information that kept law enforcement from identifying the plans of the 9/11 terrorists. DHS needed a platform that would allow the sharing of information across both the levels of government (federal, state, and local) and across the type of government (law enforcement, immigration, intelligence). In order to expedite the deployment process, DHS began looking for an existing system to meet their needs.

The Joint Regional Information Exchange System (JRIES) was an information sharing system that was born out of a specific need to share information between the California Anti-Terrorism Information Center (CATIC), the New York Police Department, and the Defense Intelligence Agency (DIA). DHS decide to adopt the JRIES system as their means to share sensitive but unclassified (SBU) information. At the time, the JRIES board of directors welcomed the addition of DHS to the program. The JRIES system used Microsoft's Groove application to share documents and allow collaborative editing of those documents. It also used the open source Jabber software for instant messaging. While JRIES satisfied the need for document sharing and instant messaging, it did not include any other collaboration tools. In September of 2003, DIA transferred control of JRIES to DHS and in February of 2004, DHS renamed JRIES to HSIN. DHS quickly expanded the JRIES membership to include members from all 50 states. With the increased number of users, the JRIES platform started to suffer performance problems. In order to handle the increased workload and to satisfy additional customer requirements, DHS converted the HSIN site to Microsoft SharePoint 2003 in March of 2005.

Unrelated to its technology decisions, the HSIN program started to experience issues with fulfilling its stated mission goals. In fact, the program had critics inside DHS, in Congress, and in the anticipated user community. In May of 2005, the JRIES board of directors voted to discontinue their relationship with HSIN. They cited concerns over the changes DHS was making without the input from the affected stakeholders. Law enforcement personnel expressed confusion between the seemingly overlapping missions of the FBI's Law Enforcement Online (LEO) and the Regional Information Sharing Systems (RISS) RISSNET services. By January 2006, DHS mandated that all components of DHS were to use HSIN for its information sharing initiatives. About that same time, the DHS Office of the Inspector General (OIG) conducted an audit of HSIN. In its June 2006 report, the OIG reported that DHS did not clearly define HSIN's role, that HSIN's efforts to solicit input from all HSIN user communities were "inadequate", and that it did not clearly define its relationship with other information sharing systems. Also in 2006, 13 US Representatives issued a report identifying 33 unfulfilled promises from DHS. Of the 33, three specifically referred to HSIN. In May of the following year, Congress held a hearing concerning HSIN. At that hearing, Rep. Jane Harman (CA) stated that after three years, instead of having a "robust system", HSIN was "kind of a mess" (US House of Representatives, 2007).

The Next Generation

By the fall of 2007, pressure on DHS to fix the HSIN program caused it to look for alternatives. In October of 2007, DHS decided to upgrade HSIN to a new platform and to include new capabilities. DHS named this new version HSIN Next Generation or HSIN Next Gen. DHS referred to the currently deployed instance of HSIN as HSIN Legacy. The project consisted of four phases or spirals. (US Government Accountability Office (GAO), 2008) In the first phase, HSIN Next Gen will establish an operational platform for 20,000 new users from the critical infrastructure user community. The next phase begins the migration of the existing HSIN Legacy users to the new platform. The third phase completes the migration of users and phase four provides improved content management, better information discovery and delivery, and improved notification capabilities.

Key Target Dates	
May 2008	Project Start
August 2008	Phase 1 Complete
May 2009	Phase 2 Complete
September 2009	Phase 3 Complete
November 2009	Phase 4 Complete

Table 1

See Appendix 2 for the complete timeline.

The Advisory Committee

Shortly after the announcement of HSIN Next Gen, DHS created the HSIN Advisory Committee (HSINAC). The initial HSINAC meeting took place at the end of October 2007 (HSIN Advisory Committee, 2007). At this meeting, a HSIN representative listed two important items. DHS did not have a "preordained path" for HSIN and that the HSINAC would not be briefed on the Next Gen project. Their intention was to keep the deliberations of the HSINAC unbiased. A HSIN representative stated that the HSAINAC should not focus on technical requirements but instead focus on policy and governance issues. The HSINAC recommended that DHS create a governance board consisting of federal, state, local, and tribal partners. The purpose of this board would be to define business processes and workflows. At the conclusion of this initial meeting, the HSINAC recommended that HSIN become the "one-stop shop" for unclassified information sharing. This all-encompassing scope would prove to be problematic for HSIN. At the second HSINAC meeting, the committee recommended that DHS create a Configuration Control Board (CCB) to manage the process of gathering requirements. At the third meeting in July 2008, a committee member stated his concern that the development was proceeding at a rapid pace without the proper management and control procedures (HSIN Advisory Committee, 2008). DHS assured the committee that the proper control measure would be in place 6 months into the project and that full management controls would be in place by July 2009 when the program was scheduled for deployment.

The HSIN Next Gen Project

DHS was experiencing their own growing pains and the Next Gen project was a victim of those difficulties. The creation of DHS was a significant undertaking that included both the establishment of a new department but it also included the reassignment and restructuring of many existing federal government agencies.

At the same time that DHS was proposing the Next Gen project, the GAO was faulting DHS for not having a full set of management controls in place for acquisitions (GAO, 2008). The GAO report specifically faulted DHS for not having a program office and for not identifying staff roles and responsibilities, for not having established a process to gather, analyze, and validate user requirements, and finally for not having a risk management plan in place. DHS staff used their own aggressive schedule as justification for proceeding without the controls in place. Additionally, DHS had not published a departmental System Life Cycle (SLC) framework and it did not complete its product acquisition policies until more than two years after the project went out for bids.

The Next Gen project also included the consolidation of 28 other web portals deployed within DHS. Each web portal had its own unique user community, workflows, and requirements. The HSIN staff at that time consisted of an average of five full-time federal employees. HSIN staff needed to hire an outside consultant/contractor to design, develop, and deploy the system. Throughout the development of Next Gen, the HSIN full-time staff experienced significant turnover with only two full-time staff remaining in the same job for the entire project, one in management and one support staff. The team also had one contract worker converted to a full-time federal employee.

The bid process took just nine months and only two vendors responded. DHS awarded General Dynamics with the contract in May of 2008 with an initial budget of $18M and a potential five-year value of $62M if DHS exercised all the options. The Next Gen project envisioned a brand new platform with state-of-the-art technology. Unfortunately, that vision did not pan out. The contract with General Dynamics did not contain the specificity required in a project of this size. Instead of focusing on specific use cases needed for each user community, the Next Gen requirements included a series of generic capabilities and features and did not include specific information sharing processes and workflows.

3. MAPPING THE DE-ESCALATION PROCESS

Although the HSIN team did not specifically model their de-escalation on the model proposed by the Keil and Montealegre (2000), the steps of their de-escalation mapped well to the model.

Step 1: Recognizing the Problem

The first step in the Keil and Montealegre (2000) de-escalation process is the recognition of a problem. This may take the form of negative feedback about the project or it may include external pressure on the project. The Next Gen project had many entities questioning its chance for success.

Problems Arise with Next Gen

The contractor had barely begun the process of creating HSIN Next Gen when problems started to arise. In July 2008, two senior members of the US House of Representative sent a letter to DHS Secretary Chertoff (Lipowicz, 2008). They asked the secretary to halt all work on HSIN Next Gen until the program's state, local, and tribal users had defined and validated all the requirements. The representatives felt that DHS left the non-federal users out of the requirements gathering process and that DHS had not identified the needs of the non-federal users. In its response, DHS defended its procurement process and stated that the requirements addressed the needs of state, local and tribal user, but ultimately DHS did not change the requirements nor did they solicit additional input from its partners in state, local, and tribal organizations. The representatives were concerned that HSIN Next Gen was repeating some of the mistakes of HSIN Legacy.

Methodology Questioned

In addition to the poor requirements, HSIN staff felt that General Dynamics focused too much on the technology and not enough on the mission. The General Dynamics team started with a variety of off-the-shelf software products and then customized each of them to meet the needs of HSIN. They selected Oracle for identity management, EMC Documentum for their content management system, RSA for two-factor authentication, and Adobe Connect for conferencing and instant messaging. While each of these products is a quality application, the development team struggled to get all of the parts to work together. The development team had to contend with external requirements that were difficult to incorporate into the suite of products. The RSA solution chosen by General Dynamics used one-time-passcode tokens for two-factor authentication. However, Homeland Security Presidential Directive number 12 (HSPD-12) mandated the use of Personal Identity Verification (PIV) cards for all government employees and contractors. The directive required that all agencies issue the new cards by October 2008. DHS did not meet that deadline. In 2010, they had still not met the directive. Since the RSA tokens were not the stated direction for authentication, DHS never issued the cards and thousands of them sat unused in storage.

HSIN hacked

In the middle of the development of the Next Gen version, the Legacy version of HSIN suffered two attacks by hackers, the first in March 2009, and the second in April 2009 (Lipowicz, 2009). The attack forced DHS to shift resources from the new system to bolster security on the old system. It also reinforced the need to replace the Legacy system and to implement two-factor authentication on Next Gen.

The Schedule Slips

The initial project plan called for a HSIN Next Gen deployment in November 2009. The HSIN team did not meet that deadline. Part of the platform was available for use but most of the required components were not available for use. Only one group of users had been migrated to the new platform and most users could not be migrated until the remaining capabilities were available. Interoperability with LEO and RISS was not functioning.

Groupthink

The HSIN team had a significant turnover and many of the General Dynamics team members had a longer tenure on the project. Subsequently, the HSIN team did not feel they had the authority to question publically some of the decisions or even question the overall viability of the Next Gen project. For much of the project the contractors outnumbered the federal staff. DHS delegated or abdicated many important policy and direction decisions to General Dynamics. To complicate matters more, General Dynamics sub-contracted some of the work creating even more layers of bureaucracy.

Work Stops

Others noticed the delays in the Next Gen project. The DHS Inspector General reported that even groups within DHS were not using

HSIN. Many DHS Fusion Centers reported that they stopped using HSIN because of the limited content and the lack of regular updates to the information. When HSIN purged Fusion Center accounts that had not been used in six months, the number of accounts dropped from 7,000 to 1,000. The Office of Management and Budget (OMB) reviewed the HSIN program in early 2010. OMB designated the HSIN program as a high risk and ordered a stop to all development work. OMB then conducted a review to determine if the program would receive any additional federal funding. The review found that HSIN was a viable program but OMB added conditions to any additional funding. The system must improve its interoperability with other systems, expand its user base, and accelerate the consolidation of the other DHS portals. The OMB review also identified a problem with the ownership of the Next Gen project. The report faulted DHS for having the DHS Office of Operations Coordination and Planning (OPS) run both the HSIN program and the Next gen project. DHS OPS was not an IT-based organization. The management of the Next Gen project transitioned to the DHS Office of the Chief Information Officer (OCIO) while the overall project remained under the control of DHS OPS.

Step 2: Reexamining the Present Course of Action

The second step in the de-escalation reexamines the decisions and plans of the current course of actions (Keil & Montealegre, 2000). This step requires management to look at the project objectively and to analyze the available information. The various stakeholders may try to pull the project in different directions. Some stakeholders may want to stay the course, while others will want to change or cancel the project. This step requires the project manager to redefine the project based on the latest information.

Decisions

The new OCIO staff had to make some decisions. The current implementation was failing and they needed to identify an alternate plan of action. They needed to analyze the situation and determine a root cause. Once they identified the root cause, or causes, they needed to decide on an escalation path or a de-escalation path. If the program was failing due to a lack of budget or staff, escalation might be the solution. If the problem was a process problem, cancelling the project may be the best course of action. Ideally, an outside consultant would look at the problem objectively. Based on the OMB funding stoppage, that was not possible. Instead, the OCIO staff created a "tiger" team to look at the problem. In the DHS parlance, a tiger team is an ad hoc group created for a single purpose and would focus using the "eye of the tiger." The team consisted of the newest members of the staff, because management felt that the newer team members would have less emotional attachment to the current solution and would thus be more objective.

The team started by identifying the required capabilities of the program. DHS had given an initial set of requirements to the General Dynamics. However, those requirements did not encompass the complete set of needs from all of the user communities and it did not include the new requirements added during the development. The tiger team analysis identified 61 operational capabilities that the system must support. The team then looked at the capabilities of the existing systems. The original HSIN Legacy system met 84% of the operational capabilities. Another existing portal, HSIN State and Local Intelligence Community of Interest (SLIC) met 35% of the capabilities. The team also found that the new Next Gen application met only 51% of the operational capabilities.

The team also found that if DHS consolidated the 28 different portals spread throughout DHS, DHS would save an additional $50M a year. The tiger team felt that the HSIN program provided a needed resource to their user community and a properly planned and executed upgrade would save the department money in the end.

The team sent their results to the HSINAC. The HSINAC accepted their results and recommendations. The tiger team then forwarded their results to DHS management. During the interviews, HSIN Staff indicated that a DHS independent verification and validation (IV&V) review corroborated the recommendations of the tiger team.

Step 3: Searching for Alternate Courses of Action

The purpose of this step is for the project manager to minimize the damage associated with the current plan and to develop an alternate plan of action (Keil & Montealegre, 2000). The project should rely on independent analysis of both the current plan and the proposed course of action.

A New-New Direction or a New-Old Direction

Identifying that the HSIN program should continue was only half of the story. The tiger team needed to identify a new course of action. When the HSIN team envisioned the Next Gen program, they focused on creating a new platform to share information between the different user communities. The team realized this was an ill-conceived vision. DHS had 28 portals that were already sharing information. The focus of the HSIN program should be on efficiently consolidating the existing portals with the hopes of sharing information among the different groups not just within each of the groups.

DHS initiated the Next Gen project as a new development effort. The tiger team recommended that HSIN not look at a new development project but instead look at a new version of the old HSIN Legacy system. With the Legacy system scoring higher than Next Gen, the tiger team recommended that HSIN just upgrade Legacy to Microsoft SharePoint 2010 and include the enhanced security of Next Gen. They felt this was the best course of action.

Step 4: Implementing an Exit Strategy

The final step in the de-escalation is the implementation of an exit strategy (Keil & Montealegre, 2000). The project manager must inform the stakeholders of the change in the project plan and then execute the closing of the old project.

Cancelling the Project

The HSIN staff was relieved that the Next Gen project was closing. When the GAO halted their funding, most of the staff focused their energy on the Legacy platform. They also took the opportunity to create a formal requirements document for the HSIN platform.

Unfortunately, DHS could not just turn off the Next Gen platform. The HSIN team had already moved a group of users from FEMA to Next Gen. In addition, HSIN staff had not tested the software for enabling the interoperability with LEO and RISS. This software was part of the enhanced security from Next Gen they wanted to implement in Legacy. However, HSIN could not afford to keep both Legacy and Next Gen running while they built a new version. The team came up with a hybrid plan. The first part of the plan included the migration of the FEMA users to the Legacy platform. The second part consisted of using Next Gen to test the interoperability software with LEO and RISS.

Next Gen Shut Down

DHS officially shut down the Next Gen platform in July 2011. The HSIN team re-scoped the project and work on the next version, HSIN-R3, started in October 2011. HSIN-R3 would not be a new development but instead be a technology refresh where the project team upgraded HSIN Legacy to Microsoft SharePoint 2010, incorporated the improved security features from Next Gen, and included the consolidation of the first ten of the 28 portals.

4. LESSONS LEARNED

Feedback from the interviews provided insight into the lessons learned from the project. The Next Gen project suffered problems from the outset. DHS had a flawed approach to the original HSIN program. DHS took an existing application, JRIES, which was serving a specific community, took complete control of the system, and then alienated the users. In retrospect, JRIES was another portal that they should have consolidated onto a common platform. The users faulted HSIN more with the content of the site than with the technology but the Next Gen project focused on the technology not on the content.

In their effort to provide an all-encompassing SBU portal, DHS focused too much on the technology and not enough on the mission. The contract with General Dynamics was rushed and DHS did not fully vet the requirements with the diverse user communities. HSIN staff felt that the technology was the driver, not the mission. HSIN tried to be the only SBU portal for all of government. Later, they realized that LEO and RISS had a different mission and a different user base. HSIN spent too much effort on those other missions instead of focusing on their users.

The transfer of the Next Gen project from DHS OPS to the OCIO was a necessary action. The OPS office did not have the technical expertise to oversee a development project of that magnitude. The OCIO staff had a departmental-wide purview and ensured that the HSIN program technology aligned with the broader

DHS goals. In addition to the HSINAC that provided input from external sources, DHS created an Executive Steering Committee that set the overall direction of the program. This guidance once again ensured that the program met departmental-wide goals and objectives.

The final lesson learned dealt with user involvement. Most of the targeted user committees felt little or no ownership in the program. DHS usually determined the schedules, requirements, and designs without sufficient input from the users. The Fusion Centers reported that while DHS spent time and effort on the technology, they failed to use the system because of the untimeliness of the data. (DHS OIG, 2010) DHS was working on a technical project when the users needed a content project.

5. CONCLUSIONS

According to Kappelman, McKeeman, & Zhang, (2006), the warning signs for a failing project fall into two categories: people oriented and process oriented. While some of the failings in the Next Gen project were people related, most notably lack of stakeholder involvement at the outset of the project, the majority of the failings were process related.

12 Early Warning Signs

People Oriented

- Lack of top management support
- Weak project manager
- No stakeholder involvement / participation
- Weak commitment of project team
- Team members lack requisite skills / knowledge
- Subject matter experts are overscheduled

Process Oriented

- Lack of documented requires / success criteria
- No change control process / management
- Ineffective schedule planning / management
- Communications breakdown between among stakeholders
- Resource assigned to a higher priority project
- No business case for the projects

(Kappleman, McKeeman, & Zhang, 2006)

Table 2

The project did not have sufficient requirement's definition, DHS did not have well-established processes, the schedules were not realistic, the communications between the end users and the project team were insufficient, and the business case for a complete rewrite was not justified.

In his book, Bennatan (2006) recommends that organizations implement an Early Warning System (EWS) to draw attention to potential problems before the problems become unmanageable. Although there were many warning signs, the HSIN team did not have Early Warning System. With an EWS the team might have been able to rescue the project instead of being forced to cancel the project.

6. CONTRIBUTIONS TO RESEARCH

This case study demonstrates that the de-escalation model of Keil and Montealegre (2000) is still a helpful tool in managing a failing project. This contribution confirms the framework continues to be representative of how de-escalation unfolds in practice.

One use of the case study methodology is to help connect academic research to industry practice and this study provides a case example that confirms the application of the de-escalation framework as a useful guide in studying real world projects as they progress through a de-escalation process. Therefore, academic de-escalation theory continues to generalize to modern information systems' projects, and the continued sustainability of the usefulness of project de-escalation academic theory for practitioner application is confirmed.

Future Research

This study also demonstrates that basic project management practices, like requirements gathering and stakeholder involvement, are lacking. Additional research is called for to identify the reasons behind the lack of proper project management.

The lack of effective project management practices reveals a gap that exists between academic project management / project risk management knowledge and industry practice. The top risk factors that led to the escalation problem with the Next Gen project (such as lack of stakeholder involvement and misunderstood requirements) are the same key risks that have consistently been identified in literature (Schmidt, Lyytinen, Keil, & Cule, 2001; Kappelman, McKeeman, & Zhang, 2006). This is especially salient for stakeholder / user involvement, which has been identified as a key factor in information systems application implementation since the 1960s (Barki & Hartwick, 1994). The question that surfaces is what has industry learned from academic literature in project risk management and why do the same key risk factors continue to be problematic? What can be done to address and prevent these risks before they result in project escalation or project failure?

There is a substantial lack of evidence that academic risk management knowledge is being applied to project management in practice (Taylor, Arman, & Woelfer, 2012). Therefore, a need is identified for future research studies to be conducted collaboratively with both academic researchers and practitioners with a goal to not only identify key risks but also formulate appropriate action plans to be taken early to prevent risk factors from escalating and cause troubled projects later in the project life cycle.

7. REFERENCES

Barki, H., & Hartwick, J. (1994). Measuring user participation, user involvement, and user attitude. *MIS Quarterly*, 18(1), 59-82.

Benbasat, I., Goldstein, D., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly*, 11(3), 367-386.

Bennatan, E. M. (2006). *Catastrophe disentanglement: getting software projects back on track.* Boston, MA: Pearson Education

Billah, K. Y., & Scanlan, R. H. (1991). Resonance, Tacoma Narrows bridge failure, and undergraduate physics textbooks. *American Journal of Physics*, 59(2), 118. doi: 10.1119/1.16590

Charette, R.N. (2005). Why software fails [software failure]. *Spectrum, IEEE* , 42(9), 42-49. doi: 10.1109/MSPEC.2005.1502528

Department of Homeland Security (DHS) Office of Inspector General (OIG). (2010). Information Sharing With Fusion Centers Has Improved, but Information System Challenges Remain, OIG-11-04.

Eisenhardt, K., & Graebner, M. (2007). Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, 50(1), 25-32.

Goldstein, H., (2005). Who killed the virtual case file? Retrieved April 29, 2013 from http://spectrum.ieee.org/computing/software/who-killed-the-virtual-case-file/

Gonzalez, T. (2013) State pulls plug on multi-million dollar computer system. *The Tennessean*. Retrieved April 29, 2013 from http://www.tennessean.com/article/20130426/NEWS0201/304260131/

Homeland Security Act of 2002, Pub. L. No. 107–296 § 102. (2002).

HSIN Advisory Committee. (2007). October 31, 2007 to November 1, 2007 meeting minutes. Retrieved March 29, 2013 from http://www.dhs.gov/xlibrary/assets/hsinac_inauguralmtg_2007-1030-1101.pdf.

HSIN Advisory Committee. (2008). July 31, 2008 to August 2, 2008 meeting minutes. Retrieved March 29, 2013 from http://www.dhs.gov/xlibrary/assets/hsinac_mtg_2008-0731-0801.pdf.

Kappelman, L. A., McKeeman, R., & Zhang, L. (2006). Early warning sign of IT project failure: the dominant dozen. *Information Systems Management,* 23(4), 31-36.

Keil, M. (1995). Pulling the plug: Software project management and the problem of project escalation. *MIS Quarterly,* 19(4), 421-447.

Keil, M., Mann, J., & Rai, A. (2000). Why software projects escalate: an empirical analysis and test of four theoretical models. *MIS Quarterly*, 24(4), 631-664.

Keil, M., & Montealegre, R. (2000). Cutting your losses: Extricating your organization when a big project goes awry. *Sloan Management Review*, 41(3), 55-68.

Lipowicz, A. (2009). Information-sharing platform hacked. Retrieved January 21, 2013, from http://fcw.com/Articles/2009/05/13/Web-DHS-HSIN-intrusion-hack.aspx

Lunenburg, F. C., (2010). Escalation of Commitment: Patterns of Retrospective

Rationality. *International Journal of Management, Business, and Administration*, 13, 1-5.

Montealegre, R., & Keil, M. (2000). De-escalating information technology projects: lessons from the Denver International Airport. *MIS Quarterly*, 417-447.

Pan, G., Pan, S., & Flynn, D. (2004). De-escalation of commitment to information systems projects: A process perspective. *Journal of Strategic Information Systems*, 13(3), 247-270.

PMSolutions. (2011). Strategies for Project Recovery. Retrieved April 28, 2013 from http://www.pmsolutions.com/collateral/research/Strategies%20for%20Project%20Recovery%202011.pdf.

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international Delphi study. *Journal of Management Information Systems*, 17(4), 5-36.

Sherriff, L. (2004) Ford dumps $200m Oracle system. Retrieved April 29, 2013 from http://www.theregister.co.uk/2004/08/18/ford_ditches_oracle/.

Simon, A., Sohal, A., & Brown, A. (1996). Generative and case study research in quality management. Part I: Theoretical considerations. *International Journal of Quality & Reliability Management*, 13(1), 32-42.

Staw, B. M., (1976). Knee-Deep in the Big Muddy: A Study of Escalating Commitment to a Chosen Course of Action. *Organizational Behavior and Human Performance*, 16, 27-44.

Taylor, H., Artman, E., & Woelfer, J. (2012), Information technology project risk management: Bridging the gap between research and practice. *Journal of Information Technology*, 27(1), 17-34.

US Government Accountability Office (GAO). (2008). Management Improvements Needed on the Department of Homeland Security's Next Generation Information Sharing System (GAO-09-40). Washington, DC: US Government Printing Office.

US House of Representatives. (2007). Hearing before the subcommittee on intelligence, information sharing, and terrorism risk assessment of the committee on homeland security. (110-34). Washington, DC: US Government Printing Office.

Vissak, T. (2010). Recommendations for using the case study method in international business research. *The Qualitative Report*, 15(2), 370-388.

WS DOT (2005). Tacoma Narrows Bridge. Retrieved April 29, 2013 from http://www.wsdot.wa.gov/TNBhistory/.

Editor's Note:

This paper was selected for inclusion in the journal as the CONISAR 2013 Best Paper The acceptance rate is typically 2% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2013.

Appendices and Annexures

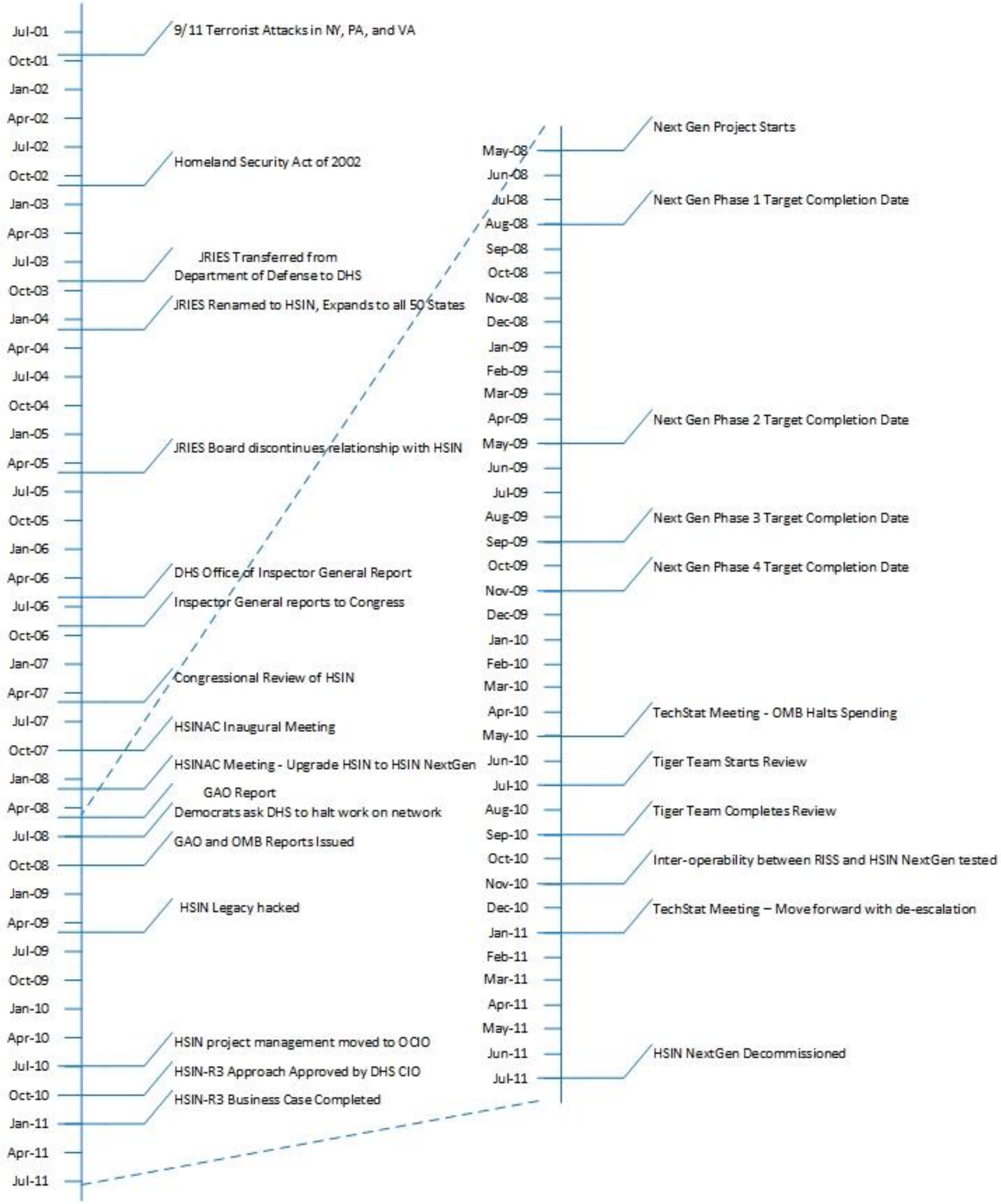
Appendix 1

INTERVIEW QUESTIONS

1. What was your involvement in the DHS HSIN Next Gen project?
2. Describe the process for assessing the project's status.
3. What indicators did you find that showed the project was having difficulties?
4. Describe the steps of de-escalating the project.
5. What was the original expectation of the de-escalation: cancellation, re-directing and re-starting the project, or was any option acceptable? Was the original expectation what actually happened because of de-escalation?
6. How did you inform the team members of the process?
7. What was the team's reaction to de-escalation?
8. What changes did you implement in the re-start of the project?
9. What changes did you make to the overall development process because of the lessons learned in de-escalation?
10. What were the key lessons learned or take-aways from this entire process?

Appendix 2

TIMELINE OF EVENTS


Jul-01
9/11 Terrorist Attacks in NY, PA, and VA
Oct-01
Jan-02
Apr-02
Jul-02
Homeland Security Act of 2002
Oct-02
Jan-03
Apr-03
Jul-03
JRIES Transferred from Department of Defense to DHS
Oct-03
Jan-04
JRIES Renamed to HSIN, Expands to all 50 States
Apr-04
Jul-04
Oct-04
Jan-05
JRIES Board discontinues relationship with HSIN
Apr-05
Jul-05
Oct-05
Jan-06
DHS Office of Inspector General Report
Apr-06
Inspector General reports to Congress
Jul-06
Oct-06
Jan-07
Congressional Review of HSIN
Apr-07
Jul-07
HSINAC Inaugural Meeting
Oct-07
HSINAC Meeting - Upgrade HSIN to HSIN NextGen
Jan-08
GAO Report
Apr-08
Democrats ask DHS to halt work on network
Jul-08
GAO and OMB Reports Issued
Oct-08
Jan-09
HSIN Legacy hacked
Apr-09
Jul-09
Oct-09
Jan-10
Apr-10
HSIN project management moved to OCIO
Jul-10
HSIN-R3 Approach Approved by DHS CIO
Oct-10
HSIN-R3 Business Case Completed
Jan-11
Apr-11
Jul-11
Next Gen Project Starts
May-08
Jun-08
Jul-08
Next Gen Phase 1 Target Completion Date
Aug-08
Sep-08
Oct-08
Nov-08
Dec-08
Jan-09
Feb-09
Mar-09
Apr-09
Next Gen Phase 2 Target Completion Date
May-09
Jun-09
Jul-09
Aug-09
Next Gen Phase 3 Target Completion Date
Sep-09
Oct-09
Next Gen Phase 4 Target Completion Date
Nov-09
Dec-09
Jan-10
Feb-10
Mar-10
Apr-10
TechStat Meeting - OMB Halts Spending
May-10
Jun-10
Tiger Team Starts Review
Jul-10
Aug-10
Tiger Team Completes Review
Sep-10
Oct-10
Inter-operability between RISS and HSIN NextGen tested
Nov-10
Dec-10
TechStat Meeting – Move forward with de-escalation
Jan-11
Feb-11
Mar-11
Apr-11
May-11
Jun-11
HSIN NextGen Decommissioned
Jul-11

Developing a Semantic Differential Scale for Measuring Users' Attitudes toward Sensor based Decision Support Technologies for the Environment

Taiwo Ajani
tajani@ferrum.edu
Computer Information Systems
Ferrum College
Ferrum, VA 24088

Elizabeth Stork
stork@rmu.edu
Organizational Leadership
Robert Morris University
Moon Township, PA 15108

Abstract

The purpose of this paper is to describe the development and testing of an Emerging Technologies Semantic Differential Scale (ETSDS) designed to measure the attitudes of potential users toward an emerging technology. The strategy consisted of identifying initial items and descriptors that may help to understand respondents' attitudes about one emerging technology; test bi-polar adjectives to construct the scale; determine representativeness of items on a particular construct domain for content validity; and finally, to test the reliability and construct validity of the instrument. The instrument development process resulted in a reliable and valid parsimonious 10-item scale for quantitatively measuring attitudes toward the deployment of global sensor networks that is easily adaptable to other emerging technologies with similar attributes. The instrument is likely to be useful to both academics and practitioners with interests in attitudes about innovations, technology adoption, and users' behavioral intention toward emerging technologies.

Keywords: Semantic differential scale (SDS), public health, attitudes, emerging technologies, scale development, environment.

1. INTRODUCTION

Emerging sensor technologies hold promise for creatively addressing modern day threats to public health and other environmental systems, however, promises are hard to deliver when such emerging technologies are poorly or not understood, and by extension, not embraced. The reasons for non-acceptance are complex and include technical and human factors that are critical to advancing selling new technologies to a variety of users. Technical factors include but are not limited to unknown manufacturing costs, form factors, means and timing for deployment, managing data communication, and issues of energy efficiency (Lau et al., 2006). Human factors include perceived usefulness of the technology and behavioral intentions toward it, in addition to knowledge of the technology, the problems it is intended to resolve, and the perceptions of the seriousness of threats. The threat of rejection of new innovations can be mitigated if, at the earliest stages, the attitudes of potential users and decision-makers are understood and addressed (Davis & Venkatesh 2004; Jain, 2006; Venkatesh, Morris, Davis, & Davis, 2003). According to Fishbein (1975) attitude as a concept is important in understanding and predicting both the reaction of people to an entity or a change and how reactions can be influenced. Instruments measuring attitudes can enable developers of emerging technologies gauge user and potential user perceptions and intentions to use the technologies and judgments about trusting them.

However, developing instruments for predicting reactions to various emerging technologies can be time consuming and cost prohibitive. Time is often of the essence when determining the prospects of a new technology as well as how much to invest to educate and promote it before too many resources are expended on research and development. This paper reports on the development of a simple and effective survey instrument to quantify potential user attitudes about one emerging technology.

2. LITERATURE REVIEW

Drake (2002) while quoting the Center for Devices and Radiological Health�much Division of Device User Programs and Systems Analysis wrote: the study of human factors is a science devoted to understanding the interaction of people (users) and equipment (p. 8). Attitudes and perceptions are significant human factors in the acceptance of new tools for information systems and communication.

Models exist for examining users' acceptance of technology and explaining the various dynamics and factors that contribute to a successful or otherwise adoption of innovations (Venkatesh, Morris, Davis, & Davis, 2003). The Technology Acceptance Model (TAM) specifies that the usage of information technology is determined by beliefs a user holds about the perceived usefulness (PU) and perceived ease-of-use of the technology (PEU) (Davis, 1989). According to Lanseng and Andreassen (2007), TAM posits that the actual use of information technology is determined by a user's intentions and attitudes more than beliefs. Jarrett (2003) said the Theory of Reasoned Action can be used to predict intent regarding adoption of innovations based on attitudes and subjective norms.

The research literature documents many instruments that have been used to measure attitudes toward technology (Bandalos & Benson, 1990; Francis & Katz, 1996; Loyd & Gressard, 1984; Masoud, 1990; Sexton & King, 1999). The Semantic Differential Scale (SDS) is a tool used frequently for measuring social attitudes (Osgood, Suci & Tannenbaum, 1957). Attitudes are evaluations--dispositions toward or away from things, people, or concepts; beliefs, on the other hand are thoughts people have about the object or construct (Intrieri, von Eye, & Kelly, 1995). Judgments must be focused on a single construct. The SDS is a seven-point bipolar rating scale that uses opposing adjective pairs from which respondents select a point corresponding to their disposition about the object or concept in question (Christensen &Knezek, 1998; Osgood, Suci & Tannenbaum, 1957). It has many advantages including its relative ease to construct, ease of use for research participants, and reliability of the quantitative data it provides. Researchers can create a scale using carefully selected opposing adjectives pairs for effectively quantifying attitudes on a wide range of constructs. Osgood et al. (1957) observed in their own studies testing this scale that the correlation scores across 100 college students surveyed on 40 items yielded a reliability coefficient of 0.85.

3. METHODS

Developing the Emerging Technologies Semantic Differential Scale

The SDS for this study was developed by providing a description of sensor devices and sensor network systems to four graduate interns in a university technology transfer organization. Each participant received a mocked-up description of the devices as well as the global deployment of sensor systems (see Appendix A) and was asked to list adjectives each would use to describe his or her impression of the object and concept. The lists were pooled then sent back to each participant for a second review. Participants were asked to identify more adjectives they preferred from their peers' suggestions or stick to their own choices. Twelve adjectives in total were agreed to by at least 75% of the participants. Two of the 12 adjectives were eliminated because of participants' perceptions that the adjectives conveyed similar meanings to other adjectives on the list. For instance, during a discussion convened after participants had made their lists, two of the participants indicated that it was difficult for them to distinguish the difference between *not complicated* and *not complex*. Likewise, the participants indicated that *novel* and *innovative* could be construed as having the same meaning. As a result, *not complex* and *novel* were dropped from the list, and the remaining ten adjectives were used to devise an Emerging Technologies Semantic Differential Scale (ETSDS).

The online Encarta World English dictionary was used to select the best antonyms for the ten adjectives; the bi-polar adjective pairs comprised the scale to measure attitudes. Adjective pairs were alternated so that positive adjectives and negative did not align on opposite sides of the scale. This step helped to prevent acquiescent responses on either side of the scale.

For example, participants were asked to rate "global deployment of sensor systems" in terms of an attribute. A feature such as value, for example, would be represented on the numeric semantic differential scale in the following form:

Unsafe <u>1 2 3 4 5 6 7</u> Safe

Participants indicated whether they judged the concept of "global deployment of sensor networks" to be extremely safe or not by marking the extremes (7 or 1 respectively) or if they have not formed a judgment, by selecting the neutral position 4, which is half-way between the two extremes.

The Emerging Technologies Semantic Differential Scale is a 10-item scale where the higher score is equal to a positive attitude.

Data Collection Procedures

The pilot testing of the ETSDS was undertaken with 85 doctoral students and doctoral graduates of an Information Systems program in a Mid-western university. Survey Monkey was used to administer the web-based survey that included demographic questions and the 10-item scale. Instructions were provided; anonymity was assured, and the study was IRB approved. Seventy-five completed and usable surveys were returned. The data were uploaded to SPSS for analysis.

4. DATA ANALYSIS

Scores for each of the items on the ETSDS were summed and divided by number of items (10) to determine each participant's score. Score interpretation included: 0.0-1.99 = very negative; 2.0-2.99 = negative; 3.0-3.99 = moderately negative; 4.0–4.99 = undecided/neutral; 5.0- 5.99 = moderately positive; 6.0– 6.99 = positive; 7 = very positive. Pearson product-moment correlation (2-tailed) was calculated to note the significance of relationships between items on the ETSDS. The assumption for using correlation technique is that mean scores are normally distributed; and all observations remain independent of each other. To determine the reliability of the Emerging Technologies Semantic Differential Scale., a factor analysis was performed.

5. RESULTS

Table 1. Factor Analysis of the ETSDS

Component Matrix

		Component	
	Bipolar Adjectives	1	2
B1	Unsafe – Safe	.645	.313
B2	Meaningful – Meaningless	.818	-.017
B3	Uninspiring – Motivating	.704	-.050
B4	Interesting – Tedious	.617	-.302
B5	Outdated – Innovative	.616	-.453
B6	Good – Bad	.806	.251
B7	Complicated – Simple	***.164***	***.847***
B8	Useful – Useless	.812	.012
B9	Unreliable – Reliable	.663	.286
B10	Time Saving - Time Consuming	.597	.082

Confirmatory factor analysis (CFA) demonstrated the existence of two factors in the ETSDS based on the pilot test; the 10 items created two scales; the bipolar adjective *complicated-simple* loads by itself on the second component while all the other items load on the first component, given that the criterion for factor item retention was a loading of at least .50 (Nunnally, 1978). Because it is not advisable to have a single item in a scale (Nunnally, 1978), a reliability check was performed to determine if the scale is unidimensional and reliable with the *complicated-simple* bipolar adjective included. The alpha for all items was 0.83, therefore a unidimensional scale works. Also the alpha increased slightly without the bipolar adjective *complicated-simple* to 0.87.

Except in associations where *complicated-simple* is one of the items, this result shows that as respondents scored higher on an item, higher scores are also observed in an associated item. The three strongest include *useless-useful* and *bad-good* (r=0.76); *useless-useful* and *meaningless-meaningful* (r=0.69) and; *bad-good* and *unsafe-safe* (r=0.67). Negative associations were observed between *complicated-simple* and each of the other items on the scale**.** (Table 2, Appendix B)

6. DISCUSSION AND CONCLUSION

This study makes several important contributions to the technology management field, and to the attitudes and technology acceptance literature. Typical studies focus on investigating attitudes using scales based on fear, anxiety, and other emotions, however, the Emerging Technologies Semantic Differential Scale was developed and demonstrated to be a simple measure that was also proven to be reliable to quantify attitudes as evaluative judgments about objects and concepts concerning information and communication technologies. Analysis centered on bipolar adjectives that have relevance to the constructs used in traditional TAM investigations (e.g. usefulness). This investigation indicates that by adopting this technique, credible results can be obtained swiftly for studies focused on technology users and stakeholders. This transcends traditional techniques and boundaries, and can be valuable for understanding attitudes. This study may inspire new research on a more global scale to investigate relationships between knowledge and attitudes to emerging technologies.

Future investigations could expand to potential users and users of emerging technologies, thereby providing further insights into evaluative judgments about little known technologies. The context of the study is relatively new in Information Systems, and thus the instruments used are not well established in this area. The use of Semantic Differential Scales seems to be an advantage for similar studies. For instance, a limitation that has been observed with semantic differential tools is the situation where responses appear to be linear on the extremes of the bipolar adjective scale, a situation that has been ascribed to the education level of respondents (Lenno, 2006). According to Sommer and Sommer (1997), people with lower levels of education often will abandon the middle points of the scale and focus on the fringes. This limitation was not observed in this study since respondents were highly educated. The terms used might have slightly different interpretations although this was minimized by using iterative process with people who have experience in assessing or evaluating new technologies. Social desirability is a limitation of this tool, especially where participants are highly invested in the study or concept being researched.

7. REFERENCES

Ajzen, I., & Fishbein, M. (1980). *Understanding attitudes and predicting social behavior*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Al-Hindawe, J. (n.d.). *Considerations when constructing a semantic differential scale.* Retrieved September 17, 2010 from: http://www.scribd.com/doc/50065193/Semantic-differential-scale

Bandalos, D., & Benson, J. (1990).Testing the factor structure invariance of a computer. *Educational and Psychological Measurements, 50*(1), 49.

Christensen, R. & Knezek, G. (1998).Parallel forms for measuring teachers' attitudes toward computers. *Society of Information Technology & Teacher Education (SITE)'s 9th International Conference*, Washington, DC, March 13, 1998.

Davis, F. D. (1986). *A technology acceptance model for empirically testing new end-user information systems: Theory and results*. Doctoral Dissertation, MIT Sloan School of Management, Cambridge, MA.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, *13*(3), 319-340.

Davis, F. D. & Venkatesh, V. (2004).Toward pre-prototype user acceptance testing of new information systems for software project management. *Engineering Management, IEEE Transactions, 51*(1), 31-46.

Drake, C. (2002). Human factors: A vital component in product development and launch. *International Journal of Medical Marketing 1469–7025 3* (1), 7–19.

Fishbein, M. A. (1975). *Belief, attitude, intention, and behavior: An introduction to theory research.* London: Addison-Wesley Publishing Company.

Francis, L.J., & Katz, Y.J. (1996).The gender stereotyping of computer use among female undergraduate students in Israel and the relationship with computer-related attitudes. *Journal of Educational Media, 22*(2), 79, 78.

Imran, A. (2009). Knowledge and attitude, the two major barriers to ICT adoption in LDC are the opposite side of a coin: Empirical evidence from Bangladesh. *42nd Hawaii International Conference on Systems Science* (HICSS-42 2009), pp. 1-10.

Intrieri, R. C., von Eye, A., & Kelly, J. A. (1995). The aging semantic differential: A confirmatory factor analysis. *The Gerontologist 35*(5), 616. Retrieved from http://search.proquest.com/docview/210925812?accountid=28365

Jain, A., (2006). *When preconceptions matters: Understanding pre-prototype usefulness of information technology, the case of a municipal wireless network.* Doctoral Dissertation, Temple University, 2006, AAT 3247270.

Jarrett, S. M. (2003). Factors affecting the adoption of e-business in the aerospace industry. [D.B.A. dissertation].Nova Southeastern University. AAT 3096349.

Lanseng, E. J., & Andreassen, T. W. (2007). Electronic healthcare: A study of people's readiness and attitude toward performing self-diagnosis. *International Journal of Service Industry Management*. *18* (4) 394-417.

Lau, S., Chang, T., Hu, S., Huang, H., Shyu, L., Chiu, C. & Huang, P. (2006). Sensor network for everyday use: The BL-Live experience. *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing 1,*336-343.

Loyd, B.H., & Gressard, C. (1984). Reliability and factoral validity of computer attitude scale. *Educational and Psychological Measurement, 44*(2), 501-505

Masoud, S.L. (1990). Factorial validity of a computer attitude scale. *Journal of Research on Computing in Education, 22*(1), 290-210.

Nunnally, J. (1978). *Psychometric theory*. New York: McGraw Hill.

Osgood, C.E., Suci, G.J. &Tannenbaum, P.H. (1957). *The measurement of meaning.* Urbana: University of Illinois Press.

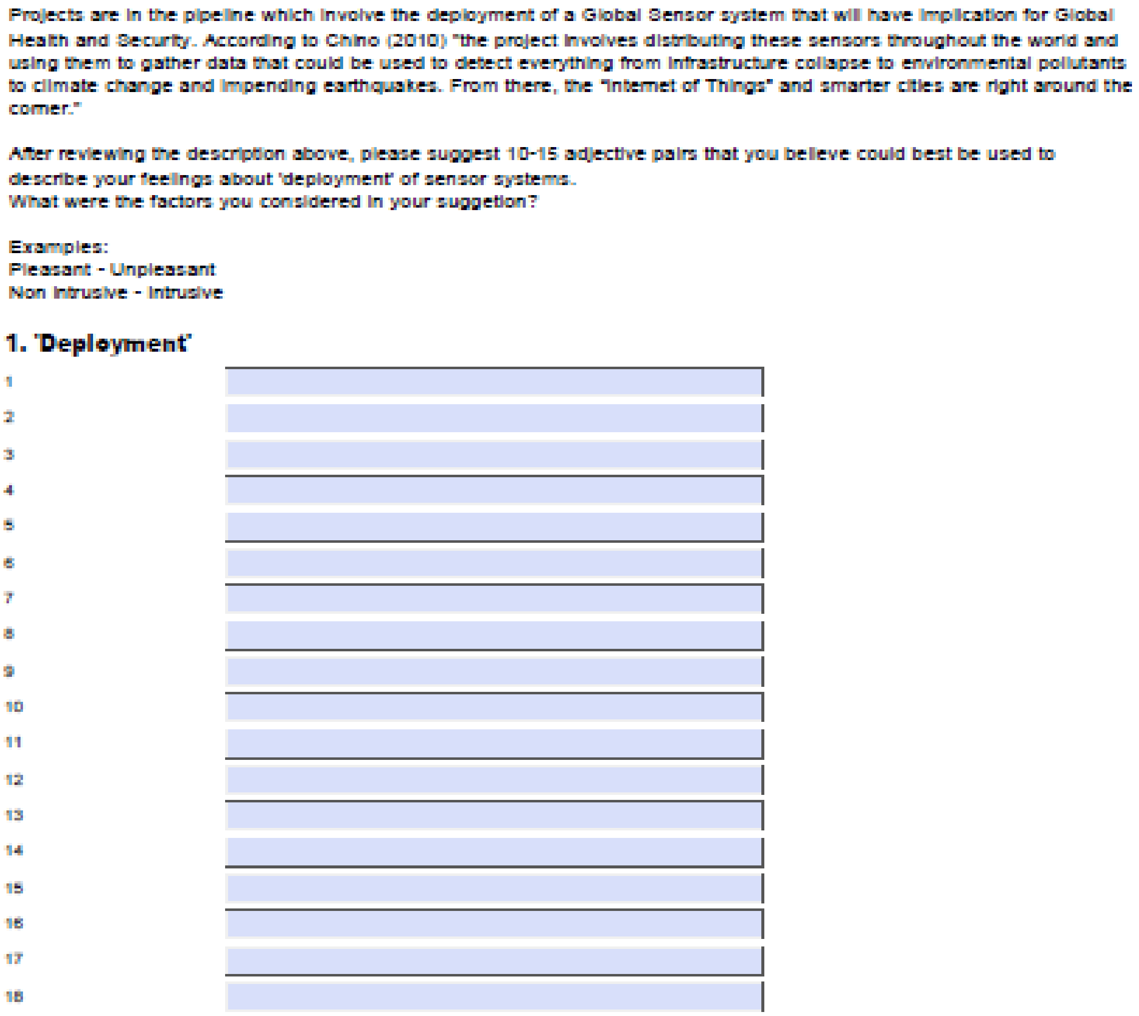
Sexton, D. & King, N. (1999). Measuring and evaluating early childhood prospective practitioners' attitudes toward computers. *Family Relations, 48*(3), 277

Sommer, B. & Sommer, R. (1997). *A practical guide to behavioral research: Tools and techniques.* New York: Oxford University Press.

Venkatesh, V. & Davis, F. D. (2004).Toward pre-prototype user acceptance testing of new information systems: Implications for software project management. *IEEE Transactions on Engineering Management*, *51*(1), 31-46.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly, 27* (3), 425-478.

Appendix A: Description of Sensor network devices



Projects are in the pipeline which involve the deployment of a Global Sensor system that will have implication for Global Health and Security. According to Chino (2010) "the project involves distributing these sensors throughout the world and using them to gather data that could be used to detect everything from infrastructure collapse to environmental pollutants to climate change and impending earthquakes. From there, the "Internet of Things" and smarter cities are right around the corner."

After reviewing the description above, please suggest 10-15 adjective pairs that you believe could best be used to describe your feelings about 'deployment' of sensor systems.
What were the factors you considered in your suggetion?

Examples:
Pleasant - Unpleasant
Non Intrusive - Intrusive

1. 'Deployment'

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

Appendix B: Table 2

Table 2 Inter-item Correlations (N=75)

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
B1	1									
B2	.487***	1								
B3	.495***	.588***	1							
B4	.347***	.627***	.482***	1						
B5	.337***	.628***	.489***	.533***	1					
B6	.673***	.605***	.605***	.515***	.535***	1				
B7	-.104	-.276***	-.133	-.333***	-.438***	-.258***	1			
B8	.475***	.687***	.567***	.514***	.555***	.759***	-.308***	1		
B9	.443***	.412***	.340***	.296***	.260***	.447***	.080	.493***	1	
B10	.367***	.443***	.430***	.452***	.430***	.605***	-.035	.575***	.378***	1

***Significant at the 0.001 level (2-tailed), $p \leq 0.001$.
B1=Unsafe-safe; B2=Meaningless-Meaningful; B3=Uninspiring-Motivating; B4=Tedious-Interesting; B5=Outdated-Innovative; B6=Bad-Good; B7=Complicated-Simple; B8=Useless-Useful; B9=Unreliable-Reliable; B10=Time consuming-Time saving.

Predicting the Terminal Ballistics of Kinetic Energy Projectiles Using Artificial Neural Networks

John R. Auten Sr.
jauten1@students.towson.edu

Robert J. Hammell II
rhammell@towson.edu

Department of Computer and Information Sciences
Towson University
Towson, MD 21252, USA

Abstract

The U.S. Army requires the evaluation of new weapon and vehicle systems through the use of experimental testing and vulnerability/lethality modeling & simulation. The current modeling and simulation methods being utilized often require significant amounts of time and subject matter expertise. This means that quick results cannot be provided to address new threats encountered in theatre. Recently, there has been an increased focus on rapid results for modeling and simulation efforts that can also provide accurate results. Accurately modeling the penetration and residual properties of a ballistic threat as it progresses through a target is an extremely important part of determining the effectiveness of the threat against that target. This paper proposes the application of artificial neural networks to the prediction of the terminal ballistics of kinetic energy projectiles. By shifting the computational complexity of the problem to the fitting (regression) phase of the algorithm, the speed of the algorithm during an analysis is improved when compared to other terminal ballistic models for kinetic energy projectiles. An improvement in overall analysis time can also be realized by removing the need for input preparation by a subject matter expert prior to using the algorithm for an analysis.

Keywords: Kinetic Energy Projectiles, Terminal Ballistics, Artificial Neural Networks, Data Mining.

1. INTRODUCTION

When a U.S. Soldier takes a weapon system into the field for the first time, that Soldier wants to know that the weapon system will perform as expected. In order to ensure that the Department of Defense (DoD) acquires systems that are safe and effective; they test the system and use modeling and simulation to augment the results from the tests. The DoD requires that Acquisition Category (ACAT) I systems undergo Live-Fire Test & Evaluation (LFT&E) (U.S. Department of Defense, 2008) to determine the Vulnerability/Lethality (V/L) of that system. Simulation models are validated to those live-fire tests and then accredited so that they can be used for future studies involving that system.

The focus of this research is on the development of an ANN that can predict the terminal ballistics of Kinetic Energy Projectiles (KEPs). This paper provides an overview of the proposed research and the current progress, specifically examining the issue of missing data. The paper is organized as follows: an introduction to V/L modeling is given in this section, followed by an overview of terminal ballistics in section 2. Section 3 describes the proposed modeling method and section 4 outlines the approach of this research. Section 5 presents current progress, followed by a discussion of conclusions and future research in the final section.

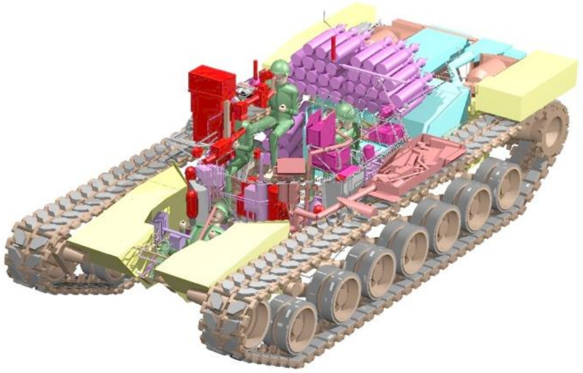


Figure 1. A CAD target model (Deitz & Ozolins, 1989)

Vulnerability/Lethality Modeling

V/L simulation models are used to analyze the vulnerability of military systems against the lethality of weapons systems. V/L models typically consist of a Computer Aided Design (CAD) model (figure 1) of the target system, engineering definitions for the systems and sub-systems in the target, engineering inputs for the probability of component dysfunction given a hit ($P_{cd|h}$) for the target critical components, methodologies for determining system capabilities after a ballistic event, and algorithms for modeling the physical interaction of the target and the ballistic threat. This work focuses on the ballistics of the physical interaction of the threat and the target.

In V/L simulations the interaction of the target and threat is modeled as a shotline going through the target. A ballistic interaction can consist of one or many shotlines depending on the threat of interest. For example, a Shaped Charged Jet (SCJ) threat that impacts armor could generate Behind Armor Debris (BAD) which may consist of thousands of fragments, each one requiring its own shotline. Furthermore, a fragment threat could fracture upon impact and separate into several shotlines of smaller fragments.

A single interaction could require many shotlines to fully analyze the ballistic event. A typical example analysis of a target and threat could run twenty-six views or more, with each view requiring hundreds of thousands of shotlines (Moulsdale, 2012). Once all of the shotlines are tallied for a full analysis the total count can be in the hundreds of millions.

For each shotline, remaining system capability is determined based on which components are damaged. Before damage can be calculated, the model must determine if the components were hit. Determination of a hit on a component is performed by calculating how far the threat can penetrate into the target on the shotline.

An example of a shotline going through a vehicle can be seen in figure 2. The components that intersect with the shotline are considered “threatened” and are highlighted in the figure. How far along the shotline the threat can penetrate will determine which “threatened” components are actually hit. Terminal ballistics models, also known as penetration models, are used to determine how far a projectile travels on a shotline. Once the distance traveled is known, the critical components that were hit by the projectile are also known. Due to the large number of shotlines and the need for accuracy, the calculation speed and correctness of a penetration model are important.

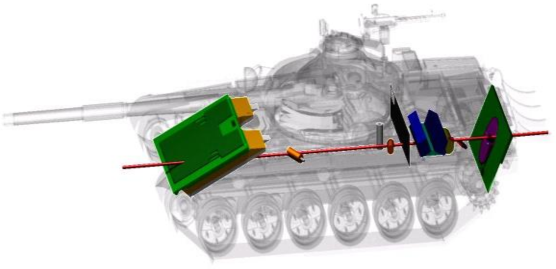


Figure 2. A shotline through a target vehicle (Dibelka, 2004)

On a particular shotline there can be many objects in the path of the projectile, so if the projectile perforates after impacting the first object on the shotline it may impact another object. For each object, a terminal ballistics model is applied to determine if the projectile will perforate the object or be defeated (Deitz, Reed Jr., Klopcic, & Walbert, 2009). The first impact event will use the initial inputs for the terminal ballistics model, and results from that impact are used as inputs to run the terminal ballistics model for subsequent impact events.

Results from the terminal ballistics model are used to determine the damage on a critical component in the target. Typically the damage to a critical component is determined using empirical models based on mass and velocity, hole size (function of projectile diameter), or energy deposited (function of mass and velocity). For each of those cases, the residual parameters of the projectile after impact are needed for determination of damage (Deitz et al., 2009). While it is important to accurately predict component perforation, it is also important to be accurate in predicting the projectile's residual parameters since they determine the damage inflicted to the target and residual penetration capability.

2. TERMINAL BALLISTIC MODELS

This research concentrates on the terminal ballistics of a particular threat type known as KEPs. KEPs are typically launched from a gun system using a sabot and can be stabilized in flight via spinning or the use of fins. They are typically made from hard and high density metals like steel, tungsten, or depleted uranium. An example KEP called an Armor Piercing Fin Stabilized Discarding Sabot-Tracer (APFSDS-T) round is shown in figure 3.

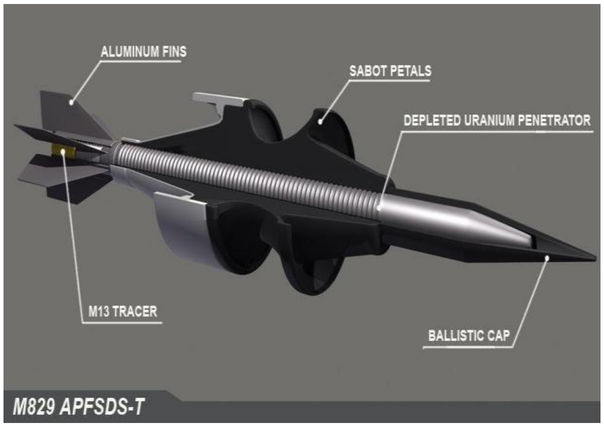


Figure 3. An APFSDS-T round ("M829", 2012)

There are several models that are currently used to predict KEP penetration. Two have been chosen for discussion: the Lanz-Odermatt model, due to its simplicity and wide spread usage and the Segletes Hybrid model, due to its broad modeling capability and correctness (Auten, 2012).

Lanz-Odermatt

The Lanz-Odermatt model (Lanz & Odermatt, 2000) is an empirical model that is fit to test data by a Subject Matter Expert (SME). The model is very fast to run since it consists of only a few equations, but it is not a generalized model. Therefore it requires different coefficients of fit for different threats and target interactions.

Segletes Hybrid Model

The Segletes hybrid model (Segletes, 2000) is a phenomenological model built on the Bernoulli equation. It is an accurate model, but requires more run time since it uses numerical integration to solve the partial differential equations associated with it.

3. PROPOSED MODELING METHOD

Test data are available with respect to KEP penetration into various materials. However, using such data with either of the current models described above will require significant computational time or will not provide a generalized model, or both. This work investigates the use of Artificial Neural Networks (ANNs) to overcome these limitations.

Tarassenko (1998) lists five key attributes of neural networks in the book "A Guide to Neural Computing Applications":

Learning from Experience

> Neural networks are particularly suited to problems whose solution is complex and difficult to specify, but which provide an abundance of data from which a response can be learned.

Generalizing from Examples

> A vital attribute of any practical self-learning system is the ability to interpolate from a previous learning 'experience'. With careful design, a neural network can be trained to give the correct response to data that it has not previously encountered.

Developing Solutions Faster with less Reliance on Subject Matter Expertise

Neural networks learn by example and, assuming sufficient examples and an appropriate design, effective solutions can be constructed far more quickly than with traditional approaches, which are entirely reliant on experience in a particular field. However, neural networks are not wholly independent of domain expertise which can be invaluable in choosing the optimal neural network design.

Computational Efficiency

Training a neural network is computationally intensive, but the computational requirements of using a fully trained neural network can be modest. For very large problems, speed can be gained through parallel processing as neural networks are intrinsically parallel structures.

Non-Linearity

Many other processing techniques are based on the theory of linear systems. In contrast, neural networks can be trained to generate non-linear mappings, giving them an advantage for dealing with complex, real-world problems.

ANNs are a common tool for performing non-linear regression, especially when the parametric form of the function is unknown and when the number of parameters is large (Gruss & Hirsch, 2001). A specific type of ANN called a Multilayer Perceptron (MLP) has been shown to be a universal approximator, meaning it is capable of arbitrarily accurate approximation to an arbitrary mapping, if there are enough hidden neurons in the hidden layer (Gonzalez-Carrasco, Garcia-Crespo, Ruiz-Mezcua, & Lopez-Cuadrado, 2011). With the appropriate parameters, a MLP should be able to accurately approximate the desired outputs. The parameters include the inputs to the model, the topology of the MLP (to include the activation functions, number of layers, and number of neurons), the error function, the training method, and the test data.

The application of a MLP for this research was chosen based on the work of I. Gonzalez-Carrasco, et al. (2011), which found the application of MLPs to outperform Radial Basis Function (RBF) networks, Support Vector Machines (SVMs), and Recurrent Neural Networks (RNNs) for predicting perforation of steel, Depleted Uranium (DU), or Tungsten Heavy Alloy (WHA) KEPs against aluminum, steel or DU targets.

4. APPROACH

This section presents the proposed research approach. The general steps for ANN design (data preparation, determination of inputs, choice of learning method, choice of global optimization method, and use of generalization techniques) will be discussed in turn. Then, the specific ANN architecture and initial prototype used in this work will be outlined.

Data Preparation

The preparation of the data for use is an extremely important step in developing an ANN model, and is often the most time consuming. As Tarassenko (1998) states:

> *Artificial Neural Network projects are data driven, therefore there is a need to collect and analyze data as part of the design process and to train the neural network. This task is often time-consuming and the effort, resources, and time required are frequently underestimated.*

Experimental test data is inherently noisy, but hidden assumptions in the data collection methods or data processing methods could cause major differences in the data. As an example, suppose there are four reports containing experimental test data, and during the test events for all of the reports the KEP fractured into smaller pieces as it perforated the target. In report number 1, the residual mass is reported as the weight of the largest piece. In report number 2, the residual mass is reported as the weight of all of the pieces. In report 3, x-ray is used to approximate the length and diameter of the largest few pieces, and then the mass is calculated using the volume and density of the rod material. In report 4, a piece of the KEP that was embedded in the target is included in the residual mass calculation.

The above scenario produces four similar test events with four different reported results. The example given shows how important it will be to find outliers in the training data and attempt to track down the cause of the discrepancies so that they can be fixed or omitted.

In order to decrease the likelihood of poor predictions when extrapolating it is important to use training data that covers the range of all possible inputs. Figure 4 shows an example of what can happen if a region of the input space is omitted from the training data. The square marks are the data points that were used for the non-linear regression, the circular marks are the data points that were omitted, and the curved line shows the model predictions. The model predicts the training data very well and accurately interpolates between the data points but because of the omitted data the wrong model was used for fitting, thus leading to poor extrapolation.

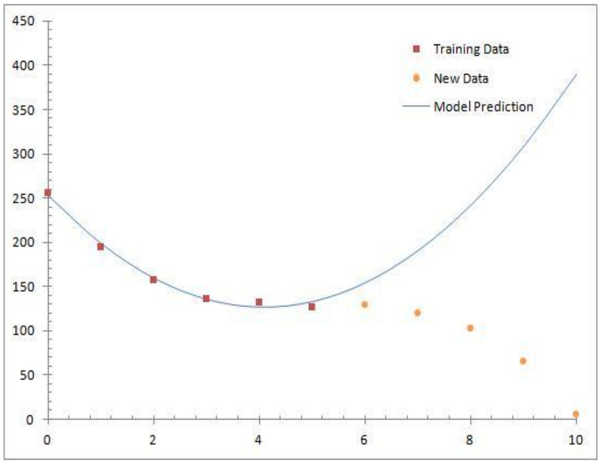


Figure 4. Example of poor extrapolation

The collection of experimental test data that is representative of the large space of possible input patterns and that can be used for training, testing, and validating the MLP, will be one of the more difficult tasks involved in this research (Fernández-Fdz & Zaera, 2008). Therefore, a large part of the effort for this research will be finding and documenting publicly available experimental test data for KEPs.

Determination of Inputs

Once the training data have been collected, the next step in the process of defining a MLP is the determination of inputs for the model (Walczak & Cerpa, 1999). The determination of what inputs to use is done early in the process because it drastically affects other parts of the MLP design. The number of inputs in a MLP is limited by the number of available input parameters in the problem, but it is possible that not all available input parameters should be utilized (Gonzalez-Carrasco, Garcia-Crespo, Ruiz-Mezcua, & Lopez-Cuadrado, 2008).

There is often a desire to include too many inputs in the MLP design due to two common misconceptions; (1) since they learn, they will be able to determine what input variables are important, and (2) like with expert systems, as much domain knowledge as possible should be included into the system (Walczak & Cerpa, 1999).

Determination of the input parameters is extremely important for two primary reasons. The first reason is that the required number of data points increases with the number of input parameters. The second reason is that including two inputs that are highly correlated introduces noise in the training data which can lead to a loss of generalization and could cause a non-convergence of the MLP (Kapoor, Pal, & Chakravartty, 2005).

Learning Methods

The next step in defining the MLP involves picking an appropriate learning method for the problem class being addressed (Walczak & Cerpa, 1999). The choice of learning method will determine how well the MLP will learn the patterns that it is being taught and includes the learning algorithm, error function, learning rate, and other optional methodologies. The optimization algorithms used for learning fall into two categories: direct (gradient-free) methods or gradient methods.

Direct methods use only the function values themselves to find the optima in question. Examples of direct methods include simulated annealing, perturbation methods, or genetic algorithms. The advantages of direct methods are that there is no need to derive or compute gradients and that the methods can find a global optimum. The disadvantages are that they can take too many iterations to converge to a solution and although they can come to a solution close to a global optimum, there is no guarantee that they will come to that exact solution.

Gradient methods use the gradient of the function to determine the optima in question and can be further defined as 1st or 2nd order. Examples of gradient methods include gradient descent, Newton method, Gauss-Newton method, and Levenberg-Marquardt method. The primary difference between a 1st order and 2nd order method is the required number of iterations prior to convergence and speed of calculation. 1st order methods only need to calculate the 1st derivative of the function which requires less calculation time, but may take a less directed approach to finding the optimum. 2nd order methods require longer to calculate 2nd derivatives or the Hessian matrix, but take a more direct approach to finding the optimum (Snyman, 2005).

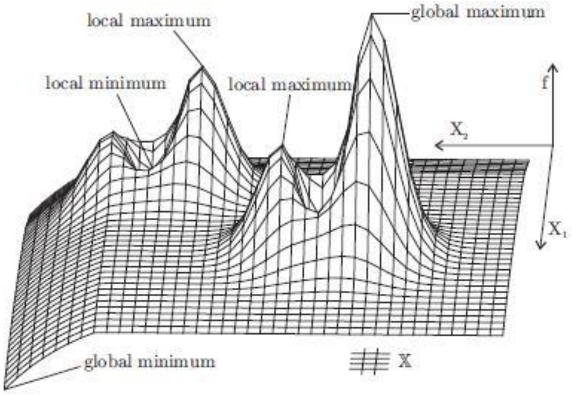


Figure 5. 3-Dimensional example of local and global optima (Weise, 2009)

Any of the example optimization methods can be used to find a minimum of an error function, however a global minimum for the error function is not guaranteed. A hybrid of the two methods will be used for this research and will be discussed in the next section.

Methods for Global Optimization

A function can have multiple optima; figure 5 shows an example function that contains four maximums and three minimums, but there is only one global (overall) maximum and only one global minimum. An optimization function that does not guarantee the convergence to a global optimum could converge to a non-optimal solution if other methods are not used.

There are several techniques available to increase the likelihood of finding the global minimum for the error function. One technique that can be used is the method of momentum; momentum is used to resist changes to the direction of the weight changes. The main reason for using momentum is to reduce the chance of oscillating around a minimum; however, there is a slight chance that since momentum can also speed up the weight adjustments that it may skip over a small local minimum (McInerney & Dhawan, 1993). Momentum was not originally designed for finding global minimums and its probability of skipping a local minimum is small, so other techniques are better suited for this purpose.

Another technique that can be used is to sample several random potential weights for the network and start with the one that has the lowest error. The random sampling technique in no way guarantees a global minimum, but does help the learning process by allowing the network to start the learning process as close to a minimum solution as possible and could start the learning process close to a global minimum (Kapoor et al., 2005). A disadvantage of this method is that since it is truly random it is not a directed approach and is therefore inefficient when compared to directed methods.

A technique that has gained popularity is to use a hybrid approach that attempts to utilize the benefits of direct and gradient optimization methods together. Since direct methods are traditionally better equipped to find global optimum, a direct method is used first to get close to a global optimum. Direct methods however are typically inefficient in converging to the optimum solution, so the next step is to apply a gradient method to assist in the convergence.

An example of this technique is the use of Genetic Algorithms (GAs); they can be used to determine starting weights for the network prior to the learning process beginning. Like with random sampling, using a GA does not guarantee a global minimum, but does increase the likelihood of finding it since it is a directed method and is more efficient than random sampling (McInerney & Dhawan, 1993). Once a criterion has been met by the GA the learning process begins using a gradient method for the determination of the required weights to reach the global minimum of the error function. This hybrid method is what will be used for this research.

Generalization Techniques

As mentioned earlier, it is important to this research project that any model developed be a generalized solution. If non-representative data is used to train the MLP then poor extrapolation could occur. But even if representative data is used for training, if the MLP is not properly designed then it could over-predict and not provide a smooth fitting of the training data. As an example, Figure 6 shows a model that has been overfit to the training data. The diagonal line represents a good fit to the training data points, but the curved line represents a solution that could come from a MLP if overfitting occurs.

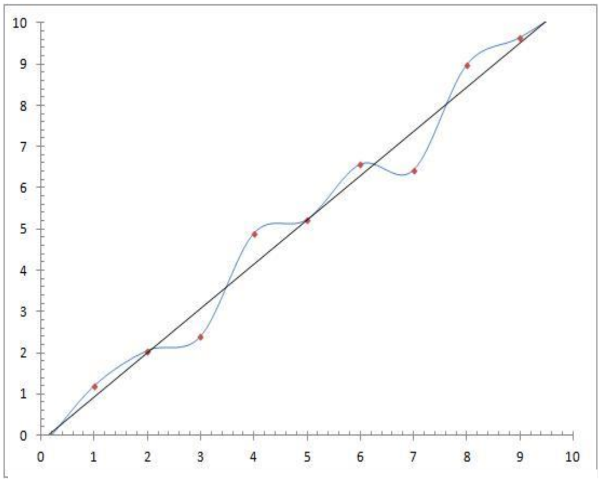


Figure 6. An example of overfitting to data

There are techniques available to increase the likelihood of producing a generalized solution and reduce the risk of overfitting. One such methodology is weight decay; it penalizes large weights in the network and causes the weights in the network to converge to smaller absolute values. Excessively large weights in the network can lead to excessive variance of the outputs from the network (Sarle, 2002). Another method for producing a more generalized model is to use early stopping. During the training phase a training set of data is used for learning as usual, but the error of a validation set of data is also tested. If the error of the validation data set begins increasing then the training is stopped early (Gonzalez-Carrasco, Garcia-Crespo, Ruiz-Mezcua, & Lopez-Cuadrado, 2011).

ANN Architecture

The work of Fernández-Fdz, Puente, and Polo (2008) used an application of ANNs that broke the prediction of residual values into a two step process. Instead of using one MLP for determining perforation and residual values, the task was broken up into a MLP for classification (perforation and non-perforation) and if perforation was predicted, a second MLP for regression of the residual values. The benefit of separating the two tasks is the reduction in complexity of the overall networks and therefore an increase in the likelihood of faster convergence.

The design for this research will follow a similar approach. The modeling of the terminal ballistics of KEPs will be broken into two sub-problems, one of classification (perforation or non-perforation) and one of regression (determination of residual parameters).

The effect that MLP complexity has on the amount of training data required can be demonstrated by using equations 1 and 2. They provide an approximation of the number of training data points required for a given network topology, or reciprocally the size limitation of a network topology due to the number of training data points (Tarassenko, 1998). In equation 1, n is the number of training data and W is the total number of network parameters (the network parameters are the weights associated with the connections between the nodes in the ANN) that must be adjusted during training.

$$W \leq n \leq 10W \quad (1)$$

$$\mathrm{W} = \sum_{\mathrm{i}=1}^{\mathrm{N}-1}(\mathrm{L_i} + 1)\mathrm{L_{i+1}} \quad (2)$$

For example, for a simple 2-layer MLP with two input neurons, two hidden neurons, and one output neuron, the recommended number of training data fall between nine and ninety. For a 3-layer MLP with six input neurons, seven hidden neurons in the first hidden layer, six hidden neurons in the second hidden layer, and three output neurons, the recommended number of training data fall between one hundred eighteen (118) and one thousand one hundred eighty (1180). The more complex the MLP the more data are required for training.

Initial Prototype

Due to the simplified complexity of the problem, the first prototype will concentrate on the problem of determining perforation of a single plate of homogeneous armor. After that ANN has been developed, the next step will be to develop an ANN to determine the residual parameters for the KEP. The ANN will be applied by determining perforation for each plate along the shotline and utilizing the residual parameters for any subsequent plate along the shotline.

As more test data become available, or as gaps in data are exposed and filled using Finite Element Methods (FEM), the ANNs can be refit and refined to better model the kinematics of terminal ballistics.

5. CURRENT PROGRESS

Experimental Test Data

The experimental test data have been categorized into three main types; semi-infinite, finite, and limit. Semi-infinite test data comes from a penetration test into a material that is of such thickness that the area of deformation in front of the projectile is not expected to reach the rear face of the target. Finite test data comes from a test where the target material is of a finite thickness and under certain circumstances the projectile could perforate the target. Limit test data comes from many finite test series to determine at what velocity perforation would occur 50% of the time; this is known as the ballistic limit or v_{50}. This research is currently focused on finite test data.

There are many physical properties that are typically recorded during experimental tests, but some of the more typical ones are:
Impact parameters such as velocity, yaw, pitch, and total yaw.
Projectile properties such as length, diameter, density, mass, and hardness.
Target properties such as thickness, obliquity, density, and hardness.
Residual values such as velocity, projectile mass, and projectile length.

XML Database

The database being used for this research was designed using an XML schema. Once the schema was developed, a Java library called JAXB was used to create an object model to store the database and provide read and write access to the XML file from within a Java Swing GUI. That tool is used primarily for data entry and querying of the XML database. A Southwest Research Institute report (SwRI) (Anderson Jr., Morris, & Littlefield, 1992) was used to populate the database with its initial data set. The report was digitally scanned and then processed using Optical Character Recognition (OCR). The data from the report was cleaned up and formatted into something that was readable by a Java program. The Java program then pulled the data into the database and wrote it out in the XML format.

The seven other reports that are currently entered into the database were entered in by hand. There are currently 25 more reports of data awaiting entry into the database.

There are 1,463 records in the database that contain semi-infinite test data, 644 records that contain finite test data, and 416 records that contain limit test data.

Data Concerns

Typical problems with using large amounts of data include incorrect recording, incorrect data entry, duplication, and missing parameters.

Of the 644 records in the database, only 75 contain all 15 of currently selected variables, 451 are missing one variable, 96 are missing two variables, and 22 are missing three variables. All of the 569 that are missing values have at least one missing value that is a dependent variable.

Statistical methods have been used to expose outlier data and subject that data to scrutiny. However, further efforts are required to ensure that the data is as clean as possible. There are statistical, clustering, pattern-based, and association rules methods for outlier detection available to help with the process of cleaning the data (Maletic & Marcus, 2005).

In order to develop the initial prototype MLP for classifying the data as perforation or non-perforation, every record that is to be used for training must contain all required parameters. There is no one solution to the problem of missing data, but through a combination of intelligent replacement, imputation, or maximum likelihood methods, suitable values can be placed into the missing data locations with minimal detrimental effect to the ability of the MLP to learn the patterns in the data. The listwise and pairwise deletion methods will be avoided if possible, due to the limited availability of test data.

One method of intelligent replacement is accomplished by making the common assumption that the diameter of the KEP does not change during penetration and by using basic geometric equations. Equation 3 can be used to solve for mass (m), density (ρ), radius (r), and length (l) as long as only one of the parameters are missing.

$$\frac{m}{\rho} = \pi r^2 l \qquad (3)$$

For the remaining missing data that cannot be addressed using intelligent replacement, a determination will need to be made whether the data is missing completely at random (MCAR), missing at random (MAR), or not missing at random (NMAR). The classification of how the data is missing has important ramifications on what methods are available to fill in the data voids (Enders, 2001).

6. CONCLUSIONS

The need for an accurate and generalized terminal ballistic model for KEPs is important due to their usage in V/L models that the U.S. Army uses to evaluate the survivability of military systems. To overcome the problems inherent in current modeling and simulation methods (slow speed, need for significant subject matter expertise), this paper proposes to use artificial neural networks to produce an accurate, general model for the prediction of the terminal ballistics of kinetic energy projectiles.

The use of ANNs for regression is a well documented process in many fields. This research proposes to use the approach in a specific area in which it has not been used before. However, this work contributes in the broader context as well by examining the issue of missing data. This is a problem in almost all data-based research, and dealing with it in an unbiased way is difficult but crucial. This research will examine multiple ways of solving the issue in a practical scenario.

A literature search has been completed for publications containing KEP experimental test data and that data has been partially entered into the database. An analysis has been performed to check the correctness of the entered data and search for outliers, but data voids still pose a problem. Further analysis will be performed to address the missing data and prepare the database for usage in the ANN.

Future Research

With the database prepared, the prototype ANN will be designed, implemented, and tested. The prototype ANN will immediately have usage as a predictor of the ballistic limit for armor and will serve as the classification step in the two part process that is proposed for this terminal ballistics model.

The next phase of this research will include further development and refinement of the database and the design, development, and testing of the regression ANN that will serve as the second part of the terminal ballistics model being proposed.

7. REFERENCES

Anderson Jr., C. E., Morris, B. L., & Littlefield, D. L. (1992). *A Penetration Mechanics Database* (Technical Report No. SwRI Report 3593/001). Southwest Research Institute.

Auten Sr., John R. (2012). *A Comparison of Kinetic Energy Rod Algorithm Predictions to Test Data* (Technical Report No. ARL-TR-6192). U.S. Army Research Laboratory.

Deitz, P. H., & Ozolins, A. (1989, May). *Computer Simulations of the Abrams Live-Fire Field Testing* (Memorandum Report No. BRL-MR-3755). Ballistic Research Laboratory.

Deitz, P. H., Reed Jr., H. L., Klopcic, J. T., & Walbert, J. N. (2009). *Fundamentals Of Ground Combat System Ballistic Vulnerability/Lethality* (E. Edwards, W. Hacker, W. Kincheloe, & D. Bely, Eds.). AIAA (American Institute of Aeronautics & Astronautics).

Dibelka, R. E. (2004, June). *MUVES-S2 Configuration Management*. Presentation at the 9th Annual Joint Aircraft Survivability Program Office (JASPO) Model Users Meeting (JMUM), June 22–25.

Enders, C. K. (2001). A primer on maximum likelihood algorithms available for use with missing data. *Structural Equation Modeling, 8*(1), 128-141.

Fernández-Fdz, D., Puente, J. L., & Polo, R. Z. (2008). Prediction of the behaviour of CFRPs against high-velocity impact of solids employing an artificial neural network methodology. *Composites Part A: Applied Science and Manufacturing, 39* (6), 989–996.

Fernández-Fdz, D., & Zaera, R. (2008). A new tool based on artificial neural networks for the design of lightweight ceramic-metal armour against high-velocity impact of solids. *International Journal of Solids and Structures, 45* (25-26), 6369–6383.

Gonzalez-Carrasco, I., Garcia-Crespo, A., Ruiz-Mezcua, B., & Lopez-Cuadrado, J. L. (2008). Neural Network Application for High Speed Impacts Classification. In *Proceedings of the World Congress on Engineering* (Vol. 1).

Gonzalez-Carrasco, I., Garcia-Crespo, A., Ruiz-Mezcua, B., & Lopez-Cuadrado, J. L. (2011). Dealing with limited data in ballistic impact scenarios: an empirical comparison of different neural network approaches. *Applied Intelligence, 35* (1), 89–109.

Gruss, E., & Hirsch, E. (2001, May). Approximating the Ballistic Penetration Function of a Jet in a Multi-Cassette Target by the use of Neural Networks. In *Proceedings 19th International Symposium of Ballistics 7–11 May 2001* (p. 1069-1075).

Kapoor, R., Pal, D., & Chakravartty, J. (2005). Use of artificial neural networks to predict the deformation behavior of Zr-2.5 Nb-0.5 Cu. *Journal of Materials Processing Technology, 169* (2), 199–205.

Lanz, W., & Odermatt, W. (2000, April 11-14, 2000). Minimum Impact Energy For KE-Penetrators in RHA-Targets. In M. Giraud & V. Fleck (Eds.), *Proceedings of the European Forum on Ballistics of Projectiles* (p. 349-365). ISL, French-German Research Institute of Saint-Louis.

M829. (2012). In *SBWiki*. Retrieved from http://www.steelbeasts.com/sbwiki/index.php/M829.

Maletic, J. I., & Marcus, A. (2005). Data Cleansing - A Prelude to Knowledge Discovery. In O. Maimon & L. Rokach (Eds.), *The Data Mining and Knowledge Discovery Handbook* (p. 21-36). Springer.

McInerney, M., & Dhawan, A. P. (1993). Use of genetic algorithms with backpropagation in training of feedforward neural networks. In IEEE *International Conference on Neural Networks* (Vol. 1, p. 203-208). IEEE.

Moulsdale, S. (2012, June). *MUVES-S2 Study: Comparison of Vulnerability Results When Varying Grid Cells Size and Number of Views* (Technical Report No. ARL-TR-5835). U.S. Army Research Laboratory.

Sarle, W. S. (2002). *Neural Network FAQ*. Retrieved from ftp://ftp.sas.com/pub/neural/FAQ.html.

Segletes, S. B. (2000, September). *An Adaptation of Walker-Anderson Model Elements Into the Frank-Zook Penetration Model for Use in MUVES* (Technical Report No. ARL-TR-2336). U.S. Army Research Laboratory.

Snyman, J. A. (2005). *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms* (Vol. 97; P. M. Pardalos & D. W. Hearn, Eds.). Springer.

Tarassenko, L. (1998). *A Guide to Neural Computing Applications*. New York: Arnold.

U.S. Department of Defense. (2008, December). *Operation of the Defense Acquisition System* (Department of Defense Instruction No. DoDI 5000.02). Washington, DC: Government Printing Office: U.S. Department of Defense. Retrieved from www.dtic.mil/whs/directives/corres/pdf/500002p.pdf

Walczak, S., & Cerpa, N. (1999). Heuristic principles for the design of artificial neural networks. *Information and software technology*, 41 (2), 107–117.

Weise, T. (2009). *Global Optimization Algorithms – Theory and Application*. it-weise.de (self-published): Germany. Retrieved from http://www.it-weise.de/projects/book.pdf.

Decision-Making via Visual Analysis using the Natural Language Toolkit and R

Musa Jafar
mjafar@wtamu.edu

Jeffry S. Babb
jbabb@wtamu.edu

Kareem Dana
kdana@wtamu.edu

Computer Information and Decision Management
West Texas A&M University
Canyon, TX 79106 USA

Abstract

The preponderance of and rate of accumulation of textual data is now outstretching our ability to comprehend this text using conventional means. We extend our existing framework for the interactive visualization of textual data in digital format by including near-real-time text analysis using the R open source statistical and its analytical package(s). We utilize R as a pre-processor to programmatically gather and preprocess raw textual data generated by social media and incorporate it into textual corpora. The extended framework's back-end is a Django-based framework that relies on both the Natural Language Processing Toolkit (NLTK 2.0) and the R language and its rich set of packages. These tools are combined to present the user with a web-based and interactive n-gram/word cloud front end to visually and statistically analyze corpora built from our backend. We illustrate the use of this framework by utilizing the Twitter API to glean social trends that amount to visualizing zeitgeist. Our framework will allow subject-matter experts, typically in the humanities and social sciences, to develop alternative analyses of social phenomenon through text mining and visualization. The intent of our tool would be that subject-matter experts are able to manipulate text without the technical background in the tools typically used for these analyses, and without having to digest the entire works themselves, which is becoming impossible.

Keywords: Natural Language Processing, R, NLTK, Word cloud, social media, data visualization, corpus linguistics, n-gram.

1. INTRODUCTION

The hallmark of our progress towards the current information age has certainly been our ability to codify and normalize means of communication, both written and oral. As our civilizing has progressed, our written language persists as an artifact from which we can understand our culture, history, governance, civilization, and society. This is so as written language remains a primary means for transacting meaning, intent, values, and knowledge. In recent years, the volume of textual information that is generated by both humans and machines is precipitously exponential such that our means to analyze this data exceeds known human capacity (Farmer, 2010). The enormity of digitized text has transformed data mining, and has given rise to the meta-phenomenon known as "Big Data" (Barlow, 2013; Jacobs, 2009; Zikopoulus, Eaton, deRoos, Deutsch, and Lapis, 2012).

Among the means of digesting this so-called "Big Data" is data visualization supported by machine learning techniques. The visual analysis of plain text presents trends and also the caveats and quirks that have allowed humanity to diversify and coalesce (Norvig, 2009). What is remarkable is that the body of digitized text generated in the last 10 years has surpassed all of the textual knowledge recorded since the beginning of recorded history (Tynan, 2010). In general, we can soon expect that the sum of digitized text will be available for exploration, analysis, and reflection. It is supposed then that visually analyzing these texts, collectively and in the whole, can reveal trends not seen within in the details of any individual text. Moreover, if in the last 10 years we have been almost exclusively generating text in digital format, then roughly half of what we can analyze is very recent. When coupled with digitized historical text, we are able to develop data visualizations that span the human record. Google's effort to digitize over 15-million books and providing public domain frameworks to access those books is one such endeavor (Jean-Baptiste, 2011; Yuri, 2012).

One source of the new text being added to the overall textual record is that being generated by social media. These data are typically informal and represent, collectively, a zeitgeist for humanities and social scientists to examine and explore. The Oxford English Dictionary defines zeitgeist as: "the defining spirit or mood of a particular period of history as shown by the ideas and beliefs of the time." Despite hash tag metadata for categorization, data analysis tools, such as the interactive n-gram word cloud visualization framework presented in this paper, will better enable a grasp of the zeitgeist that near-real-time analysis of social media textual data affords. Note that in computational linguistics, an n-gram is the contiguous sequence of phonemes, syllables, letters, or words that are considered to go together sequentially (for instance, "I love you" is a three-gram sequence).

Recent phenomenon to which the zeitgeist of social media has been ascribed as being partially causal is the Arab Spring social upheaval in 2011 and 2012 in several Middle-Eastern countries (Howard et al., 2011). Many have ascribed the fluidity and feasibility of the Arab Spring as being due to the rapid availability of real-time information via social media channels.

Our objective is to provide a framework and tools where a non-technical researcher or practitioner, perhaps in the digital humanities (such as a journalist, a social scientist, or in any non-technical discipline,) may analyze a body of text without deep knowledge of the underlying technologies used to implement the framework and without having to read the text itself. This last aim is perhaps the most compelling: what can visualization of text tell without having to actually examine the entire text?

2. RELATED WORK

In recent years, word clouds (sometimes referred to as tag clouds) have become a popular means to obtain the gist of a body of text. Such visualizations have been popularized in work of Viegas and Wattenberg (2007, 2009) in their *Many Eyes* and *Wordle* projects. These projects utilize manipulations of font-size, word placement, and word color to display the frequency distribution of words within a document. Viegas et al. (2009: 1137) state: "…users seem to revel in the possible applications of word clouds, scientists wordle genetic functions, fans Wordle music videos, … spouses Wordle love letters, … Wordles have made their way into corporate PowerPoint slides and houses of worship; they have adorned T-shirts, magazine covers, … have graced posters, scrapbooks, birthday cards, and valentines."

These projects, while inherently useful, are somewhat constrained by the minimal interactivity afforded in their presentation, such as the exclusion of words, modification of layout, and direction and color management are not within the users' control.

There remains a wide-spread appeal of word clouds where people have used them in many aspects of life. For instance, Sharma et al. (2012) use a word cloud approach to infer the social connections and networks (a 'who-is-who' approach) to highlight the characteristics of highly-regarded Twitter users by exploiting the metadata in the profiles of those users. The key words in their word cloud provide hyperlinks to more searches on Twitter.

In another case, Baroukh et al. (2011) used a WordCram (Oesper, 2011) word cloud to quickly summarize knowledge of biological terms. As WordCram is Processing-based (Processing being a Java library and framework for visualization), it is potentially not desirable as a browser-based framework which relies on the HTML5 and AJAX capabilities of modern web browsers.

Kim et al. (2011) used a novel and network-oriented node-link approach to analyze test using word clouds where the nodes represent entities or keywords and the links (or edges) represent the relationship or co-occurrences between the entities to build word clouds of the nodes and their relationships.

In general, the aesthetics of presentation and the management of layout and placement of many existing word cloud frameworks are very appealing. In our framework, the aesthetic and layout issues continue to be addressed. However, our approach continues to work towards function and interaction with textual data aggregation and analysis as a primary concern; with the visualization optimizations following this. The advantage of our interactive n-gram word cloud approach is its increased interactivity with the word cloud to access its back-end natural language processing support. Our framework does not merely supply a static/semi-static web-page interface that heavily utilizes JavaScript or a standalone Java applet. Rather, our approach heavily relies on back-end processing, using both the Natural Language Toolkit (NLTK) and R, to analyze text beyond static presentation as a word cloud. Our framework provides a user interface to the NLTK package for the purpose of analyzing the text and inferring knowledge about it.

A major extension of our framework, developed to address an emergent need, is our use of R in addition to the NLTK, to examine near-real-time social media data, such as what can be obtained from Twitter. Social media (Twitter, Facebook, Google Plus, etc.) and news media (New York Times and others) companies are increasingly providing free web-oriented Applications Programming Interfaces (APIs) access to their data. In recognition to the fact that we are generating new text data at a highly rapid pace, we recognize the need to retool our framework to accommodate the data available through these APIs. As the NLTK generally lacks deep statistical analytics, we have also incorporated R to assist in obtaining data from these APIs. Ultimately, our aim is to provide a portal for both subject-matter experts and laypersons to see the emergent zeitgeist inherent in our rich corpora of text.

3. OUR FRAMEWORK

Both the NLTK and R are at the center of our approach and framework. Recent additions to our framework include an ability to generate new corpora from a collection of separate files, to generate new corpora from the contents of a compressed Zip archive, and to provide more interactivity in the word cloud interface. The interface improvements increase interactivity by allowing users to select words for inclusion or exclusion and allowing for a side-by-side comparison between the modified and original word clouds.

We introduced R into our framework for its superior statistical and analytical capabilities. The contribution of the open source community to R made it the tool of choice for statistical and analytical data processing. The NLTK is a vital component as it is a very capable library for accessing various corpora and lexical resources for the classification, tokenization, stemming, tagging, parsing, and semantic reasoning required as a prerequisite for the visual analysis of digitized text. Thus, we require the NLTK as a precursor for successful visualization as our visualization engine requires the outputs of computational linguistics. Furthermore, we require R to assist in analysis and to utilize its own extensive libraries and extensions to assist in obtaining social media data and building corpora of it.

Figure 1 shows the dashboard for our framework. Figure 5, 4, 5 and 6 are screenshots of the various modules of our framework. For the purpose of this paper, and to illustrate the current state of our framework, we used R and several R-packages (*twitteR*, *ROAuth*, *RJSONIO*) geo-locate the White House (President of the United States residence). We then requested 1000 tweets that have the word "white house" in them which originated within 5 miles of the White House. We then used the *tm*, *RWeka* and *Snowball* R-Packages to preprocess and clean the data and then generate a corpus based on the 1000 found tweets (Figure 7). While deeper analysis is possible within R, R mainly serves as a pre-processor in our architecture at the moment. We then used our framework's administrative interface to load the 1000-Tweets corpus into the framework for further analysis.

With our framework, we can create, edit or delete corpora, and build a corpus by adding (deleting) new files into an existing corpus. Another means of upload is to obtain text data from a compressed Zip archive; such as those containing 1000 text files based on tweets pertaining to a certain subject or from a given user. The framework allows the user to analyze the whole corpus or a single document, by selecting the *generate word cloud* function of the framework. This creates a one-gram word cloud with the NLTK's point-wise mutual information (PMI) scoring method as the default. A two-gram or three-gram cloud can also be generated by the user along with various statistical measures such as t-tests, chi-squared tests, and others. For this transformation to occur, all documents are pre-processed in our framework using NLTK. Once tokenized into n-grams, we can use our visual interface to highlight a token. For instance, a user could click on the two-gram "geese honking" in the word cloud to see all the sentences of the token across the different documents encompassed by the corpus. We also provide a sorted list of all the tokens/frequencies as well as a regular expression search to find all of the token's usage in the body of text and the corresponding word-cloud grams.

What sets our approach apart from others that we have examined is that our approach uses the word cloud as an interactive n-gram viewer for the purpose of text analytics. We do not only provide an n-gram word cloud, we also use the word cloud as an interactive analysis tool of the corpus. Thus, our Django-based web interface (both the administrative and user interfaces) serve as the front-end to an NLTK-driven (post processing) and R-driven (pre processing for now) back-end. After a document is rendered, we use AJAX to communicate with the backend to facilitate a user's ad-hoc queries. In addition to supporting, one-, two-, and three-gram word analysis (tokenization), we further allow a user to drill-down to the underlying sentences of the tokens and to selectively filter out the x-gram that is being displayed via interaction with the visualization graphics directly.

4. OUR ARCHITECTURE

Figure 6 illustrates the architecture of our dashboard that connects extensively with the NLTK. One of our design goals was to make the NLTK more visual and more accessible to the end user. To that end, we designed the NLTK Command Center as a web application. The frontend runs in a modern browser. It utilizes JavaScript, AJAX (Mozilla 2013), JSON (Mozilla 2013), jQuery (jQuery 2013), HTML5 (specifically the HTML5 Canvas and Local Storage features), and CSS. These latest web technologies enabled us to write a fully-featured, graphical, and interactive web application. We implemented an existing open-source HTML 5 Canvas-based word cloud library named *HTML 5 Word Cloud* written by Chien (2012).

The backend architecture is written in the Python programming language as the NLTK is a Python library. Accordingly, we used Django, which is a Python-based web application framework, as it can directly interface with the NLTK. For data persistence, we use a SQLite embedded database to store user session information and metadata regarding analyzed documents and corpora. The Django Framework provides a development web server to host and test Django applications during the development process. Our previous iteration of the architecture (Jafar, Babb, and Dana, 2012) utilized the Apache web server and MySQL database server. With this recent iteration, we now use the Django development web server and SQLite database. While Apache and MySQL are excellent tools, we have switched to the embedded web server and SQLite as both are lightweight in that they are easier to develop, test, and deploy. This switch has saved a lot of development time that would otherwise be spent maintaining and managing server infrastructure. Despite our entirely open-source implementation approach, our framework can be deployed on any other platform that Django supports, such as Windows and Internet Information Server (IIS).

By removing the dependencies of Apache and MySQL from the NLTK Command Center architecture, we have addressed previous design challenges of performance and scalability for our architecture. This is so as, in the current iteration of our framework, each user hosts an instance of the application on their client end. Such an approach paves the way to a flexible and extensible deployment solutions such as

Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) options. Accordingly, as it matures, we can refactor our architecture to support the Google App Engine or another cloud based service.

To facilitate our design goals, we added the ability for users to easily (through an intuitive web interface) create and modify entire corpora instead of just individual documents as was the case with our initial architecture. Corpora are stored directly on the file system and metadata including corpus name and contents are stored in the SQLite database. This feature allows users to combine similar documents into a single corpus and perform analysis.

Finally, our architecture now supports the vast lexical database of the Princeton WordNet through the NLTK-WordNet interface (Miller 1995, Fellbaum 2005). WordNet provides part of speech tagging, synonyms, antonyms, definitions, and other lexical data that enhance the usefulness of our tool, especially when presented in a visual manner. Currently our interface with WordNet is rudimentary. We do envision a more robust interface for the purpose of word-sense-disambiguation, part-of-speech tagging and sentiment analysis and classification purposes.

5. IMPLEMENTATION DETAILS

When a user first connects to the NLTK Command Center website they are presented with a graphical user interface that allows them to perform various NLTK tasks (
Figure 2). This is facilitated by JavaScript running on the client's web browser that sends an AJAX call to the web server, and ultimately to the NLTK, which then performs the task and sends the results back serialized as JSON. We then use JavaScript to display the results graphically. Currently we provide the following features through a combination of JavaScript and AJAX-oriented Python methods which provide responsive access to the NLTK services in our framework:

- **Create Corpus** – Users of the NLTK Command Center can create a corpus (one or many) through the Django administrative interface. Text documents can be uploaded one at a time or many at once through a Zip archive. The user can add those documents to one or many corpora. This feature is usually the first step to using our application. Once a corpus is created and populated with texts, it can be visually analyzed by any of our other features.

- **Draw Word Cloud** – This is the main feature of our web application. One can select a document, multiple documents, or an entire corpus and draw a word cloud representation of those texts. Words or phrases that appear with the highest frequencies will appear larger in the word cloud. The word cloud is also interactive, so one can click on a word or phrase and get more information including the part-of-speech, definition, and frequency. Implementation wise, drawing the word cloud is broken down into two distinct steps: analyzing the text using the NLTK on the back end and rendering the word cloud itself in the user's web browser using HTML 5 and JavaScript.

- **Exclude Word Table** – When a user draws a word cloud, a table of all the words and phrases and their frequencies is also available. This table can be searched or sorted by word or frequency. Words can be excluded or explicitly included by using this table and the word cloud redrawn. Certain words may not be useful or interesting – this is where those words can be excluded all at once and only the words or phrases that the user is interested in can be shown in the word cloud. Clicking on a word in this table also highlights it in the word cloud making it very easy to find any word from any document.

- **Compare Word Clouds** – Using the same exclude word table, a user can fork multiple word clouds for side by side comparison. A new window will open in the web browser with the newly drawn word cloud so the user can compare multiple word clouds.

- **Highlight Words** – Being able to find words or phrases in the word cloud is a useful feature, especially if one draws a word cloud with hundreds of words. Using the exclude word table as describe above, one can sort the words or search for a specific word and then click on it to highlight that word in the word cloud. Clicking again will undo the highlight. This is particularly useful for low frequency words that appear small in the word cloud but might still have some importance to the user or text in context.

- **Show Sentences** – This functionality allows the user to right-click on a word or phrase in the word cloud and request a list of all the sentences that word or phrase appears in. The NLTK processes each sentence of the text and looks for the phrase. The list of sentences is then returned as JSON and displayed in the browser. The phrase is highlighted in each sentence while it is displayed in the word cloud visualization.

  Being able to analyze multiple documents or an entire corpus means that a single word or phrase in the word cloud may actually appear in many sentences across multiple documents. We implemented a sentence-document map in Python that maps sentences to the document they are in and provide that information back to the user. Showing sentences will also show which documents those sentences are from.

- **WordNet Analysis** – In addition to showing the sentences a word appears in, right-clicking on a word (1gram support only) will also provide WordNet data about that word. Currently we display part-of-speech tagging, synonyms, and the definition.

In Figure 2 we illustrate general end-user workflow while using the framework. Users may select and/or upload their documents and render a word cloud from it. Interactivity allows for the user to exclude words and generate a modified word cloud which can then be compared to the original. If the text is new, it can be included an existing corpus or a new corpus can be generated. Textual analysis can be performed such that a sentence-document map is obtained. Analysis is performed against Princeton's WordNet lexical database for English (Felbaum, 2005). Though not yet implemented in our framework entirely, sentiment analysis is an area of great interest to us, especially the availability of social media data.

6. SUMMARY AND FUTURE WORK

With the rapid growth of textual data presaged by Big Data, we anticipate the need to perform textual analysis on a daily basis. We envision an end user who may lack the technical skills to issue hundreds of R and NLTK commands to perform these repetitive tasks. We present a framework that provides a layer of abstraction, and the underlying infrastructure, to facilitate these tasks. We utilized existing open source frameworks (NLTK, R, R-packages, Word Clouds, Django, etc.) to compound a framework that can address this need.

It is important to note that this architecture facilitates growth for semantic analysis by way of its use of R and packages. The interactive analysis also facilitates a visual sense-making for the user that sits between machine-learning semantic analysis and statistical summarization of the frequency of words.

In summary, we provide a framework that allows a user with minimal technical experience to create, delete and modify corpora generated from digital text documents. The user is able to interactively analyze their corpora with an interactive n-gram word cloud viewer, include and exclude n-grams from their analysis, and drill down to the row sentences and their corresponding text documents. This technical approach is possible by utilizing R as a preprocessor and the NLTK for post processing and analysis.

With our interactive n-gram word cloud framework, our objective is to create tools that allow users to be engaged in the discovery of hidden characteristics and meanings ensconced within digital text. Our premise rests within the context of the Big Data such that full comprehension of knowledge from texts will require information processing tools such as our framework. Facilitating knowledge through the design and implementation of information systems has long been a focus for the information systems discipline and, as such, we see the need for improving information interfaces with information visualization. Toward this end, we continue to progress our framework with an aim to enable those with minimum computing knowledge to analyze digital text through interactive visualization. Our latest iteration of this framework seeks to not only serve as a decision support system, but to also develop a greater awareness of the meaning of these texts. For instance, we envision the humanities researcher who may be well equipped to understand intrinsic and underlying literary and historical context of the text, but is inhibited from analyzing the data of the text by its sheer volume. We anticipate that the outputs of our research will allow for growth and impact not only in our field, but also in the humanities, in business, and in any areas where digitized text is available and text and data analysis is important.

As we grapple with the Big Data phenomenon, it is clear that our tools will need to take further analytical steps to tell the user what a body of text means. At the root of meaning is an aggregation and generalization of what other people think. In fact, much of the growth of textual data is opinion-rich: online communities, review sites, personal blogs, etc. (Pang and Lee, 2008). Our illustration using Twitter data is not an entirely new development and it makes clear the potential for the inclusion of sentiment analysis into our framework (Luce, 2012; Pak and Paroubek, 2010). While our emergent framework started as a generic tool for text representation, we realize the need for the tool's utility to grow such that it not only beautifies information, but also guides the user. Algorithmic approaches can render an initial portrayal of sentiment from which the end user, novice or expert, can then draw further conclusions. We have no doubt that attempts to enter into the foray of sentiment analysis will raise concerns for subjectivity (Liu, 2010), however we are interested in providing tools for others to make decisions.

Finally, our framework is designed to allow a user to analyze text without reading it. While this may seem counter-intuitive, decisions on how to use precious time in extracting value from text may be improved by seeing a text before reading it. Moreover, our framework is moving towards going beyond interactive visualization to performing sentiment analysis, and other Big Data-oriented tasks, in order to make the volume of data digestible. If successful, we envision our framework to be an interactive web-based friendly layer on top of the Natural Language Processing toolkit (NLTK) that utilizes the power of R and R packages for the pre-processing of the raw data and for the analytical aspects of digital textual analysis.

7. REFERENCES

Baroukh, C., Jenkins, S. L., Dannenfelser, R. and Ma'ayan A. (2011). Genes2WordCloud: a quick way to identify biological themes from gene lists and free text. *Source Code for Biology and Medicine*, 6(15).

Barlow, M. (2013). Big data vs. Big reality. O'Reilly Radar. http://radar.oreilly.com/2013/06/big-data-vs-big-reality.html. Accessed June 28, 2013.

Howard, P.N., Duffy, A., Freelon, D., Hussain, M., Mari, W. & Mazaid, M. (2011). Opening Closed Regimes: What Was the Role of Social Media During the Arab Spring? Seattle: PIPTI. Retrieved June 28, 2013 from http://pitpi.org/index.php/2011/09/11/opening-closed-regimes-what-was-the-role-of-social-media-during-the-arab-spring/

Farmer, D. (2010). Study Projects Nearly 45-Fold Annual Data Growth by 2020. EMC Corporation. http://www.emc.com/about/news/press/2010/20100504-01.htm. Accessed June 28, 2013.

Fellbaum, Christiane (2005). WordNet and wordnets. In: Brown, Keith et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Elsevier, Oxford, 665-670

Chien, T., G. (2012). HTML 5 Word Cloud. http://timc.idv.tw/wordcloud/en/. Accessed June 28, 2013.

Jacobs, A. (2009) The Pathologies of Big Data, *ACM Queue*, 7(6), 10.

Jafar, M., Babb, J., and Dana, K. (2012). A Framework for an Interactive Word-Cloud Approach for Visual Analysis of Digital Text using the Natural Language Toolkit. *2012 Proceedings of the Conference on Information Systems Applied Research*. November 1-4, 2012. pp. 10.

Jean-Baptiste, M., et.al. (2011). Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, 332, 176.

Kim K., Ko, S. Elmqvist, N., and Ebert, D. (2011) WordBridge: Using Composite Tag Clouds to Node-Link Diagrams for Visualizing

Content and Relations in Text Corpora. *44th Hawaii International Conference on System Sciences (HICSS)*.

jQuery Foundation (2013) http://jquery.com/ Accessed Sept 13 2013

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2, 568.

Luce, L. (2012). Twitter sentiment analysis using Python and NLTK. *Technical blog on web technologies*. http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/ Accessed July 03, 2013.

Miller, A. George (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

Mozzilla Developer Network (2013) AJAX https://developer.mozilla.org/en-US/docs/AJAX
Accesses Sept 13, 2013.

Norvig, Peter(2009). Natural Language Corpus Data, in Segaran, T. and Hammerbacher, J (eds.) *Beautiful Data* (219-240), O'Reilly Pub.

Oeper, L., Meico, D., Isserlin, R. and Bader, G. D. (2011) WordCloud: Cytoscape plugin to create a visual semantic summary of networks. *Source Code for Biology and Medicine*, 6(7).

Pak, A., & Paroubek, P. (2010, May). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In LREC.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and trends in information retrieval, 2(1-2), 1-135.

Sharma, N., Ghosh, S., Benevnuto, F., Ganguly, N. and Gummadi, K. P. (2012). Inferring Who-is-Who in the Twitter Social Network. *ACM SIGCOMM Workshop on Online Social Networks (WOSN'12)*, Helisinki, Finland.

Tynan, D. (2010). Prepare for Data Tsunami, Warns Google CEO. ITWorld, August 6, 2010. http://www.pcworld.com/article/202817/prepare_for_data_tsunami_warns_google_ceo.html. Accessed June 28, 2013.

Viegas, F. B., Wattenberg, M., Ham F. V. Kriss J. and McKeon M. (2007). Many Eyes: A Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6).

Viegas, F. B., Wattenberg, M., Feinberg J. (2009). Participatory Visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15(6).

Yatani, K., Novati, M., Trusty, A., and Troung, K.N. (2011) Analysis of Adjective-Noun Word Pair Extraction Methods for Online Review Summarization. *Proceedings of the 22nd International Join Conference on Artificial Intelligence*.

Yuri L., et.al. (2012) Syntactic Annotations for the Google Books Ngram Corpus. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (169-174).

Zikopoulos, Eaton, deRoos, Deutsch, and Lapis, (2012). Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw Hill, New York.

Figures

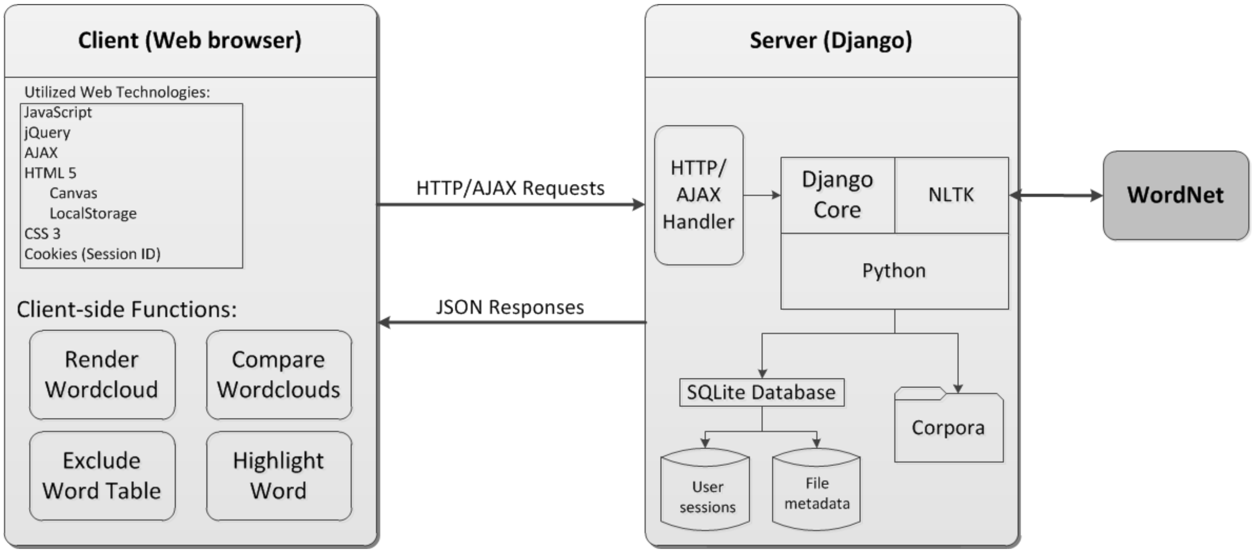


Figure 1 The Framework's Application Architecture

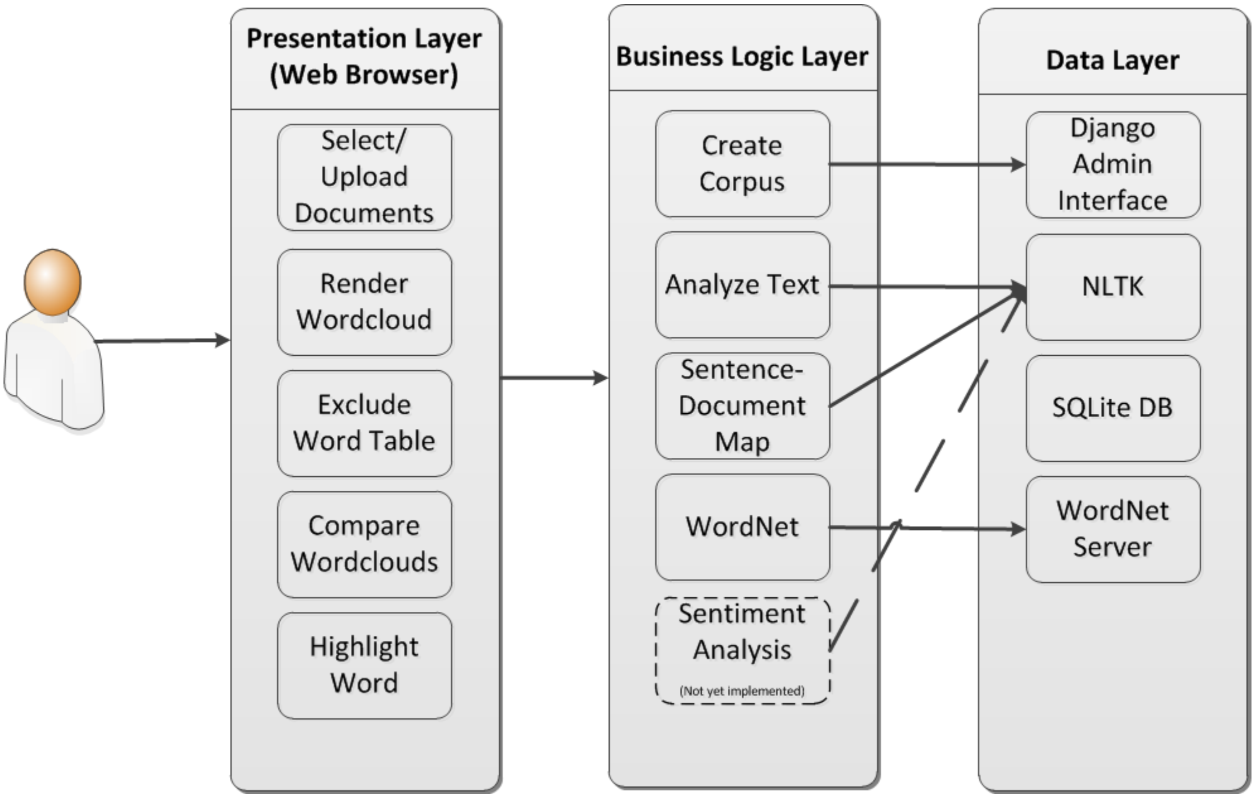
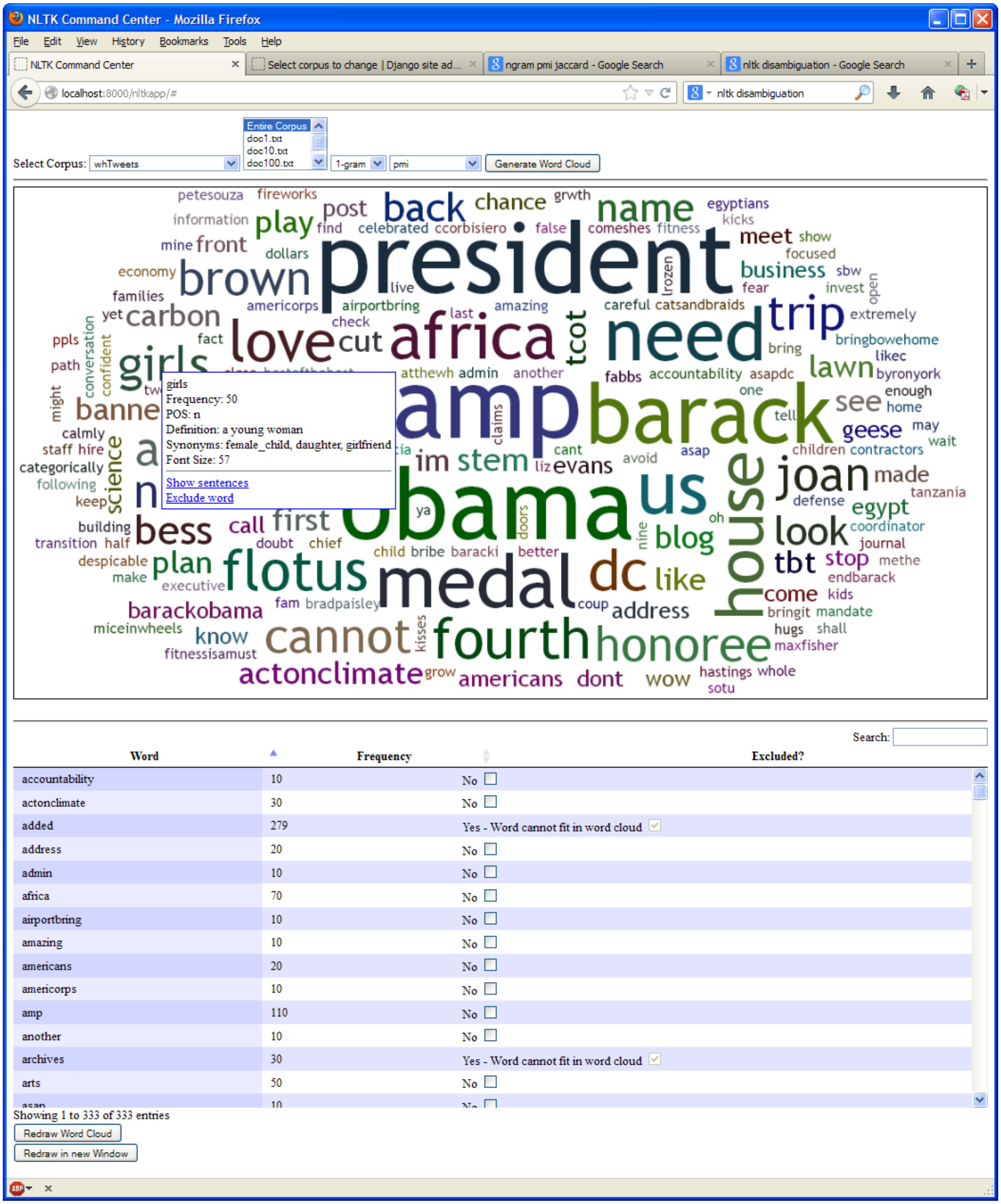


Figure 2 User Workflow through the Framework



Select Corpus: whTweets | 1-gram | pmi

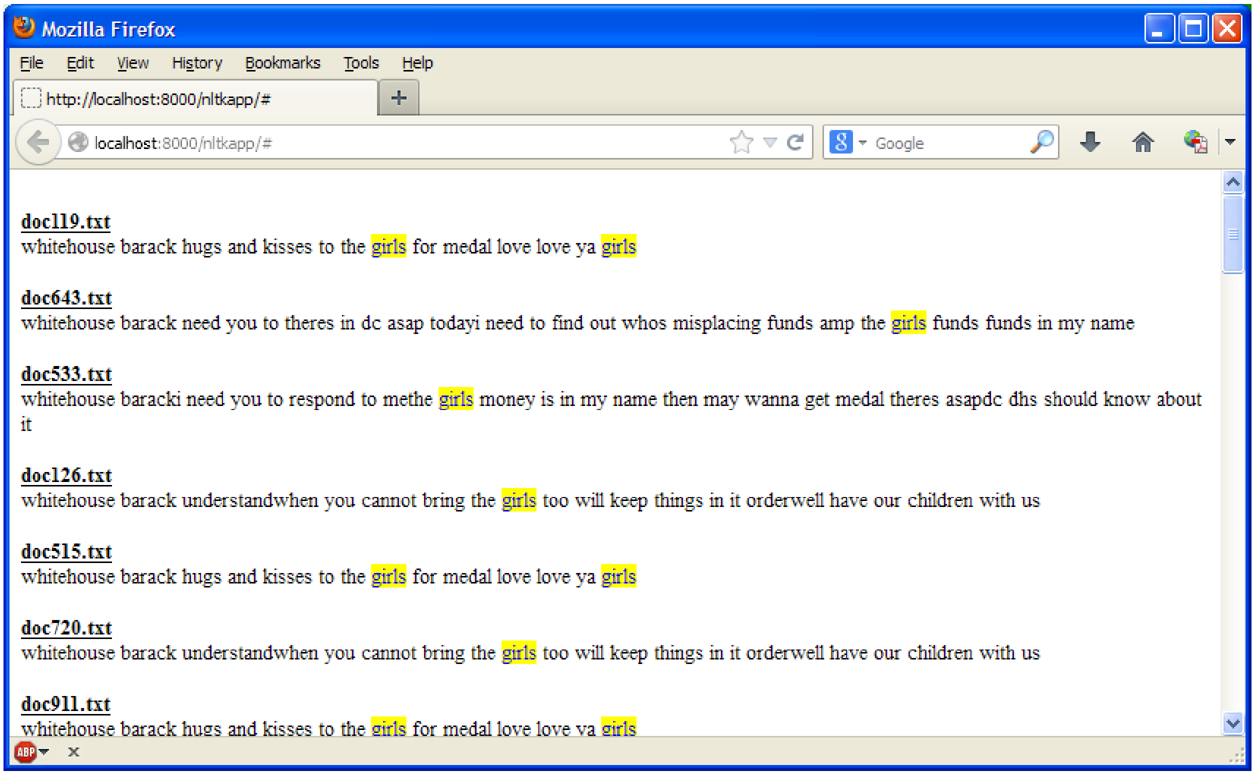
girls
Frequency: 50
POS: n
Definition: a young woman
Synonyms: female_child, daughter, girlfriend
Font Size: 57
Show sentences
Exclude word

Search:

Word	Frequency	Excluded?
accountability	10	No ☐
actonclimate	30	No ☐
added	279	Yes - Word cannot fit in word cloud ☑
address	20	No ☐
admin	10	No ☐
africa	70	No ☐
airportbring	10	No ☐
amazing	10	No ☐
americans	20	No ☐
americorps	10	No ☐
amp	110	No ☐
another	10	No ☐
archives	30	Yes - Word cannot fit in word cloud ☑
arts	50	No ☐
asap	10	No ☐

Showing 1 to 333 of 333 entries

Figure 3 An Interactive 1gram word cloud visualization of 1000 tweets within 5 miles off 1600 Pennsylvania Ave NW Washington, DC 20500


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8000/nltkapp/#
localhost:8000/nltkapp/#
Google

doc119.txt
whitehouse barack hugs and kisses to the girls for medal love love ya girls

doc643.txt
whitehouse barack need you to theres in dc asap todayi need to find out whos misplacing funds amp the girls funds funds in my name

doc533.txt
whitehouse baracki need you to respond to methe girls money is in my name then may wanna get medal theres asapdc dhs should know about it

doc126.txt
whitehouse barack understandwhen you cannot bring the girls too will keep things in it orderwell have our children with us

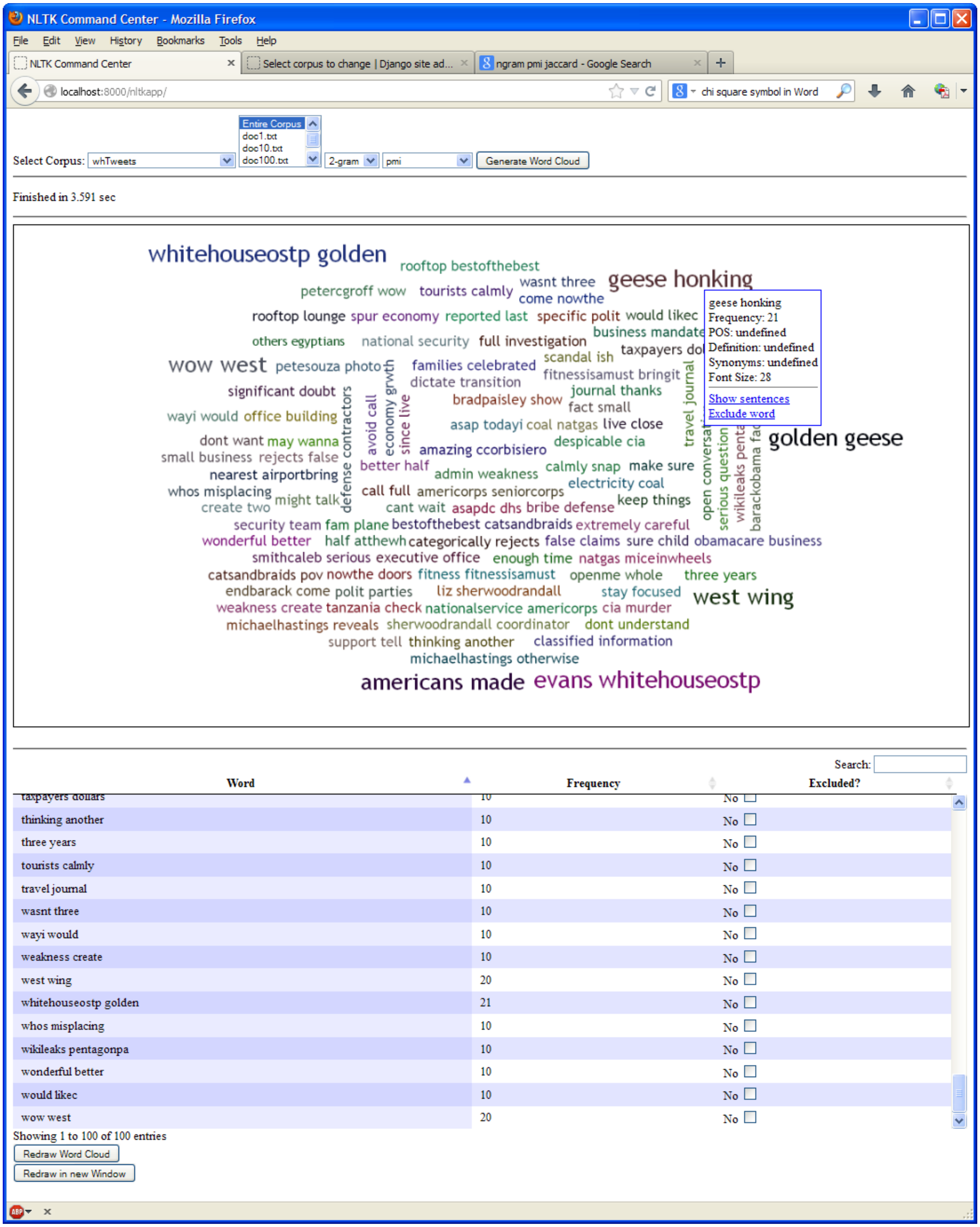
doc515.txt
whitehouse barack hugs and kisses to the girls for medal love love ya girls

doc720.txt
whitehouse barack understandwhen you cannot bring the girls too will keep things in it orderwell have our children with us

doc911.txt
whitehouse barack hugs and kisses to the girls for medal love love ya girls

Figure 4 A Drill Down on girls from

Figure 3 show the various tweets that contained the word



NLTK Command Center - Mozilla Firefox
File Edit View History Bookmarks Tools Help
NLTK Command Center
Select corpus to change | Django site ad...
ngram pmi jaccard - Google Search
localhost:8000/nltkapp/
chi square symbol in Word
Generate Word Cloud
whitehouseostp golden
rooftop bestofthebest
petercgroff wow tourists calmly wasnt three come nowthe geese honking
rooftop lounge spur economy reported last specific polit would likec
others egyptians national security full investigation business mandate taxpayers dol
wow west petesouza photo families celebrated dictate transition scandal ish fitnessisamust bringit
significant doubt bradpaisley show journal thanks fact small
wayi would office building asap todayi coal natgas live close
dont want may wanna amazing ccorbisiero despicable cia golden geese
small business rejects false better half admin weakness calmly snap make sure
nearest airportbring call full americorps seniorcorps electricity coal
whos misplacing might talk cant wait asapdc dhs bribe defense keep things
create two
defense contractors avoid call economy grwth since live travel journal open conversat serious question wikileaks penta barackobama fac
security team fam plane bestofthebest catsandbraids extremely careful
wonderful better half atthewh categorically rejects false claims sure child obamacare business
smithcaleb serious executive office enough time natgas miceinwheels
catsandbraids pov nowthe doors fitness fitnessisamust openme whole three years
endbarack come polit parties liz sherwoodrandall stay focused west wing
weakness create tanzania check nationalservice americorps cia murder
michaelhastings reveals sherwoodrandall coordinator dont understand
support tell thinking another classified information
michaelhastings otherwise
americans made evans whitehouseostp
Redraw Word Cloud
Redraw in new Window

Select Corpus: whTweets

Entire Corpus
doc1.txt
doc10.txt
doc100.txt

2-gram | pmi

Finished in 3.591 sec

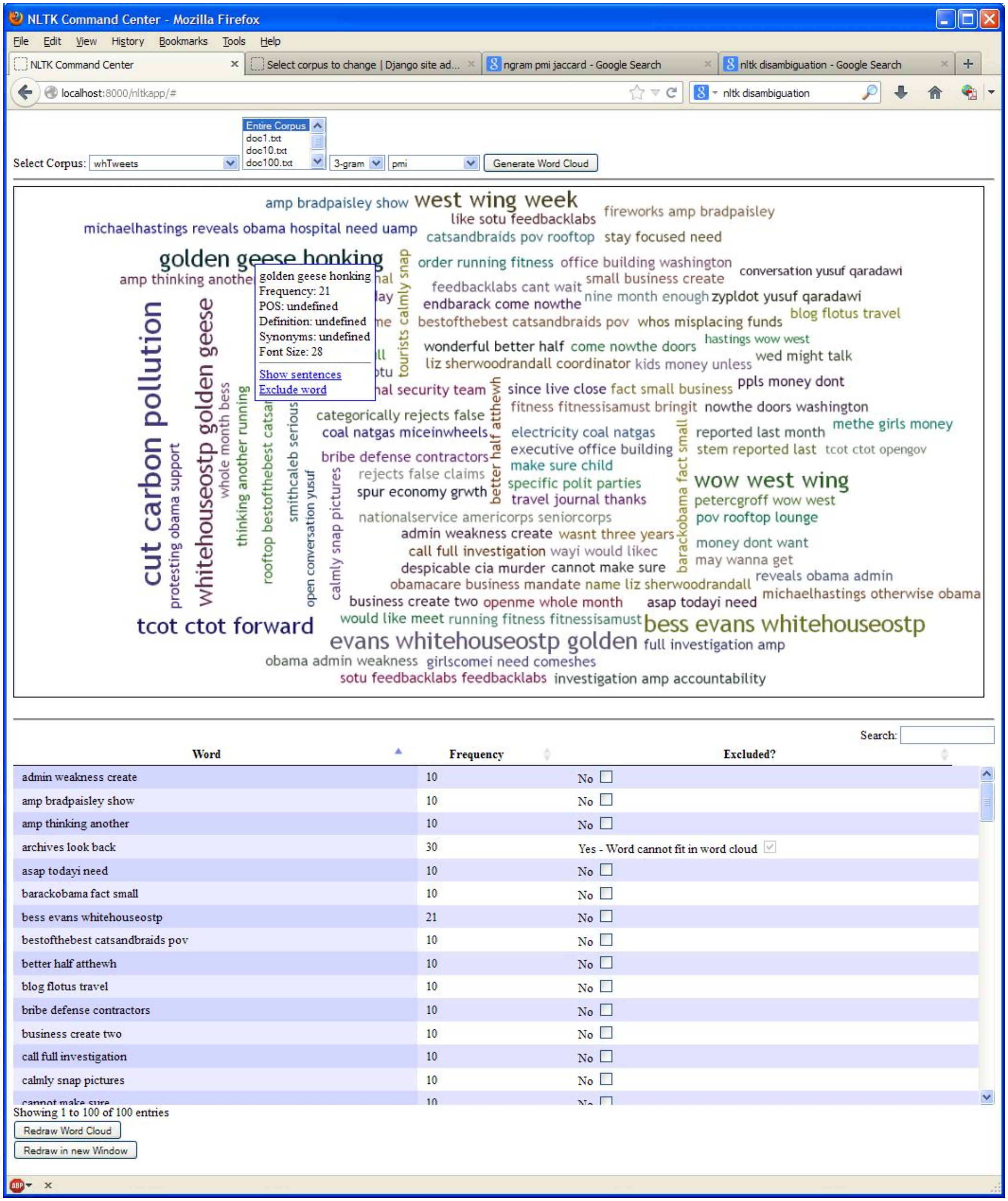
geese honking
Frequency: 21
POS: undefined
Definition: undefined
Synonyms: undefined
Font Size: 28
Show sentences
Exclude word

Search:

Word	Frequency	Excluded?
taxpayers dollars	10	No ☐
thinking another	10	No ☐
three years	10	No ☐
tourists calmly	10	No ☐
travel journal	10	No ☐
wasnt three	10	No ☐
wayi would	10	No ☐
weakness create	10	No ☐
west wing	20	No ☐
whitehouseostp golden	21	No ☐
whos misplacing	10	No ☐
wikileaks pentagonpa	10	No ☐
wonderful better	10	No ☐
would likec	10	No ☐
wow west	20	No ☐

Showing 1 to 100 of 100 entries

Figure 5 An Interactive 2gram word cloud visualization of 1000 tweets within 5 miles off 1600 Pennsylvania Ave NW Washington, DC 20500

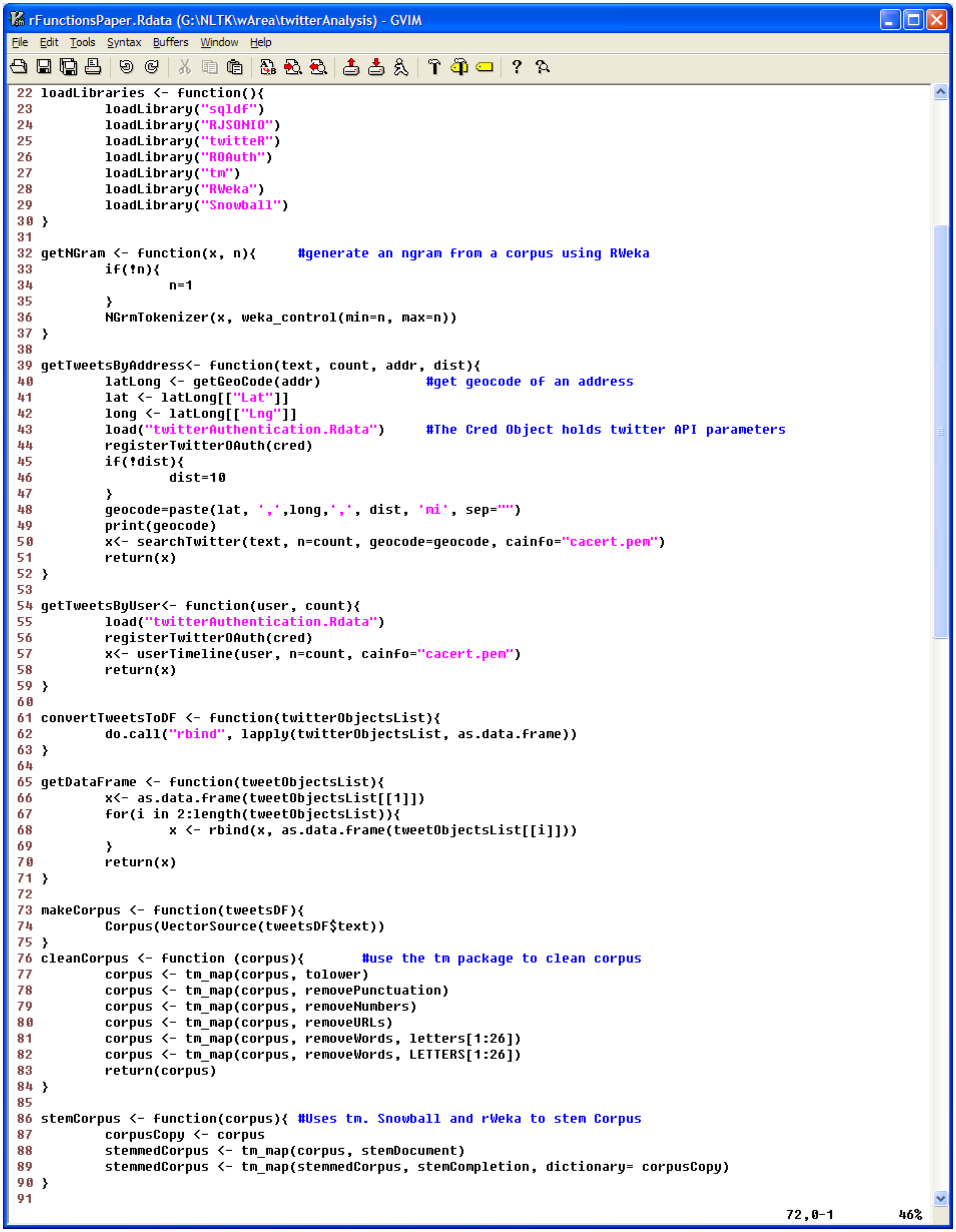


golden geese honking
Frequency: 21
POS: undefined
Definition: undefined
Synonyms: undefined
Font Size: 28

Word	Frequency	Excluded?
admin weakness create	10	No □
amp bradpaisley show	10	No □
amp thinking another	10	No □
archives look back	30	Yes - Word cannot fit in word cloud ☑
asap todayi need	10	No □
barackobama fact small	10	No □
bess evans whitehouseostp	21	No □
bestofthebest catsandbraids pov	10	No □
better half atthewh	10	No □
blog flotus travel	10	No □
bribe defense contractors	10	No □
business create two	10	No □
call full investigation	10	No □
calmly snap pictures	10	No □
cannot make sure	10	No □

Showing 1 to 100 of 100 entries

Figure 6 An Interactive 3gram word cloud visualization of 1000 tweets within 5 miles off 1600 Pennsylvania Ave NW Washington, DC 20500


rFunctionsPaper.Rdata (G:\NLTK\wArea\twitterAnalysis) - GVIM
File Edit Tools Syntax Buffers Window Help
72,0-1
46%

```
loadLibraries <- function(){
        loadLibrary("sqldf")
        loadLibrary("RJSONIO")
        loadLibrary("twitteR")
        loadLibrary("ROAuth")
        loadLibrary("tm")
        loadLibrary("RWeka")
        loadLibrary("Snowball")
}

getNGram <- function(x, n){      #generate an ngram from a corpus using RWeka
        if(!n){
                n=1
        }
        NGrmTokenizer(x, weka_control(min=n, max=n))
}

getTweetsByAddress<- function(text, count, addr, dist){
        latLong <- getGeoCode(addr)             #get geocode of an address
        lat <- latLong[["Lat"]]
        long <- latLong[["Lng"]]
        load("twitterAuthentication.Rdata")     #The Cred Object holds twitter API parameters
        registerTwitterOAuth(cred)
        if(!dist){
                dist=10
        }
        geocode=paste(lat, ',',long,',', dist, 'mi', sep="")
        print(geocode)
        x<- searchTwitter(text, n=count, geocode=geocode, cainfo="cacert.pem")
        return(x)
}

getTweetsByUser<- function(user, count){
        load("twitterAuthentication.Rdata")
        registerTwitterOAuth(cred)
        x<- userTimeline(user, n=count, cainfo="cacert.pem")
        return(x)
}

convertTweetsToDF <- function(twitterObjectsList){
        do.call("rbind", lapply(twitterObjectsList, as.data.frame))
}

getDataFrame <- function(tweetObjectsList){
        x<- as.data.frame(tweetObjectsList[[1]])
        for(i in 2:length(tweetObjectsList)){
                x <- rbind(x, as.data.frame(tweetObjectsList[[i]]))
        }
        return(x)
}

makeCorpus <- function(tweetsDF){
        Corpus(VectorSource(tweetsDF$text))
}
cleanCorpus <- function (corpus){       #use the tm package to clean corpus
        corpus <- tm_map(corpus, tolower)
        corpus <- tm_map(corpus, removePunctuation)
        corpus <- tm_map(corpus, removeNumbers)
        corpus <- tm_map(corpus, removeURLs)
        corpus <- tm_map(corpus, removeWords, letters[1:26])
        corpus <- tm_map(corpus, removeWords, LETTERS[1:26])
        return(corpus)
}

stemCorpus <- function(corpus){ #Uses tm. Snowball and rWeka to stem Corpus
        corpusCopy <- corpus
        stemmedCorpus <- tm_map(corpus, stemDocument)
        stemmedCorpus <- tm_map(stemmedCorpus, stemCompletion, dictionary= corpusCopy)
}

```

Figure 7 Some of the R extraction and pre-processing code