Modeling and Detection Techniques for Counter-Terror Social Network Analysis and Intent Recognition

Clifford Weinstein, William Campbell, Brian Delaney, Gerald O'Leary
MIT Lincoln Laboratory [1][2][3]
244 Wood Street
Lexington, MA 02421
781-981-7621
cjw@ll.mit.edu

Abstract—In this paper, we describe our approach and initial results on modeling, detection, and tracking of terrorist groups and their intents based on multimedia data [Popp 2006]. While research on automated information extraction from multimedia data has yielded significant progress in areas such as the extraction of entities, links, and events [Doddington 2004], less progress has been made in the development of automated tools for analyzing the results of information extraction to "connect the dots." Hence, our Counter-Terror Social Network Analysis and Intent Recognition (CT-SNAIR) work focuses on development of automated techniques and tools for detection and tracking of dynamically-changing terrorist networks as well as recognition of capability and potential intent. In addition to obtaining and working with real data for algorithm development and test, we have a major focus on modeling and simulation of terrorist attacks based on real information about past attacks. We describe the development and application of a new Terror Attack Description Language (TADL), which is used as a basis for modeling and simulation of terrorist attacks. Examples are shown which illustrate the use of TADL and a companion simulator based on a Hidden Markov Model (HMM) structure to generate transactions for attack scenarios drawn from real events. We also describe our techniques for generating realistic background clutter traffic to enable experiments to estimate performance in the presence of a mix of data. An important part of our effort is to produce scenarios and corpora for use in our own research, which can be shared with a community of researchers in this area. We describe our scenario and corpus development, including specific examples from the September 2004 bombing of the Australian embassy in Jakarta and a fictitious scenario which was developed in a prior project for research in social network analysis. The scenarios can be created by subject matter experts using a graphical editing tool. Given a set of time ordered transactions between actors, we employ social network analysis (SNA) algorithms as a filtering step to divide the actors into distinct communities before determining intent. This helps reduce clutter and enhances the ability to determine activities within a specific group. For modeling and simulation purposes, we generate random networks with structures and properties similar to real-world social networks. Modeling of background traffic is an important step in generating classifiers that can separate harmless activities from suspicious activity. An algorithm for recognition of simulated potential attack scenarios in clutter based on Support Vector Machine (SVM) techniques is presented. We show performance examples, including probability of detection versus probability of false alarm tradeoffs, for a range of system parameters.

TABLE OF CONTENTS

1

[1] 978-1-4244-2622-5/09/$25.00 ©2009 IEEE.

[2] IEEEAC paper #11526, Version 9, Updated 2008:10:24

[3] This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

1. INTRODUCTION

The increasing complexity and irregularity of threats to national security, as infamously exemplified by the tragic events of September 11, 2001, have motivated a great deal of R&D on the application of advanced information technology to help in the analysis, anticipation, and countering of these threats. Excellent and comprehensive descriptions of relevant work in this important area are presented in the recent book on *Emergent Information Technologies and Enabling Policies for Counter-Terrorism*, edited by Robert L. Popp and John Yen [Popp 2006]. In this paper, we describe our approach and results to date on a particular aspect of the application of information technology to counter-terrorism. Specifically, we describe our approach and initial results on modeling, detection, and tracking of terrorist groups and their intents based on extraction of information from multimedia data. While research on automated information extraction from multimedia data has yielded significant progress [Doddington 2004], [Olive 2008], less progress has been made in the development of automated tools to "connect the dots." Hence, our Counter-Terror Social Network Analysis and Intent Recognition (CT-SNAIR) work focuses on development of automated techniques and tools for detection and tracking of dynamically-changing terrorist networks as well as recognition of capability and intent.

In carrying out this work, we have built wherever possible on the prior research of others, many of whom are cited in the references. Our approach to statistical modeling and recognition of attack scenarios owes a great deal to the work of Krishna Pattipati and colleagues, see. e.g, [Pattipati 2006]. Our work on social network analysis is built upon the work of many others, including especially Kathleen Carley and colleagues e.g., [Carley 2008] who provided us with software tools that we could directly exploit. Our work on information extraction has directly utilized systems developed by BBN for automatic content extraction [Boschee 2005].

In the CT-SNAIR project, we have endeavored to build beyond these prior efforts and others to develop a comprehensive, end-to-end approach to SNAIR. Our approach includes: a substantial effort in modeling and simulation of terrorist networks and attacks, so that a significant set of end-to-end experiments in SNAIR can be conducted; development of a new language, which we refer to as Terror Attack Description Language (TADL), to aid in modeling networks and attacks; an approach to social network analysis which emphasizes utilization of extraction of information from content, as well as on specific links identified from communication patterns; and intent recognition based on detection of activity patterns which are similar to previously-modeled attack patterns. These efforts are described in the ensuing sections.

When we initiated this work, we found that it was very difficult to obtain truth-marked data against which SNAIR algorithms could be tested. This has been a part of our motivation for focusing a substantial portion of our effort on modeling and simulation. In addition to SNA and IR algorithm development, one of our goals is to produce a corpus of truth-marked data to enable further R&D in this area. Progress and future plans on development of such a corpus will be discussed later in the paper.

2. SYSTEM FRAMEWORK

An overview of the CT-SNAIR system framework is shown in Figure 1. The input to the system is a large volume of raw multimedia data, which would include voice, text, network sessions, sensor data, reports, and other sources. We assume that the data includes both information about the data (e.g., source and destination address) and the contents of the data. The information processing block extracts information primarily from the contents of the data. We are particularly interested here in the extraction of entities, links or relations, and events or transactions, as extracted in the multi-organization R&D program on Automated Content Extraction (ACE), see, for example [Doddington 2004], and in similar efforts such as Global Autonomous Language Exploitation (GALE). The information extraction achieved in these R&D programs is imperfect, but is regularly being improved by many researchers. Our object here is to investigate how we can use that extracted information to perform social network analysis and intent recognition, which we believe has significant potential for helping analysts to connect the dots in order to obtain early indications of terrorist threats. Hence the goal of our CT-SNAIR project is to develop and demonstrate the feasibility of automated tools to help analysts detect and track threat networks and their intent. In order to make this very difficult problem more manageable, we have generally made the assumption that, rather than having to simultaneously solve the "needle-in-a-haystack" problem and the connect-the-dots problem, we start with some initial clues -- for example we may start with knowledge of a key player or players, and build our social network around that set of people. We believe that significant progress on this more limited problem, in addition to being an important piece of the larger, massive data analysis problem, would be useful in and of itself. Another important aspect of our approach is that we are addressing modeling of network and attacks as a key part of the work, and that we are assuming that our SNAIR (Social Network Analysis and Intent Recognition) algorithms utilize these models in performing pattern recognition, as indicated in Figure 1. Our modeling effort will be described in more detail later in this paper. Our approach also considers SNA and IR to be joint, interdependent processes. The evolution of the network and the progress of the threat scenario are tightly coupled, and the modeling and recognition algorithms need to take this into account. Finally, we want to emphasize that the automated tools are intended to become an aid for the analyst, who would interact with the system to investigate and refine hypotheses. Hence the 2-way arrow between the users and the system in Figure 1. Study of this interaction is beyond the scope of this paper, but is a very important topic for future research.

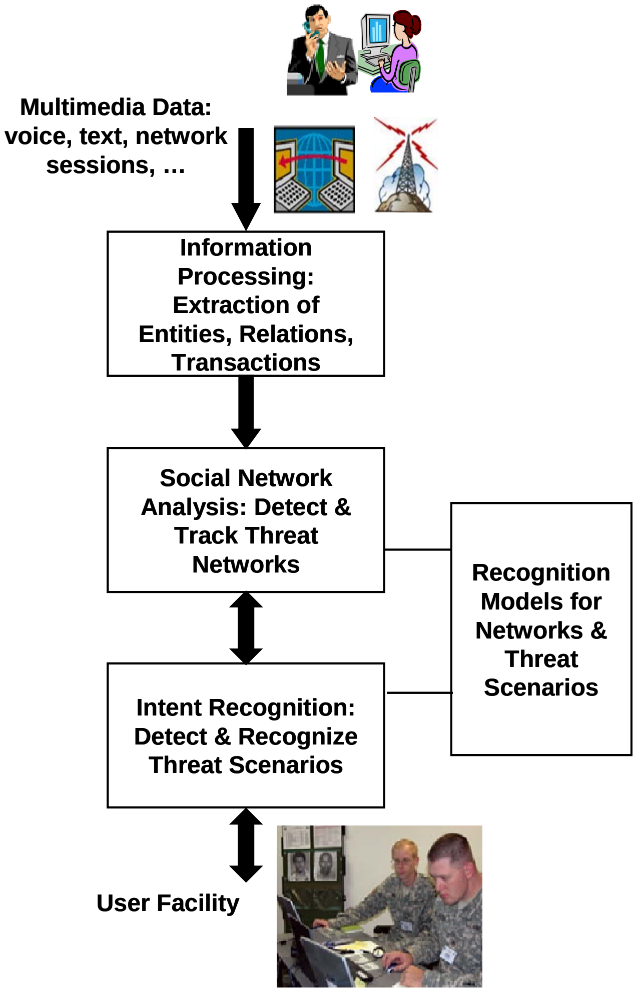

Multimedia Data: voice, text, network sessions, …
Information Processing: Extraction of Entities, Relations, Transactions
Social Network Analysis: Detect & Track Threat Networks
Recognition Models for Networks & Threat Scenarios
Intent Recognition: Detect & Recognize Threat Scenarios
User Facility

Figure 1: CT-SNAIR system framework.

3. Sketch of Example Scenario & Analysis

In order to develop our system approach to facilitate initial experiments, we found that it was very helpful to work on a sample, realistic terrorist scenario. For this purpose we selected the 2004 bombing of the Australian embassy in Jakarta, Indonesia [Jakarta 2004 Wikipedia]. A good deal of information about this attack, including names of the perpetrators and a reasonable timeline of events could be gleaned from open sources, including court records. However, specific transactions, such as activities of and communications among, the terrorists, were not available. So we constructed a series of hypothetical transactions consistent with the scenario, coded the transactions in TADL, and used the TADL together with our HMM-based simulator to conduct a series of experiments in SNAIR on the Jakarta scenario. A few of these hypothetical transactions are depicted in Figure 2. Note that the names of the individuals are real, but that the transactions are completely hypothetical. In keeping with the idea that, rather than dealing with a "needle-in-the-haystack" problem, we would have an initial pointer to key individuals, we might assume that "bad guy A" was a known target, and that our approach to SNAIR would be to build up a network around that tagged person, and to update the estimated probability that an attack was being prepared as the events unfold.

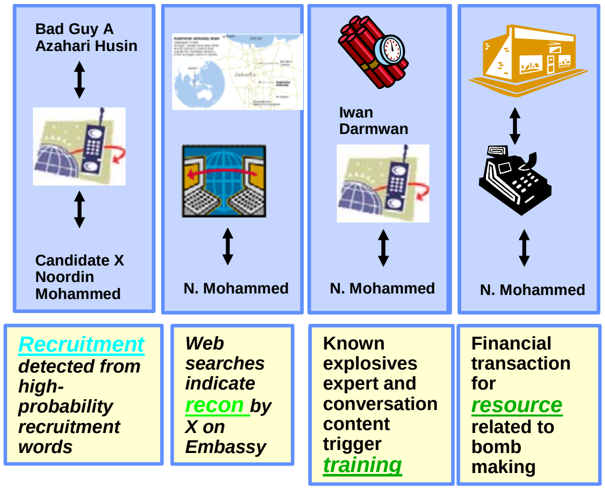

Bad Guy A Azahari Husin
Candidate X Noordin Mohammed
N. Mohammed
Iwan Darmwan
N. Mohammed
N. Mohammed
Recruitment detected from high-probability recruitment words
Web searches indicate recon by X on Embassy
Known explosives expert and conversation content trigger training
Financial transaction for resource related to bomb making

Figure 2: Hypothetical sequence of transactions for part of the Jakarta bombing scenario.

Our experiments with the Jakarta scenario are described in sections to follow, and they enabled us to establish our system framework and conduct initial tradeoff studies including investigation of the effects of various models for the clutter transactions which needed to be combined with the attack scenario transactions in order to measure both probability of detection and probability of false alarm.

4. Modeling & Simulation of Terrorist Networks & Attacks

Motivation for Modeling and Simulation

A significant component in statistical modeling of terrorist networks and attacks is understanding the *space* of possible attacks. Two potential sources of information are—prior attacks and hypothetical scenarios. For the former situation, which we refer to as *forensic scenarios*, we can putatively gather data and the corresponding transactions and form a time-ordered list of transactions for the scenario. For the latter situation, there exists no prior transaction data representing the entire process. Since we would like to handle both situations, it is natural to perform simulation to generate unseen scenarios.

Simulation of scenarios fulfills multiple goals. First, simulation provides a method of formally coding scenarios which are typically represented in non-precise terms. For instance, for typical Chemical, Biological, Radiological, Nuclear, and Explosive (CBRNE) scenarios such as those in [Howe04], there is a gross scenario overview, implications, etc., but no sequence of what critical events must occur to execute the scenario. Second, simulation provides a method for generating truth-marked data that can be used for statistical machine learning methods. This approach has been used in prior research for plan recognition, see [Blaylock05]. Third, simulation methods can be used to explore the robustness and limits of social network analysis techniques through parameter variation. Finally, simulation methods can act as a query process for an analyst. An analyst can dynamically construct scenarios simulations which have attributes and events which are of interest. The resulting simulations can be used for intent recognition and social network analysis.

M&S Plan and Framework

Our basic framework for modeling and simulation is shown in Figure 3. The overall flow is that modeling and simulation produces a data set of truth-marked transactions. These truth marked transactions are then used in a variety of ways—scenario model training, hypothetical studies, and social network analysis.

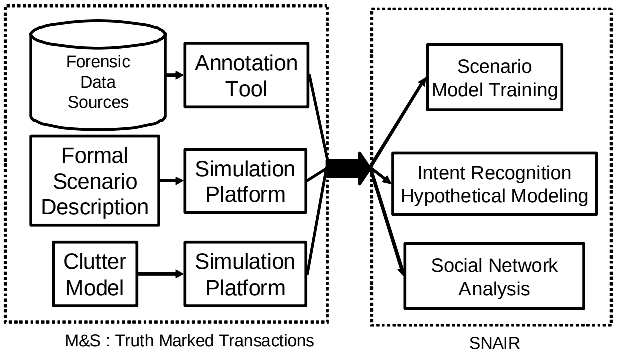
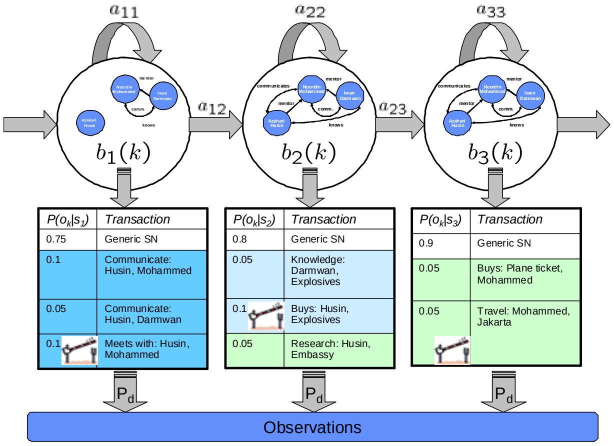


Figure 3: Modeling and Simulation Framework for CT-SNAIR

HMM-Based Simulation

The goal of simulation is to produce transactions representative of a scenario. The programming representation should provide enough variability to "span the space" of possible scenarios. Our initial approach is to use a language which stochastically generates transactions.

We base our simulation of scenarios upon discrete Hidden Markov Model (HMM) modeling, see, for example [Pattipati 2006]. For each state, we allow multiple different transaction types to occur (see Figure 4.) For instance, in state 1, there are four possible transactions. Each of these has an emission probability given that we are in the state. For instance, *p*(Meets: Husin, Darmawan| state=1) = 0.1. For each state, we allow for various directives such as "Generic SN." The "Generic SN" finds a random link in the current social network and produces a transaction describing that link.



$P(o_k\|s_1)$	Transaction
0.75	Generic SN
0.1	Communicate: Husin, Mohammed
0.05	Communicate: Husin, Darmwan
0.1	Meets with: Husin, Mohammed

$P(o_k\|s_2)$	Transaction
0.8	Generic SN
0.05	Knowledge: Darmwan, Explosives
0.1	Buys: Husin, Explosives
0.05	Research: Husin, Embassy

$P(o_k\|s_3)$	Transaction
0.9	Generic SN
0.05	Buys: Plane ticket, Mohammed
0.05	Travel: Mohammed, Jakarta

Figure 4: Hidden Markov Model for transaction generation

The overall state architecture controls the general flow of the scenario. We have constructed a tool which provides a method of graphically constructing the state transition matrix, p(state=j|state=i), to easily describe scenario execution paths.

Several other features have been included in our HMM simulation architecture. First, we allow for a change in the social network over time. The state circles shown in Figure 4 illustrate the graph of the social network. By altering this over time, the simulation allows actor roles to change. A second feature of our HMM simulation architecture is the introduction of gated transactions; see, for instance, the gate icon in state 1. A gated transaction requires that a certain transaction occur before proceeding to the next state. A third feature of our simulation is the use of an overall observation probability, P_d. This probability determines whether a transaction produced by the HMM is actually observed.

As a note, we mention that the HMM architecture has advantages and disadvantages for scenario description. The advantage of the HMM structure is that it is possible to easily control overall sequencing of a scenario events. Also, an HMM provides a rigorous theoretical structure for analysis and modeling. A drawback of HMM modeling is that it can be difficult to represent many phenomena that occur in scenarios. For instance, multiple tasks may be interleaved. Each of these tasks may be conveniently represented by an HMM, but combining tasks in a single HMM structure is more difficult.

Ontology for Entities, Relations, and Transactions

With the overall simulation theoretical framework in place, we can now describe some of the formal aspect of scenario coding. The first aspect that we consider is the formal coding of the transactions. We use an ontology for this process. We note that the main goal of our ontology is to have a standard representation for transactions, not to perform formal reasoning.

Ontologies are a common way of encoding a structure for knowledge representation [Brachman04]. Several common sources for transaction ontologies are available in the literature. First, the NIST-sponsored automatic content extraction (ACE) evaluations are one source [NIST08]. Another source is the social network literature. Examples of typical predicates in our ontology are shown in Table 1. Note that we use a many sorted (or typed) ontology. A full description of our ontology is beyond the scope of this paper.

Table 1: Examples of Predicates in an Ontology for SNAIR Simulation and Annotation

Predicate	Arguments	Meaning
Member	Per, Org	A person Per is a member of organization Org
Meets	Per1, Per2	A meeting has taken place between Per1 and Per2
SN_knows	Per1, Per2	Would recognize by name, face, speak to
SN_communicates_with	Per1, Per2	Talks to, writes

We impose some conditions on our ontology. First, we require that symbol grounding for the predicates in the system can be done automatically. That is, information content extraction and information retrieval methods can be used to extract assertions with a reasonable accuracy given input text. A second requirement for our ontology is that the predicates are built on the current tools being developed in the literature. Eventually it is expected that feedback between the information extraction community and SNAIR users will enrich this process.

Terror Attack Description Language (TADL)

A second aspect of our process of creating formal representations of scenarios is a description of the ordering and probabilities of transactions. As explained in a prior section, we have adopted initially an HMM model for activities. We code the HMM structure in a programming style language that we call *TADL*. The development of the TADL syntax is ongoing; here we present some elements of the language. A graphical interface is also being development to simplify the entry process.

The TADL interpreter at a top level reads in multiple components and simulates a scenario. The components read in include: a description of the HMM structure and states, a transaction ontology, a knowledge base, and an observation model.

The description of an HMM state and the transition matrix structure is straightforward. Since we are using a discrete HMM structure, we must specify the potential outputs per state and the related emission probabilities. This is done as follows:

```
state 5
xact 0.05 Buys(John,Ticket)
xact 0.60 Research(Bob,Monument)
xact-gate 0.30 Visit(John,Monument)
xact 0.05 SN
end_state
```

In the example, we've have three possible outputs indicated in lines 2-5; e.g., a possible output in state 5 is `Buys(John,Ticket)` with emission probability 0.05. The SN directive in line 6 indicates that a random social network relation should be emitted with probability 0.05. The `xact-gate` is a method of ensuring a transaction is emitted before transitioning to the next state; in this case we require that John visit the "Monument." For the HMM, we describe an arbitrary state transition structure using a sparse matrix format.

The remaining components in TADL are a knowledge base and an observation model. The knowledge base pre-defines symbols with the types in our ontology. The observation model encodes various parameters which describe the overall simulation. For instance, we have an overall probability which can gate whether a transaction occurs to simulate "drop-outs." We also specify a prior probability distribution for modeling clutter transactions.

Modeling & Simulation of Clutter Networks and Activities

In addition to modeling the target terrorist cell, we need to effectively model the background activities of both the target cell actors and an additional population of clutter actors. The target actors may interact with the clutter actors in various ways, and the clutter actors will interact amongst themselves. In order to provide variation across multiple training and test runs, the network should be randomly generated but with structures representative of real-world social networks.

As in the modeling of terrorist attacks, we wish to automatically generate sequences of transactions that are representative of the underlying network. Purely random graphs, such as the Erdős–Rényi model, have been shown not to adequately model real-world graphs. Many real-world graphs are thought to exhibit a scale-free property or power-law degree distribution. That is, a small number of nodes are highly connected while most nodes have small degree. While the underlying mechanisms of real-world graphs are not always understood, graphs such as computer networks, citation networks, protein interaction networks, and social influence networks have often been considered scale-free.

Many algorithms exist to randomly generate scale-free graphs. A comprehensive review of these algorithms is beyond the scope of this paper. In this work, we have chosen to use the R-MAT algorithm [Chakrabarti 2004] for its ease of implementation and good performance. This is an instance of a more general class of graphs known as Kronecker graphs. The R-MAT algorithm operates on an adjacency matrix which is recursively partitioned into random quadrants until no further partitioning is possible. At this stopping point, an edge is selected between nodes represented by the rows and columns of the matrix. This process continues until the desired number of edges is reached. The probability distributions of the four quadrants make up the set of parameters that determine some of the graph characteristics.

Using the R-MAT algorithm, we can generate many communities of background actors and embed the target community within them. In this way, the target actors interact with the background actors as would be expected in a real-world situation. Once the randomly generated social network is in place, we generate a TADL script for our simulator that will slowly reveal the format of the underlying graph. We do this by first selecting a link between actors at random and then generating a social network transaction according a probability distribution. At this time, it is difficult to know what the "true" distribution of social network transactions is supposed to be, but the parameterization allows the model to change as new data or theory becomes available.

Relation to prior M&S work

Prior work in modeling and simulation has had various foundations. First, a significant amount of modeling for counter-terrorism has been at the conceptual level. That is, scenarios are described in text form with methods of attack, implications, and defensive strategies, e.g. [Kumagi06]. This modeling is an excellent starting point, but does not provide a formal framework for simulation. Second, simulation of terrorist activities "in the large" for strategic purposes is common. These studies tend to focus on organization, disruption, and overall characteristics with agent simulation frameworks and/or game theory. Several works in this area are the Hats Simulator [Cohen04], Game theoretic results [Sandler08], and Dynamic Network Analysis [Carley07]. A third area of simulation looks at more detailed signatures of terrorist activities. The closest approach to ours in this area is [Pattipati06] which uses HMM models of terrorist activities. That work is described further in [Singh06]. We note that this prior work had significant influence on our current approach, but did not have the same focus as our efforts; i.e., Pattipati's approach is not focused on automatically extracted information from content, social network structure, and simulation/recognition frameworks.

5. Social Network Analysis Approach & Experiments

Given a set of time ordered transactions, we can construct a social network graph using the actors in the transactions. The different transaction types provide important information about the nature of relationships between individuals. This graph evolves over time as more transactions are provided. In this section, we will outline a series of experiments that highlight some of our social network analysis techniques. We concentrate mainly on static analysis of the graph, which could take place after a sufficient number of transactions have been received. First, we show that it is possible to construct a social network graph from analysis of unstructured text. Next, we describe a series of community detection experiments with terror networks embedded in clutter. Finally, we show some preliminary results of social network filtering using the synthetic terrorism scenario.

Figure 5 shows the overall process in generating the social network graph from transactions. A weighted multiplex graph is created where the link types are defined by the social network transaction types. Weights are provided by the content extraction algorithm as a confidence measure of the observation. For example, the natural language processing algorithm may be able to determine with high confidence that two people know each other but may be less confident about a business relationship between them. Each of these link types can be thought of as a separate graph which can be fed into social network algorithms in part or as a whole.

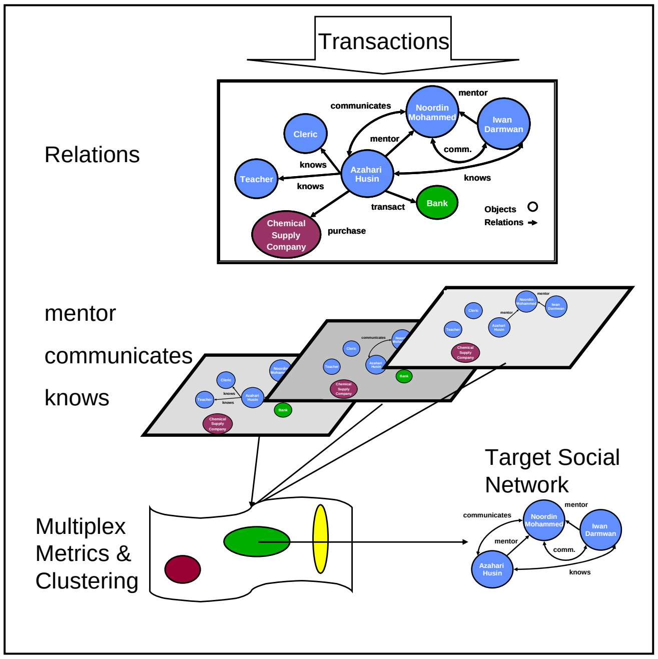


Figure 5: Multiplex social network graph generation

The relationship types between individuals ought to give important clues to the structure and nature of the network, but most social network algorithms do not handle these kinds of multiplex graphs directly. Our current implementation flattens the network into a single weighted graph, which allows us to use a variety of readily available SNA algorithms.

Of particular interest are community detection algorithms, where nodes of a graph are grouped into distinct communities. Among these is the Girvan-Newman algorithm [Girvan 2002] which divides the graph into communities by repeatedly removing edges with high edge betweenness, a measure of which edges lie on the shortest path between many nodes. The Girvan-Newman algorithm does not scale well to large graphs as the expensive edge betweenness metric must be recomputed after every edge removal. Faster algorithms based on the concept of network modularity have been introduced more recently [Newman 2006]. Modularity measures the quality of a selection of communities within a given network. We have found Carnegie-Mellon's Organizational Risk Analysis tool [Carley 2008] and the general graph analysis package iGraph [Csardi 2006] to be useful in our experiments. ORA provides a good interface for visualization with some capability to perform analysis, while iGraph contains a more complete set of algorithms that will work on a variety of different graph formats.

Experiments using Information Extraction from Text

Our basic hypothesis is that one can extract a reasonable social network from unstructured content using automated analysis tools. In this section, we explore the use of several different techniques for generating social networks from a corpus of text data. Related work includes the AutoMAP system [Diesner 2005], where link association is assumed by sentence co-occurrence. While this simple method yields reasonable results, AutoMAP relies on a manual process for entity detection and disambiguation and is not practical for use on large amounts of unseen data. More complex methods are presented in [McCallum 2007], where networks and individuals are simultaneously analyzed and clustered using topic discovery techniques. Our approach takes the middle ground of applying state-of-the-art entity and relation extraction and building social networks from the resulting structured content.

Our test corpus consists of 60 articles on a recent terrorist event, totaling about 200,000 words. A labeled social network was generated by a subject matter expert using only these 60 documents. This network will serve as "truth" for our analysis. The truth graph contains places, organizations, and events as well as individuals. For this study, we consider only person-to-person links and only the core individuals directly involved in the terrorist event. Our error metric will consist of precision and recall measures on the set of person-to-person links our algorithms return as compared to the truth graph. Precision is the number of hypothesized links that are correct divided by the total number of hypothesized links. Recall is the number of hypothesized links that are correct divided by the total number of truth links in the graph. Thus precision indicates how accurate the system is; while recall gives an indication of how much is missed.

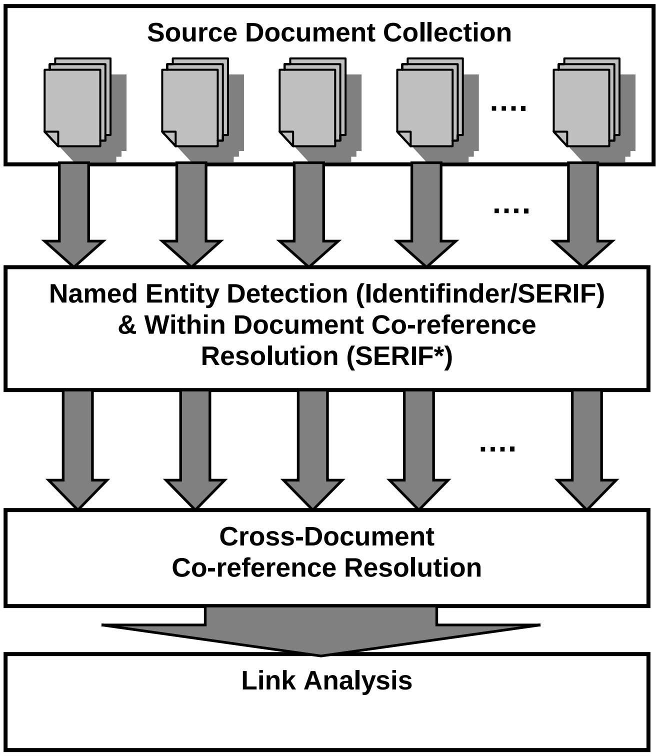

Source Document Collection
....
....
Named Entity Detection (Identifinder/SERIF)
& Within Document Co-reference
Resolution (SERIF*)
....
Cross-Document
Co-reference Resolution
Link Analysis

Figure 6: Block diagram describing derivation of social networks from text

The overall system used to generate social networks from text is shown in Figure 6. The collection of source documents is run through state-of-the-art natural language processing software to identify named entities and possibly events and relations. We used BBN Identifinder [Bikel 1999] to form a baseline for our system. Identifinder will tag named entities, including people, places, and organizations from a given source text. However, it will not perform within document co-reference resolution. Co-reference resolution involves the mapping of multiple expressions to the same real-world concept or entity. This can include pronoun references or descriptive phrases (e.g. the President of the U.S.A. and George Bush are the same person.) We used a more complex system known as SERIF (Statistical Entity & Relation Information Finding) also from BBN [Boschee 2005], to perform further analysis on the text, including within document co-reference and event and link extraction. The resulting set of entities then need to be resolved across documents in order to account for differences in spelling and naming conventions. In our original experiment with the Identifinder output, this was performed manually. For the SERIF output, this was accomplished via an automatic document clustering algorithm that uses as features: sentence context from each entity mention and a spelling distance measure on the core name. The resulting clusters indicate which entities are the same across documents. While the automatic cross-document co-reference worked well, we hand corrected a few entries in order to have a valid comparison against the manual resolution from the BBN Identifinder output.

In order to determine if a link exists between two people, we first consider sentence co-occurrence. If two people are mentioned in the same sentence, then it is likely that there is some link between then, although the negative is also possible. We also compare this simplistic technique with SERIF's relation finding algorithms, which reports both the evidence of a relation as well as the type as determined from the content.

The results from our experiment are shown in Table 2. In the first line, we use only Identifinder named entity tagging output with sentence co-occurrence to establish links. While precision is reasonable, the low recall suggests that many links are missing. Next, we use only the relations and events between persons given by the SERIF tool. The results indicate the SERIF returns a small set of more accurate links. Next, we apply sentence co-occurrence link detection to the SERIF output which gives the best overall performance. Some errors are made, as evident by the lower precision scores, but the recall increases dramatically. The addition of within document pronoun reference in SERIF is largely responsible for this gain over the Identifinder system. In the last line, we take the union of all links between the SERIF specific relations and the sentence co-occurrence relations. In this case, precision drops as several more incorrect links are added. While the SERIF system with sentence co-occurrence yields the best overall performance, the combination of the two provides useful content information such as the type of relation (family, business, etc.) between entities. Additionally, the high precision of SERIF allows us to infer certain relations with higher confidence. All of this information can be used to improve additional social network analysis, including community detection.

Table 2 – Results from social network analysis from text

Entity and Link Detection Method	Precision	Recall
Identifinder (Sentence Co-occurrence)	0.73	0.42
SERIF (Relations & Events)	0.77	0.32
SERIF (Sentence Co-occurrence)	0.70	0.64
SERIF (Relations, Events, & Sentence Co-occurrence)	0.67	0.64

Experiments using Simulated Attack and Clutter

In this section, we describe community detection experiments on simulated data from our Jakarta example. The simulated background clutter is generated according to the R-MAT algorithm and the number of nodes and edges are varied across runs. The simulated terrorist cell is embedded in the clutter network such that the terrorist actors communicate with the clutter actors but no new edges between the terrorist actors are introduced.

In our experiment, we performed community detection on the simulated graph and analyzed the results as follows. We search through the detected communities and select the one that has the highest number of terror cell actors, who, for the purposes of this experiment, are known in advance. We then count the number of terrorist and clutter actors in this community. Each clutter actor represents a false alarm, and each terrorist actor represents a positive detection. Given these counts, we can once again calculate precision and recall measures on the detected community. Because of the random nature of the graph generation process, we average the results across many runs.

The results of this experiment are shown in Table 3. We vary the number of nodes for the clutter network from 16 to 256 nodes and keep the edge count at twice the number of nodes. We calculate precision and recall as described above for both the Girvan-Newman and Newman modularity community detection algorithms. For smaller graphs the community detection performs quite well with high precision scores. However, as the graph gets larger, precision scores begin to drop dramatically. In the Girvan-Newman algorithm, recall remains about the same regardless of graph size. This indicates a fixed percentage of missed terrorist actor nodes. The Newman modularity algorithm shows a higher recall measure in general, which increases with the number of nodes. This higher recall is advantageous for our application as a filtering mechanism to pare down the community size before intent recognition. In this case it is probably better to include more clutter nodes (lower precision) rather than miss terrorist actor nodes (low recall). However, we are still far from perfect performance, indicating that more research is needed in this area.

Table 3 – Community Detection of Target Group in Random Clutter

Clutter Parameters		**Group Detection Metrics**			
		Girvan-Newman Betweenness		**Newman Modularity**	
Nodes	*Edges*	*Precision*	*Recall*	*Precision*	*Recall*
16	32	0.93	0.69	0.67	0.79
32	64	0.86	0.65	0.67	0.83
64	128	0.79	0.64	0.58	0.89
128	256	0.53	0.66	0.48	0.96
256	512	0.23	0.64	0.35	0.97

Experiments with Ali Baba Data

The final experiment involves the synthetic Ali Baba data. The Ali Baba data set has been used by other researchers, see e.g., [Gerdes07]. For the Ali Baba data set, we used scenario 1 which consists of 800 synthesized documents that replicate intelligence reports of suspected terrorist activity in southern England. The documents are labeled as either being part of the scenario or as clutter. We created discrete transactions for each synthetic document using an in-house annotation tool. We use these hand-annotated transactions to build a multiplex social network and perform community detection using the Girvan-Newman algorithm. The transactions are then filtered into overlapping "buckets" of transactions based on the detected communities. For each community, we select all transactions that contain at least one of the individuals in that community. Given that we know the members of the target cell in the Ali Baba scenario, we can calculate the percentage of "truth" transactions present in each bucket of transactions. In this case, the truth set of transactions contains all transactions where at least one of the target cell actors is present in the transaction.

Table 4 – Community Detection Results for Ali Baba Data

Community ID	**Percentage of Target Transactions Covered**
1	82.5%
2	15.9%
3	0.9%
4	0.6%
5	12.9%
6	2.4%
7	5.4%
8	0.6%
9	0.9%
10	1.8%

The total Ali Baba data set consists of 785 total actors with 25 of them belonging to the core target cell (including aliases, etc.) There are 2385 total transactions, with approximately 337 of these belonging to the "truth" set of target cell actors. In this experiment, we use a weighted sum of multiplex relations to create a single weighted link between two actors. This allows weak relations such as `Is_Aware_Of` to have less effect on the final set of communities while strong relations have greater effect. The weights were hand tuned but could be determined automatically on a held-out social network. After running the community detection algorithm on this weighted network, there were 89 detected communities. Most of these communities were isolated from the rest of the graph and only contain two to three actors.

The results of the transaction filtering based on community detection are shown in Table 4 for the 10 largest detected communities. The first community contains the largest percentage of target cell actors, and therefore the largest percentage of target transactions. There are two other communities that contain more than 10% of target transactions, but we would expect the intent recognition algorithm to reject these as the set of transactions would contain mostly clutter.

6. Intent Recognition (IR)

This section describes our work on intent recognition, in which we focus on detection of target scenarios in a transaction stream.

Problem Definition and System Framework

Intent recognition is part of the overall system framework presented in Section 2. The goal of intent recognition is to provide an indication to an analyst of which threats may be present in a transaction stream. The exact process of intent recognition could potentially involve several tasks—*detection* of known or hypothetical target scenarios, *prioritization* of target scenarios, and *interpretation* of the resulting detection.

In this work, we focus on *detection* of target scenarios. We assume that we have a known target scenario. The transaction stream is observed for some percentage of the entire scenario; e.g., we observe 25% of the transactions. Typically, this 25% would be from the beginning of the scenario. We then have a target scenario detector which produces a score that is compared to a threshold, and a scenario-present or clutter-only decision is produced.

Detection of terrorist scenarios is implemented using a standard statistical train/test in our methodology. Our basic setup using a Support Vector Machine (SVM) based detector is shown in Figure 7. In the figure, background clutter and the scenario are generated using the TADL simulation we described in Section 4. We interleave clutter and scenario, represented by C and S, respectively, producing a combined transaction stream. We then truncate the output to some percentage of the scenario. Finally, this stream is introduced into the SVM detector which has SVM scenario models.

Training of a target scenario detector is straightforward with the setup in Figure 7. To produce true trials, we turn off clutter and then run the TADL simulation multiple times. Since the TADL interpreter is stochastic, multiple distinct outputs will be produced. To produce false trials, we turn off the scenario model and use only clutter. Once we have false trials and true trials, we can train a detector to produce a likelihood ratio or a posterior probability, *p(*scenario|xact). The features and classifier structure are, of course, the challenge, and we describe a preliminary approach in the next section.

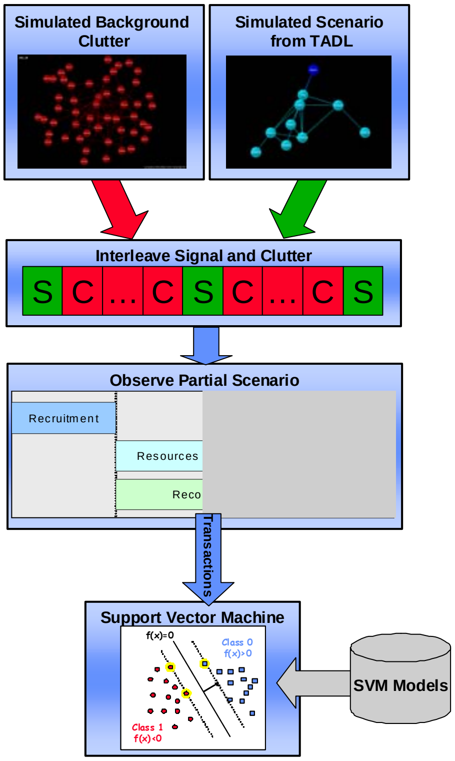


Figure 7: Setup for Scenario Training & Recognition using Simulated Data

As a performance criterion for the detection task, we propose the use of standard ROC or detection error tradeoff (DET) curves. These curves illustrate tradeoffs between probability of false alarm and probability of miss. Although we do not expect the performance criterion to produce an absolute measure of effectiveness of an intent recognition system, it should show relative gains for different system implementations and also provide a measure for contrastive experiments.

SVM Techniques for Training and Recognition

As a recognition system for detection of scenarios, we selected a support vector machine. An SVM, *f(x)*, [Cristianini00] is a two-class classifier constructed from a sums of a kernel function *K(x,y)*,

$$f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) + b$$

where the x_i are support vectors from the training set, and α_i and *b* are trained parameters. For classification, a class decision is based upon whether the value, *f(x)*, is above or below a threshold.

The choice of an SVM for recognition is based upon multiple considerations. First, at a top level, the use of the simulation models for scenario and clutter should be optimal for recognition. But, using these models for recognition will not create a robust detection system. For instance, in real situations, scenarios can be reordered subject to their dependencies. Using the generation model to detect a rearranged scenario in this case will result in a low detector score and probably a miss by the detector. Therefore, separating the detection framework from the simulation framework is critical in our modeling. Other reasons for choosing an SVM are it's flexibility in incorporating multiple feature types, good detector performance, and a well-developed tool set.

We consider two types of modeling for the SVM. First, we used bag-of-events (BOE) modeling. The features in this situation are the counts of *n*-grams of events in a transaction stream. By event, we mean the predicate name only and not its arguments. For instance, for a predicate *Meets(Bob,John)*, we only record only the fact that *Meets* takes place and not the specific actors. The second type of SVM modeling uses bag-of-events and bag-of-arguments combined together (*SVM BOEA*). In this case, since we are using a typed ontology, we only use *n*-grams of types that cover a general scenario. For instance, in our experiments we do not include names of specific people in our *n*-gram representation, since they could be arbitrarily renamed. An example in this case, is that if the predicate was *Recon(Bob,White House)*, then the output *n*-grams (for unigram events and bigram arguments) would be *Recon_White, Recon_White_House, Recon_House*.

The SVM kernel for both approaches is based upon a linearized likelihood ratio kernel presented in [Campbell07]. Note that in both types of SVM models, we only have partial information about the sequence ordering because of *n*-grams. This technique ensures some robustness to scenario reordering.

The difficulty, in general, with designing classifier features based on predicates and their arguments is that the representation space is large and has both discrete and continuous aspects (e.g., names of individuals, ages of individuals). Also, there is a tradeoff between features representing specific versus generic aspects of an entity. For instance, naming specific terrorist targets such as buildings or people may be of interest in detecting some scenarios. In other scenarios, we may be looking only for generic aspects of targets—nationality, ownership, infrastructure role, etc. These issues are certainly a topic of future research.

Experiments

In our first set of experiments, we constructed a scenario from the Jakarta Embassy bombing that occurred on Sept. 9, 2004. The basic outline of events was taken from approximately 50 open source new articles. We found that a reasonable timeline of events could be gleaned from open sources, but that specific transaction were difficult to document. In some cases, details from court records highlighted in articles provided interesting insight. From these open sources, we constructed a series of hypothetical transactions consistent with the scenario and coded a simulation model in TADL.

Experiments were performed using the setup in Figure 7. For our first set of experiments, we generated scenario transactions using a uniform prior for clutter transactions. We generated simulated training and test data using the TADL interpreter. An SVM with a unigram BOE model was used as a detector. Initially, we considered various percentages of observation of the scenario and interleaving. We swept the percentage of the scenario observed, *P*, from 0 to 100%. We also swept the duty cycle *D*—the percentage of scenario transactions to clutter (as shown in the interleaving in Figure 7)—up to 25%. Note that the interleaving is done randomly. The equal error rate (EER, Pmiss=Pfa) as a function of these parameters for a clutter network of 1000 actors is shown in Figure 8. Several observations can be made from this figure. First, for *D* and *P* greater than about 20% we are getting good performance. Second, duty cycle appears to be the more significant parameter in the simulation process.

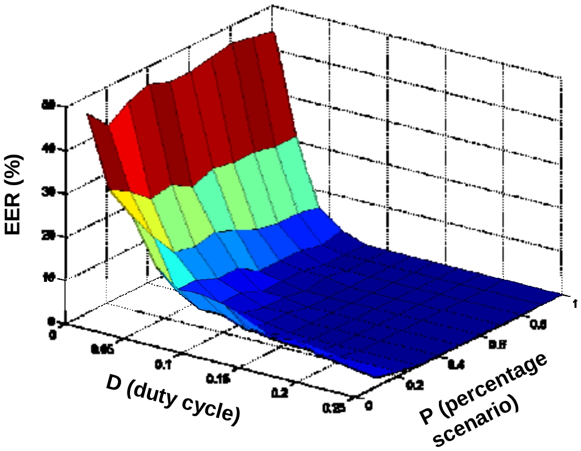


Figure 8: EER performance of the Jakarta SVM Scenario Detector with varying parameters

As part of our experiments, we also tested the effect of the prior probability distribution for the clutter model on recognition performance. We tried to match the prior for clutter transactions closer to the actual scenario. As expected, this created a more challenging detection task. An ROC curve comparing uniform clutter and more challenging clutter is shown in Figure 9.

We remark at this point that the simulations for the Jakarta scenario were a proof of concept for intent recognition.

One difficulty with a known scenario is that it is essentially "linear"; i.e., there is very little variation in the ordering of events. Also, the data simulation is known and matches the experimenters' pre-conceived concept of the problem; i.e., there is no red-teaming in the process.

As a next step, we considered other sources of data. We found that the Ali Baba simulated data set provided an interesting second way of testing our system. As in our social network analysis experiments, we use the synthetic documents from scenario 1 of the Ali Baba data set, which includes labeling of scenario vs. clutter for each transaction.

For our experiments, we formed two teams. One team took the Ali Baba documents and hand-annotated transactions using the TADL ontology. Another team constructed a TADL simulation based only upon a high-level overview of the scenario. The goal in this case was to see if a scenario could be detected given only a minimal top level description.

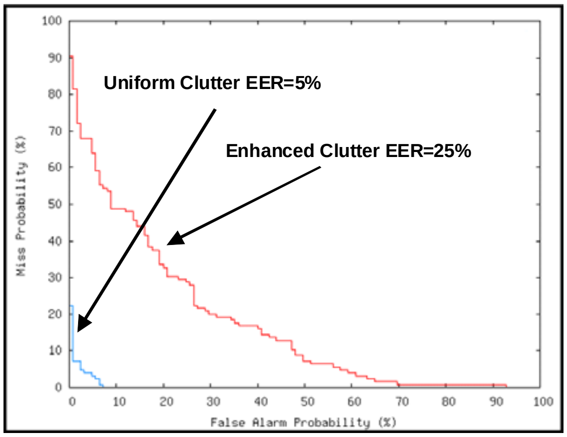


Figure 9: ROC Comparison for the Jakarta Simulation with Different Clutter Generation Models. *P=0.6. D=0.1*

Our experiments used the TADL simulator to generate data and train an SVM scenario detector as in Figure 7. Initially, we constructed a clutter model based upon event priors from the Ali Baba data set. For the SVM, we used a BOEA model where the events were unigrams and the arguments were all *n*-grams up to 4. Person arguments were excluded from the BOEA modeling.

Results for the detection using various subsets of the annotated document sequence are shown in Figure 10. Note that the Ali Baba data set has only one scenario instance, so in order to generate multiple trials we had to Monte Carlo sample the scenario. The sampling was always performed with a time window covering the required duration (e.g., 25%). If clutter only was desired, then the terrorist scenario documents were removed from the sequence. From the figure, we note that as we observe more of the scenario the EER drops. With about 85% observation of the scenario, the detection performance is good.

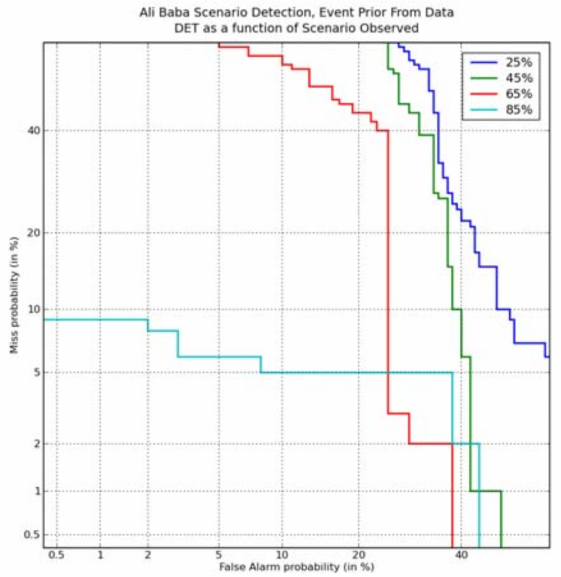


Figure 10: Ali Baba Scenario Detection with a simulated event prior

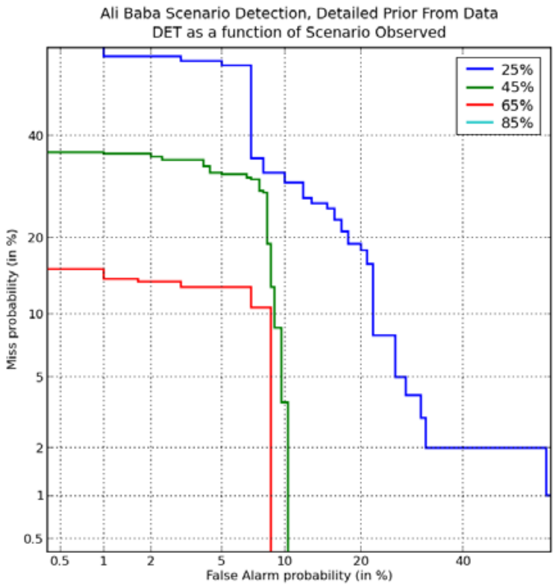


Figure 11: Ali Baba Scenario Detection with clutter sampled from the Ali Baba data set

We performed another set of experiments where we sampled a small amount of the clutter (about 5%) and then used this as training data for the false class for the SVM. Training data for the true trials was done using the TADL simulated output. The results of this set of experiments are shown in Figure 11. We see that the detection performance improves dramatically.

The Ali Baba and Jakarta experiments illustrate several points. First, modeling of the prior for clutter should be a key part of the detection process. Theoretically, if we have a large data source, the parameters of the clutter model can be learned automatically. The example of using samples of the Ali Baba data illustrates if we have an "oracle" prior then good detection is possible (compare Figure 10 and Figure 11). Second, our experiments illustrate that the process of simulation and recognition can be decoupled. The scenario detector does not have to be of the same complexity as the simulation. In fact, a simpler detector may be more robust. A third point illustrated by our experiments is that significant work is needed in features for simulation and recognition. Our BOEA model uses very specific features that may not generalize well. One way to think of the process is as a query task. We would like to give TADL examples that produce detectors that generalize to unseen situations.

Relation to prior work

Our work in intent recognition is related to prior work in plan or goal recognition. We have highlighted some references for this area in Section 4. Our current approach is statistical and is thus distinct from classical plan recognition. Our general methodology is comparable to the ASAM system [Pattipati06], but our specific detection strategy using an SVM is distinct from the prior work in this area.

7. Ongoing and Future Work

LL-SNAIR Corpus Development

One of the main difficulties we found when first considering this work is the lack of truth-marked data for development. Ideally, having raw data and markings such as a social network structure, roles, and events and clutter in a scenario is critical to benchmarking and developing algorithms. Although some data is available, it is usually incomplete; for instance, a social network may be available, but the documents it was derived from are not. One goal for future work is to develop truth-marked corpora for algorithm development and validation of technologies. We plan to create corpora using open source scenarios and documents with a mix of simulated and real data.

Future R&D Directions

The area of social network analysis with respect to intent recognition contains many opportunities for additional research. In particular, we will consider community detection algorithms that allow actors to be members of more than one group as well as provide probabilistic features of group membership for intent recognition. This should help avoid the pitfalls of making hard decisions early in the pattern recognition process. Dynamic features of social networks, particularly those which can aid in intent recognition, represent additional research opportunities.

In the area of intent recognition, there are many areas of possible future research. For simulation, we plan to create more sophisticated scenarios and tools to address the issue of partial ordering of events. For recognition, we want to combine social network analysis, entity attributes, etc. into the intent recognition process to improve generalization.

8. Summary and Conclusions

This paper introduced a framework of social network analysis and intent recognition from multimedia input. We demonstrated our framework in the context of multiple sources of data—actual text documents, simulated threat scenarios, and the Ali Baba data set. Also, we showed there is a significant need for corpora with truth-marked social network structure and threat scenarios. Overall, further research on this area should focus on building both data sets (real and simulated) and algorithms for recognition to understand the challenges and benchmark performance.

Acknowledgements

Our work is greatly indebted to the guidance of Zachary Lemnios of MIT Lincoln Laboratory and Robert Popp of National Security Innovations. Both of these individuals provided key inputs and ideas to guide our research process.

References

[Allanach 2004] J. Allanach, H. Tu, S. Singh, K. Pattipati and P. Willett, "Detecting, Tracking and Counteracting Terrorist Networks via Hidden Markov Models," IEEE Aerospace Conference, Big Sky, MT, March 2004.

[Blaylock05] N. Blaylock and James Allen, "Generating artificial corpora for plan recognition," in Liliana Ardissono, Paul Brna, and Antonija Mitrovic, editors, User Modeling 2005, number 3538 in Lecture Notes in Artificial Intelligence, pages 179-188. Springer, Edinburgh, July 24-29 2005.

[Bikel 1999] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. 1999. An algorithm that learns what's in a name. Machine Learning, vol. 34, no. 1-3.

[Boschee 2005] E. Boschee, R. Weischedel and A. Zamanian, Automatic Information Extraction, Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, VA, 2-4 May 2005.

[Brachman04] R. J. Brachman and H. J. Levesque, Knowledge Representation and Reasoning, Morgan Kaufmann Publishers, 2004.

[Campbell07] W. M. Campbell, J. P. Campbell, T. P. Gleason, D. A. Reynolds, and W. Shen, "Speaker Verification using Support Vector Machines and High-Level Features," IEEE Trans. Audio, Speech and Language Processing, Sept. 2007, vol. 15, no. 7, pp. 2085-2094.

[Carley 2007] Carley, Kathleen. “Destabilizing Terrorist Networks, “Proceedings of the 8th International Command and Control Research and Technology Symposium, conference held at the National Defense War College, Washington DC. Evidence Based Research, Track 3, Electronic Publication.

[Carley 2008] Carley, Kathleen & Columbus, Dave & DeReno, Matt & Reminga, Jeffrey & Moon, Il-Chul. (2008). ORA User's Guide 2008. Carnegie Mellon University, School of Computer Science, Institute for Software Research, Technical Report CMU-ISR-08-125

[Chakrabarti 2004] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-mat: A recursive model for graph mining. In SDM, 2004.

[Coffman 2004] Thayne R. Coffman, Sherry E. Marcus, Dynamic Classification of Groups Through Social Network Analysis and HMMs, 2004 IEEE Aerospace Conference Proceedings.

[Cohen04] P. R. Cohen and C. T. Morrison, “The HATS simulator,” Proc. of the 2004 Winter Simulation Conference, 2004.

[Csardi 2006] G Csardi, T Nepusz, The igraph software package for complex network research, - InterJournal Complex Systems, 2006.

[Diesner 2005] Diesner, Jana & Carley, Kathleen. (2005). Exploration of Communication Networks from the Enron Email Corpus. Proceedings of the Workshop on Link Analysis, Counterterrorism and Security, SIAM International Conference on Data Mining 2005, pp. 3-14. Newport Beach, CA, April 21-23, 2005., 3-14.

[Doddington 2004] G. Doddington, A. Mitchell, M. Pryzbocki, L. Ramshaw, S. Strassel, R. Weischedel, The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation, Proceeding of LREC 2004 Conference on Language resources and Evaluation

[Gerdes07] D. A. Gerdes, C. Glymour, and J. Ramsey, “Who’s Calling? Deriving Organization Structure from Communication Records,” in Information Warfare and Organizational Decision Making, ed. Alexander Kott, 2007.

[Girvan 2002] Girvan M. and Newman M. E. J., Proc. Natl. Acad. Sci. USA 99, 7821-7826 (2002)

[Howe04] Howe D., “Planning Scenarios: Executive Summaries,” The Homeland Security Council, 2004.

[Jakarta 2004 Wikipedia], “2004 Australian Embassy Bombing,”http://en.wikipedia.org/wiki/2004_Australian_embassy_bombing.”

[Jensen 2007] Proximity 4.3 QGraph Guide, November 2007, http://kdl.cs.umass.edu/proximity/documentation/QGraphGuide.pdf.

[Kumagi06] J. Kumagi, ed., “Nine Cautionary Tales,” IEEE Spectrum, Sept. 2006, pp. 36-45.

[McCallum 2007] Joint Group and Topic Discovery from Relations and Text. Andrew McCallum, Xuerui Wang and Natasha Mohanty, Statistical Network Analysis: Models, Issues and New Directions, Lecture Notes in Computer Science 4503, pp. 28-44, (Book chapter), 2007

[Neville 2007] Neville, J. and D. Jensen, Relational Dependency Networks. Journal of Machine Learning Research. 8 (March, 2007): 653-692. http://jmlr.csail.mit.edu/papers/volume8/neville07a/neville07a.pdf

[Newman 2006] M. E. J. Newman (2006). "Modularity and community structure in networks". Proc. Natl. Acad. Sci. USA 103: 8577–8582

[NIST08] “Automatic Content Extraction: 2008 Evaluation Plan,” http://www.nist.gov/speech/tests/ace.

[Olive 2008] J. Olive, Global Autonomous Language Exploitation (GALE), Program description, http://www.darpa.mil/ipto/programs/gale/gale.asp

[Popp 2006] R. Popp and J. Yen (editors): Emergent Information Technologies and Enabling Policies for Counter-Terrorism (in this comprehensive reference see especially chapter 2, Hidden Markov Models and Bayesian Networks for Counter-Terrorism by K. Pattipati, et. al.), IEEE Press, 2006

[Popp 2005] R. Popp, K. Pattipati, P. Willett, D. Serfaty, W. Stacy, K. Carley, J. Allanach, H. Tu and S. Singh, “Collaborative Tools for Counter-Terrorism Analysis,” IEEE Aerospace Conference, Big Sky, MT, March 2005.

[Pattipati 2006] K. R. Pattipati, P.K. Willett, J. Allanach, H. Tu and S. Singh, “Hidden Markov Models and Bayesian Networks for Counter-terrorism,” R. Popp and J. Yen (editors) Emergent Information Technologies and Enabling Policies for Counter Terrorism, Wiley-IEEE Press, May 2006, pp. 27-50.

[Sandler08] Todd Sandler and Kevin Siqueira, “Games and Terrorism: Recent Developments,” Simulation & Gaming, Sep 2003; vol. 34: pp. 319 - 337.

[Singh06] S. Singh, W. Donat, J. Lu, K. Pattipati, and P. Willett, “An Advanced System for Modeling Asymmetric

Threats," IEEE International Conference on Systems, Man, and Cybernetics, October 2006.

BIOGRAPHY

Clifford J. Weinstein *leads the Information Systems Technology Group at MIT Lincoln Laboratory and is responsible for initiating and managing research programs in speech technology, machine translation, and information assurance. He received S.B., S.M., and Ph.D. degrees in electrical engineering from MIT. He has made technical contributions and carried out leadership roles in research programs in speech recognition, speech coding, machine translation, speech enhancement, packet speech communications, information system assurance and survivability, integrated voice/data communication networks, digital signal processing, and radar signal processing. In 1993, Dr. Weinstein was elected to the Board of Governors of the IEEE Signal Processing Society. From 1991-93, he was chairman of the IEEE Signal Processing Society's Technical Committee on Speech Processing. In 1976-78, he was chairman of that Society's Technical Committee on Digital Signal Processing. In 1993, Dr. Weinstein was elected as a Fellow of the IEEE for technical leadership in speech recognition, packet speech, and integrated voice/data networks. From 1986-1998, Dr. Weinstein was U.S. technical specialist on the NATO RSG10 (now IST-01) Speech Research Group, in which capacity he authored a comprehensive NATO report and journal article on opportunities for applications of advanced speech technology in military systems. From 1989-1994, he was chairman of the coordinating committee for the DARPA Spoken Language Systems Program, which was the major U.S. research program in speech recognition and understanding, and which involved coordinated efforts of a number of leading U.S. research groups. From 1999-2003, he served on the DARPA Information Sciences and Technology (ISAT) Panel, a group which provides DARPA with continuing assessments of the state of advanced information science and technology, and its relationship to DoD issues.*

William M. Campbell *received the B.S. degrees in Electrical Engineering, Computer Science, and Mathematics from the South Dakota School of Mines and Technology in 1990, the M.S. degree in Applied Mathematics from Cornell University in 1993, and the Ph.D. degree in Applied Mathematics in 1995 from Cornell University (under a NSF fellowship). From 1995 to 1999, he was a senior research scientist at the Motorola Space and Systems Technology Group (SSTG) in the Speech and Signal Processing Lab. While at Motorola SSTG, he did research in Biometrics, speech interfaces for wearable computing, and communications for the battlefield. He participated in the creation of numerous products including the Tenor Pager, the CipherVox speaker verification SDK, the Force XXI wearable computer voice control, and the Rome Labs adaptive-rate voice communication system. From 1999 to 2002, he was a principal research scientist in Motorola Labs where he worked on machine learning, biometrics, and telematics. Since 2002, he has been a staff member of the Information Systems Technology Group at Lincoln Laboratory, where he is involved in digital signal processing and machine learning for speech applications. He has contributed numerous publications to journals and publications. He has received the Motorola Distinguished Innovator award and holds 12 patents. He is a member of the IEEE, American Mathematical Society, Tau Beta Pi, and Eta Kappa Nu.*

Brian Delaney *received a B.S. degree in Computer Engineering in 1997, an M.S. degree in Electrical Engineering in 1999, followed by a Ph.D. degree in Electrical Engineering in 2004, all from the Georgia Institute of Technology. His thesis work concentrated on distributed speech recognition on a wearable SmartBadge computing device from Hewlett-Packard Laboratories, including issues and tradeoffs concerned with energy consumption and the effect of bit errors on speech recognition accuracy. During his graduate studies, he acted as a consultant to Vocalocity, Inc., a provider of telephone based voice response systems based on the VoiceXML standard. He has been a member of the Technical Staff at MIT Lincoln Laboratory since 2004 where he has worked on statistical methods for machine translation of speech input as well as more recent efforts in social network analysis. He is a member of the IEEE and has published numerous conference papers.*

Gerald C. O'Leary *received S.B and S.M. degrees in Electrical Engineering and the Electrical Engineer degree from M.I.T in 1964 and 1966. From 1966 to 1971, he worked at MITRE Corp. in the Advanced Radar Techniques Department. From 1971 to 1977, he worked for Signal Processing Systems, Inc. in the area of programmable digital processors for communications applications. In 1977 he joined the Technical Staff of M.I.T. Lincoln Laboratory. There he has served as Associate Leader of the Information Systems Technology Group from 1984 to 1998 and of the Tactical Communications Systems Group from 1998 to 2000. He has worked in the areas of satellite communications, speech processing, digital networking and information security. He is currently a Senior Staff in the Information Systems Technology Group. He is a member of the IEEE.*