Storytelling in Entity Networks to Support Intelligence Analysts

M. Shahriar Hossain[1], Patrick Butler[1], Arnold P. Boedihardjo[2], Naren Ramakrishnan[1]
[1]Virginia Tech, Blacksburg, VA 24061
[2]U. S. Army Corps of Engineers, Alexandria , VA 22315
Email: {msh, pabutler}@cs.vt.edu, arnold.p.boedihardjo@usace.army.mil, naren@cs.vt.edu

ABSTRACT

Intelligence analysts grapple with many challenges, chief among them the need for software support in storytelling, i.e., automatically 'connecting the dots' between disparate entities (e.g., people, organizations) in an effort to form hypotheses and suggest non-obvious relationships. We present a system to automatically construct stories in entity networks that can help form directed chains of relationships, with support for co-referencing, evidence marshaling, and imposing syntactic constraints on the story generation process. A novel optimization technique based on concept lattice mining enables us to rapidly construct stories in massive datasets. Using several public domain datasets, we illustrate how our approach overcomes many limitations of current systems and enables the analyst to efficiently narrow down to hypotheses of interest and reason about alternative explanations.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Data mining*; H.3.3 [**Information storage and retrieval**]: Information search and retrieval—*Search process, Retrieval models*

1 Introduction

Intelligence analysts today are faced with many challenges, chief among them being the need to fuse disparate streams of data, and rapidly arrive at analytical decisions and quantitative predictions for use by policy makers. Although there are available catalogs of problem solving strategies suitable for intelligence analysis (e.g., see [1, 2]) and several visual analytic tools (e.g. [3–8]), our understanding of underlying user needs is constantly evolving.

After a thorough user study, Kang and Stasko [9] suggested several design implications for systems supporting intelligence analysis. Along with several suggestions related to information management, two of the suggested design requirements are: (1) help analysts create a convincing production by supporting insight provenance and sanity checks, and (2) help analysts continuously build a conceptual model. In our work, we design an algorithmic framework to help intelligence analysts make connections by providing evidence and explanations to build a mental model.

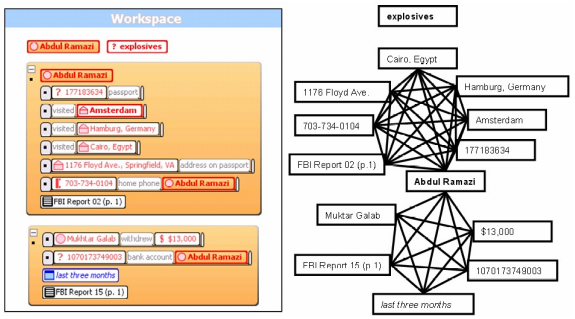
Here, we focus on the task of storytelling [10], i.e., connecting the dots between disparate entities in an attempt to discover relationships between hitherto different concepts and suggest hypotheses. For example, what is the connection between Osama Bin Laden and Igor Kolokov? What other entities relate them and what documents provide evidence for the underlying connections?

KDD'12, August 12–16, 2012, Beijing, China.

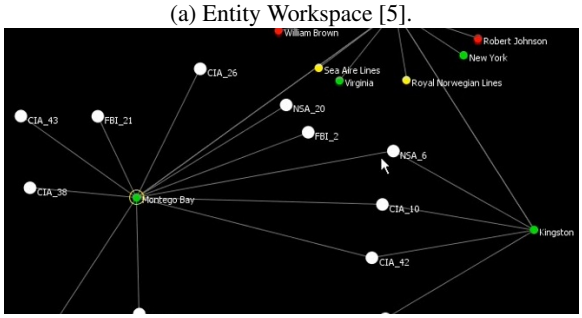
1.1 Limitations of Current Tools

Current tools in this space—e.g., Entity Workspace [5], Jigsaw [8], NetLens [11], and Sentinel Visualizer [4]. Entity Workspace—use entity recognizers to infer a graph of relationships between entities (see Fig. 1). (There are other tools that we do not survey here, e.g., Palantir, but their characteristics are similar.) Entity Workspace focuses more on information representation goals and does not provide graph exploration or summarization capabilities, whereas Jigsaw, NetLens, and Sentinel Visualizer provide exploration techniques with different emphases. In Jigsaw and NetLens, the user can traverse from entities to documents (in which the entities appear), and vice versa. Sentinel Visualizer supports exploration of the entity network and can hide document level details by using concepts of degree centrality, betweenness centrality, and closeness. In all cases, however, manual exploration becomes difficult as networks get dense with increasing numbers of entities. More broadly, significant shortcomings of the above tools are:

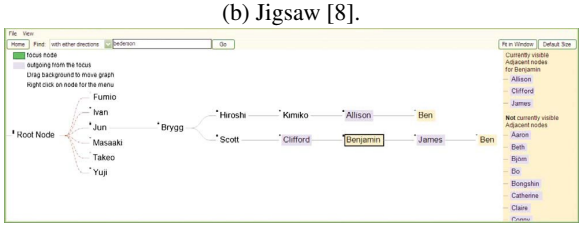
A. *Lack of support for "evidence marshaling"*: The tools described above employ entity network models where the edges represent only boolean associations between entities, i.e., presence/absence in the same document(s). They require that the documents be manually read by analysts to marshal evidence.
B. *Lack of support for explanation:* Extracted and inferred relationships between entities should be explained by surrounding evidence. With the above tools, the user is relegated the task of explaining the visualized relationships. Because multiple explanations can exist for a relationship, the analyst is further burdened with the manual task of sorting through these explanations.
C. *Directed search:* Current tools do not support automated directed search to discover relationships between user-specified entities. Manual exploration often results in failed searches that loop back to the origin instead of leading towards a destination entity.
D. *Syntactic constraints:* Analysts bring in significant domain knowledge that manifest as syntactic constraints on path finding. In current systems, users must manually apply constraints at each step of the exploration which significantly prolongs the discovery process.
E. *Lack of support for entity disambiguation:* A person named "Igor Kolokov" might be referred to in different ways across a document (e.g., "Igor Kolokov" can be variously extracted as "Igor Kolokov", "Kolokov", and "Igor"). In current visual analytics tools for intelligence analysis, entity disambiguation is a manual process left to the user. Further, current



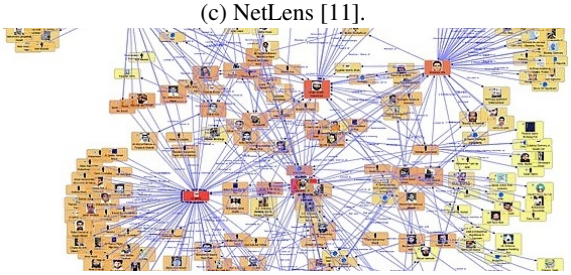
(a) Entity Workspace [5].



(b) Jigsaw [8].



(c) NetLens [11].



(d) Sentinel Visualizer [4].

Figure 1: State-of-the-art visual analytic tools for intelligence analysis.

entity recognition tools will be unable to extract an entity from a sentence if the sentence contains pronominal references (e.g., "that guy" instead of "Kolokov").

F. *The curse of large datasets and massive networks:* All the above tools require that a pre-generated or fully materialized network of all relationships be computed prior to exploration. As a result, exploration of large networks becomes computationally taxing and practically infeasible with even few tens of thousands of entities.

1.2 Contributions

The above shortcomings motivate our story generation methodology. A collection of documents is mined and indexed to provide efficient access to nearest-neighbor queries, and a storytelling algorithm is used to direct searches toward desired destination entities. Various pre- and post- processing techniques are utilized to discover stories and explain them. Our contributions are:

1. *Cliques:* We introduce a new network model that employs cliques to define weighted edges between entities (different from the boolean edge models of existing systems). Cliques are used as evidence to support a particular connection between two entities and serve as the primary building block of storytelling. This component addresses issue (A).
2. *Explanation Chain (Clique Chain):* To provide effective explanations, we propose a relationship model between two entities (a story) as a path traversed from one entity to another through a series of cliques. Thus the story between two entities is explained as a chain of cliques where evidence is marshalled by the entities' surrounding neighbors within their respective cliques. This component addresses issue (B).
3. *Efficient Automated Search for Storytelling:* We propose a formulation of heuristic search for finding explanations/clique chains between two entities. The search algorithm can efficiently provide multiple explanations for the identical start and end entities based on two user-defined parameters. This component address issue (C).
4. *Syntactic Storytelling:* User can provide syntactic constraints to the proposed search algorithm to obtain relevant stories and improve processing time. This contribution addresses issue (D).
5. *Coreferencing:* We use coreferencing to disambiguate pronominal references as well as references to the same person. This component addresses issue (E).
6. *Induced Similarity Networks:* Our algorithms operate directly on the vector space model without materializing the network to obtain significant computational efficiency. We demonstrate our ability to scale to large datasets and to rapidly generate/verify multiple hypotheses. This component addresses issue (F).

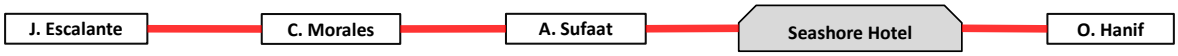
2 Related Projects

Intelligence analysis: Research work into software tools for intelligence analysis can be grouped into model-guided systems [12–14], multi-agent systems [15], graph-based analytic tools [16–19], and collaborative systems [5,20,21]. Our work falls in the category of graph-based analytic tools. We support exploration of the entity-entity graph (without materializing it) to help intelligence analysts to marshal evidence, explain connections, and form hypotheses.

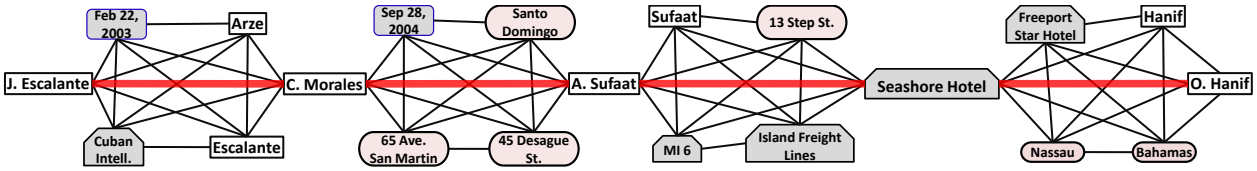
Visual analytic tools: A different way to classify software tools is in terms of the activity they support, i.e., information foraging vs. sensemaking [22]. Some of the tools in the former category are *IN-SPIRE* [6], and *NetLens* [11]; they leave the reasoning processes to the analyst. Other tools, such as *Analyst's Notebook* [3], *Sentinel Visualizer* [4], *Entity Workspace* [5], *Jigsaw* [8], and *Palantir* [7] focus more on the the sensemaking process, and while many of them ostensibly support information foraging, some of these tools are primarily for late stage sensemaking and presentation.

Connecting the dots: The "connecting the dots" problem is not new and has appeared before in a variety of contexts: entity networks [23], image collections [24], cellular networks [25], social networks [26], and document collections [10, 27, 28]. While some of these works can be adapted toward our problem context, the notion of a story in intelligence analysis often deals with relating entities sequentially such that neighboring entities share commonality, whereas the above projects typically require a stronger connecting thread through all entities, not just neighboring entities. Swanson refers to the notion of neighboring commonality as complementary but disjoint (CBD) structures [29], whereby two arguments may exist separately that when considered together lead to new insights, but the objects exhibiting these two arguments are unaware of each other.

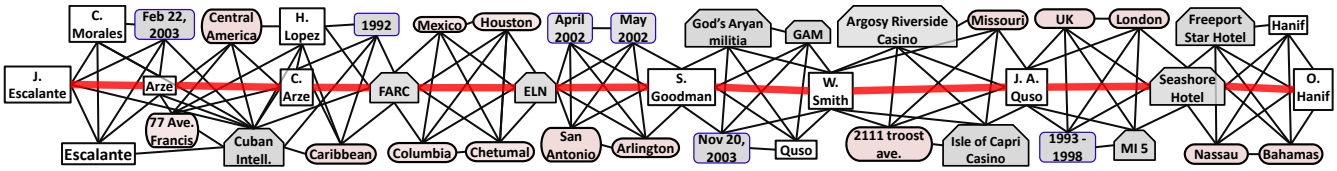
This paper builds upon our prior work in [10,30] in many important ways. First, we build stories through entity networks extracted from documents whereas these prior projects build stories through documents directly. This emphasis on entity networks is closer to how intelligence analysts reason about connections. Second, the



(a) Clique size, k=2 and distance threshold, θ=0.99: Laboratory technician Jose Escalante is connected to Omar Hanif, a key player of a possible terrorist attack and an Al-Queda operative. The connection is explained by a story with three junction nodes: Carlos Morales, Ali Sufaat, and Seashore Hotel.



(b) Clique size, k=6 and distance threshold, θ=0.99: A larger clique size requirement adds more evidence to the story and better explains the connection.



(c) Clique size, k=6 and distance threshold, θ=0.93. A longer clique-chain with $k = 6$ provides another detailed explanation of the connection between Jose Escalante and Omar Hanif.

Figure 2: Different explanations for the connection between two entities: Jose Escalante and Omar Hanif.

present paper significantly generalizes the work in [10] by incorporating the notion of cliques in stories, thus supporting evidence marshalling and explanation generation. Third, we present a novel optimization technique for large databases based on concept lattice mining to support faster story construction.

3 Problem Setting

A story between entities e_1 and e_t is a sequence of intermediate entities e_2, e_3, ..., e_{t-1} such that every neighboring pair of entities satisfies some user defined criteria. We model entities and the documents they occur in using a traditional vector space model. The problem of finding a story then can be modeled as a path search problem in the induced entity-entity graph $\mathcal{E}$, but direct materialization of $\mathcal{E}$ with hundreds of thousands of entities and billions of edges is infeasible. What is needed is support for directed exploration of the graph toward desired entities.

Given a start and end entity, our algorithm induces the network on the fly from the vector space model and finds a path. We allow the analyst to influence the story construction using two distinct criteria: clique size and distance thresholds. Given a story connecting a start and an end entity (see Fig. 2(a)), analysts can perform one of two tasks: they can either aim to strengthen the individual connections resulting in a longer path (see Fig. 2(b)), or they can organize evidence around the given connection (see Fig. 2(c)). We use the notions of distance threshold and clique size to mimic these behaviors. The distance threshold refers to the maximum acceptable distance between two neighboring entities in a story. Lower distance thresholds impose stricter requirements and lead to longer paths. The clique size refers to the minimum size of the clique that every pair of neighboring entities must participate in. Greater clique sizes provide more evidence and tend to provide longer stories.

We use the term *edge* to refer to the basic unit of a direct link between two entities. A k-clique has k entities and is composed of $k(k-1)/2$ edges where every edge satisfies the distance threshold θ. A *clique chain* is composed of a number of consecuting cliques connecting as start and end entity. Clique chains of Fig. 2(a, b, and c) are respectively composed of 2-, 6-, and 6-cliques. Applied distance thresholds to these three clique chains are 0.99, 0.99, and 0.93. Each clique chain provides alternative explanations for the relationship between the same pair of entities. We use the term *story* to refer to a relation between two entities via the junction entities of the corresponding clique chain. The stories are highlighted by thick lines in the clique chains of Fig. 2.

We use the *Soergel distance* between two entities e_1 and e_2 to measure the strength between them:

$$\mathbb{D}(e_1, e_2) = \frac{\sum_{f \in \mathcal{F}} |V(e_1, f) - V(e_2, f)|}{\sum_{f \in \mathcal{F}} \max(V(e_1, f), V(e_2, f))}$$

where $V(e, f)$ indicates the weight of feature f for entity e. Here, the features are the different documents in which the entity appears. Let $e(f)$ be the set of entities associated with feature f, and $f(e)$ be the set of features associated with entity e. Soergel distance is a true distance measure: it is exactly 0.0 when the entities e_1 and e_2 have exactly the same features is symmetric, and obeys the triangle inequality. For entities in a document collection, the weight $V(e, f)$ can be defined as

$$V(e, f) = \frac{(1 + \log(n_{e,f})) \left(\log \frac{|\mathcal{E}|}{|e(f)|}\right)}{\sum_{j=1}^{|f(e)|} \left((1 + \log(n_{e,j})) \left(\log \frac{|\mathcal{E}|}{|e(j)|}\right)\right)^2}$$

where $n_{e,f}$ is the frequency of entity e in document f, $|e(f)|$ is the number of entities in document f, $|e(j)|$ is the number of entities in document j, and $|\mathcal{E}|$ is the total number of entities. Note that this is a variant of tf-idf modeling with cosine normalization in the entity-document space.

4 Approach

Figure 3 summarizes our storytelling framework. The framework takes a document corpus as input, applies algorithmic approaches to handle the research issues in story generation, and outputs stories at the end of the pipeline. We describe the entire process in detail in this section.

Our overall methodology is based on using a concept lattice framework to structure the search for stories. A concept lattice [31] structures the membership of entities in documents into sets of overlaps and relationships between these sets. Recall that the two parameters influencing the quality of the path—distance threshold and clique size—impose a duality. The distance threshold is posed over feature sets whereas the clique size is over entities. We use the clique

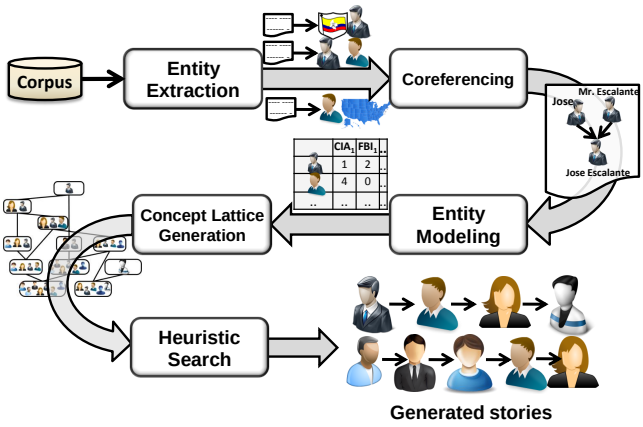


Figure 3: Storytelling pipeline.

size to prune the concept lattice during construction (by incorporating it as a support constraint) and the distance threshold to select candidates for dynamic construction of paths. There are three key computational stages: (i) construction of the concept lattice, (ii) generating promising candidates for path following, and (iii) evaluating candidates for potential to lead to destination. Of these, the first stage can be viewed as a startup cost that can be amortized over storytelling tasks.

We adopt the CHARM-L [31] algorithm of Zaki for constructing concept lattices. The second and third stages are organized as part of an A* search algorithm that begins with the starting entity, uses the concept lattice to identify candidates satisfying the distance threshold and clique size requirements, and evaluates them heuristically for their promise in leading to the end entity. In practice, we will place a limit on the branching factor (b) of the search, thus sacrificing completeness for efficiency. We showcase these steps in detail below, including the construction of admissible heuristics.

4.1 Concept Lattice Construction

We employ the notion of a concept lattice [31] to capture similarities between entities. Each concept/closed set is a pair: (entity set, document set). Concepts capture a maximal co-occurrence between entity sets and document sets, i.e., it is not possible to add more entities to a concept without losing some documents, and vice versa. We order the entity list for each concept by the number of documents. Note that we can find an approximate set of nearest neighbors for an entity e from the entity list of the concept containing e and the longest document set. The concept lattice is a data structure that models conceptual clusters of entities and feature overlaps and is used here as a quick lookup of potential neighbors that will satisfy the distance threshold and clique constraints.

4.2 Successor Generation

Successor generation is the task of, given an entity, using the distance threshold and clique size requirements, to identify a set of possible successors for path following. Note that this does not use the end entity in its computation. We study three techniques for successor generation:

1. Cover Tree Nearest Neighbor,
2. Nearest Neighbors Approximation (NNA), and
3. k-Clique Near Neighbor (kCNN).

Among these three techniques, NNA and kCNN approaches are our contributions and the cover tree approach is used for comparison purpose only.

4.2.1 Cover Tree Nearest Neighbor

The cover tree [32] is a data structure for fast nearest neighbor operations in a space of entities organized alongside any distance metric (here, we use the Soergel distance [33]). The space complexity is $O(\|\mathcal{E}\|)$, i.e., linear in the entity size of the database. A

Algorithm 1 NNA(e)

Input: An entity $e \in \mathcal{E}$
  $fringe \leftarrow \top(e)$ order by feature set size
  $prospects \leftarrow \emptyset$
  while $fringe \neq \emptyset$ **do**
    $r \leftarrow$ dequeue from $fringe$
    while $prospects \neq \emptyset$ **do**
      $e' \leftarrow$ head $prospects$
      if $J(e,e') > \frac{|\text{Items}(e)|}{|\text{Items}(r)|}$ **then**
        yield e'
        dequeue $prospects$
      else
        break
    for all $r' \in$ ChildrenOf r **do**
      add r' to $fringe$ order by feature set size
      for all e' in $e(r')$ **do**
        add e' to $prospects$ order by $J(e,e')$

nearest neighbor query requires logarithmic time in the entity space $O\left(c^{12}\log(n)\right)$ where c is the expansion constant associated with the feature set dimension of the dataset (see [32] for details).

4.2.2 Nearest Neighbors Approximation (NNA)

The second mechanism we use for successor generation is to approximate the nearest neighbors of an entity using the concept lattice. We use the Jaccard coefficient between two entities as an indicator to inversely (and approximately) track the Soergel distance between the entities. In order to efficiently calculate an entity's nearest neighbors, however, we cannot simply calculate the Jaccard coefficient between it and every other entity. We harness the concept lattice to avoid wasteful comparisons.

A formal description of our NNA algorithm is shown in Algorithm 1. The set $\top(e)$ is the set of all redescriptions [31] which form the upper edge in the concept lattice where e appears. $J(e,e')$ of Algorithm 1 refers to *Jaccard coefficient* between two entities is defined as

$$J(e,e') = \frac{|f(e) \cap f(e')|}{|f(e) \cup f(e')|}$$

which is a measure of how similar two entities are based upon how many features they share in proportion to their overall size. NNA returns better approximate redescriptions of an entity first. This is still, however, an approximation since it uses the Jacccard coefficient rather than the Soergel distance.

4.2.3 k-Clique Near Neighbor (kCNN)

The basic idea of the kCNN approach is, in addition to finding a good set of successor nodes for a given entity e, to be able to have

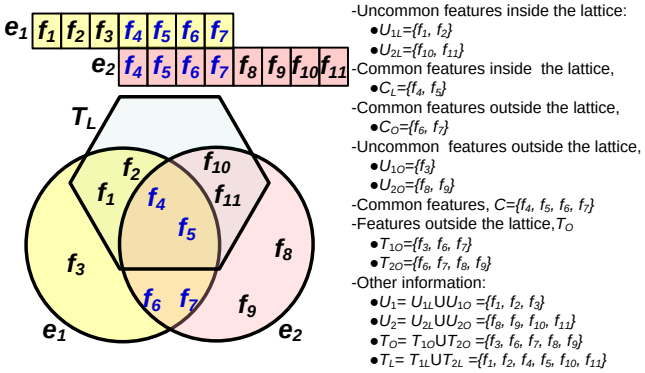


Figure 4: Distribution of common and uncommon features of entities e_1 and e_2 inside and outside the concept lattice. (The hexagon indicates the concept lattice).

$$\mathbb{D}_{\text{mixed}}(e_1, e_2) = \frac{|U_{1L}| \times \text{minw}(e_1) + |U_{2L}| \times \text{minw}(e_2) + \sum_{f \in C_L} |V(e_1, f) - V(e_2, f)|}{|U_{1L}| \times \text{minw}(e_1) + |U_{2L}| \times \text{minw}(e_2) + \sum_{f \in C_L} \max(V(e_1, f), V(e_2, f)) + |T_O| \times \max(\text{maxw}(e_1), \text{maxw}(e_2))}$$

Figure 5: The mixed mode Soergel distance equation.

sufficient number of them so that, combinatorially, they contribute a desired number of cliques. With a clique size constraint of k, it is not sufficient to merely pick the top k neighbors of the given entity, since the successor generation function expects multiple clique candidates. Given that this function expects b clique candidates, minimum number m of candidate entities to identify can be cast as the solution to the inequalities:

$$\binom{m-1}{k} < b \text{ and } \binom{m}{k} \geq b$$

The entity list of each concept of the lattice is ordered in the number of features and this aids in picking the top m candidate entities for the given entity e. We pick up these m candidate entities for o from the entity list of the concept containing the longest feature set and redescription set containing e. Note that, in practice, the entity list of each concept is much larger than m and as a result kCNN does not need to traverse the lattice to obtain promising candidates. kCNN thus forms combinations of size k from these m entities to obtain a total of b k-cliques. Since m is calculated using the two inequalities, the total number of such combinations is equal to or slightly greater than b (but never less than b). Each clique is given an average distance score calculated from the distances of the entities of the clique and the current entity e. This aids kCNN in returning a priority queue of exactly b candidate k-cliques.

4.3 Evaluating Candidates

We now have a set of candidates that are close to the current entity and must determine which of these has potential to lead to the destination. We present two operational modes to rank candidates: (i) the normal mode and (ii) the mixed mode.

4.3.1 Normal Mode

The normal mode is suitable for the general case where we have all the entities and features resident in the database. The primary criteria of optimality for the A* search procedure is the cumulative Soergel distance of the path. We use the straight line Soergel distance for the heuristic. It is easy to see that this can never overestimate the cost of a path from any entity e to the goal (and is hence admissible), because the Soergel distance maintains the triangle inequality.

4.3.2 Mixed Mode

The mixed mode distance measure is effective for large datasets where only important information is stored while other information is removed from the system after recording some of their aggregated information to save on space and cost. With the mixed mode approach, for simplicity, we assume that all the information about items outside the concept lattice are absent but some of their aggregated information like number of features truncated are provided. Figure 4 shows the distribution of common and uncommon features of entities e_1 and e_2 inside and outside a concept lattice.

Figure 5 shows our formula for the mixed mode approach. Consider the set of features T_O that do not appear in the lattice due to the support threshold of the concept lattice, *minsup*. Some features of T_O can be common to both entities e_1 and e_2. $|U_{1O}|$ and $|U_{2O}|$ are the numbers of uncommon features in entities e_1 and e_2, which are thus outside the frequent concept lattice. Length $|T_O|$ is a known variable due to the recorded aggregated information, but $|U_{1O}|$ and $|U_{2O}|$ are unknown. This is why $|U_{1O}|$ and $|U_{2O}|$ do not appear in $\mathbb{D}_{\text{mixed}}(e_1, e_2)$. For $\mathbb{D}_{\text{mixed}}(e_1, e_2)$, we consider that all the features of T_O (i.e., features outside the lattice) are common in both entities e_1 and e_2 and all these features have the same weight which is $\max(\text{maxw}(e_1), \text{maxw}(e_2))$.

To be able to use $\mathbb{D}_{\text{mixed}}(e_1, e_2)$ as a heuristic, it should be mentioned that $\mathbb{D}_{\text{mixed}}(e_1, e_2)$ must never overestimate the original Soergel distance $\mathbb{D}(e_1, e_2)$. The proof is omitted due to space constraints.

4.4 Implementation Details

The aim of the algorithms we have described so far is to support intelligence analysts in marshaling thoughts and evidence in order to generate hypotheses and then to generate defensible and persuasive arguments on hypotheses that are most favored by the evidence. Note that significant domain knowledge is still required to analyze a dataset to find a set of good stories. This raises some implementation issues regarding pre- and post processing, starting points, and exploration constraints. This subsection describes these implementation issues.

4.4.1 Entity Extraction

We used a number of named-entity recognition (NER) APIs to extract entities from a document collection. We used multiple NER APIs because we observed that some named entities are better extracted by one tool but missed by another. The NER APIs we used are LingPipe [34], OpenNLP [35], and Stanford NER [36]. We combined the entities extracted by these three named-entity recognizers and modeled the entities in the vector space model (entities are objects and documents are features).

4.4.2 Syntactic Storytelling

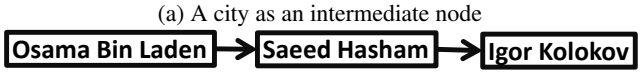
The kCNN approach already uses two constraints: clique size and distance threshold. It is possible to include a syntactic constraint to the successor generation process (Figure 6). For example, we can restrict the exploration in such a way that the junction nodes can be of certain types of entities, e.g., people or organizations. Other entity types such as places, dates, money amount, phone number, etc. are better used as surrounding evidence, and hence can be part of a clique but not as a junction point.

Figure 6(a) shows that the connection between "Osama Bin Laden" and "Igor Kolokov" has a place "Cairo" as a junction node. One way to interpret this connection is that Laden and Kolokov visited Cairo. This interpretation becomes unimportant if the dates of their travel to Cairo are far apart. On the other hand, Figure 6(b) shows that Laden and Kolokov are connected via "Saeed Hasham". This new junction node warrants further investigation about "Saeed Hasham". In the Atlantic Storm dataset discussed later, Saeed Hasham is a direct recruitee of Osama Bin Laden and Igor Kolokov is recruited by Saeed Hasham. There is no direct connection between Laden and Kolokov. Therefore, a syntactic constraint applied on the successor generation module can provide better connections. In the experimental results section we provide performance and quality comparison between two strategies: with and without person/organization as junction nodes (Section 5.4).

4.4.3 Coreferencing

Coreference is the art of determining when two entity mentions in a text refer to the same entity. We use LingPipe's heuristic coreference package [34] to resolve pronominal references to entities. LingPipe's coreference package is based on a greedy algorithm called CogNIAC [37] that visits each entity mentioned in a doc-

(a) A city as an intermediate node



(b) A person as an intermediate node

Figure 6: Using syntactic constraints on story generation.

Text: A Russian named Igor Kolokov was arrested in Cairo on 29 January, 2003 and charged with assault on an Egyptian police officer who had attempted to arrest him for being drunk in public. Kolokov sells medical supplies throughout the Middle East and represents a company in Moscow called Medikat.

Coreferred text: A Russian named Igor Kolokov was arrested in Cairo on 29 January, 2003 and charged with assault on an Egyptian police officer who had attempted to arrest Igor Kolokov for being drunk in public. Igor Kolokov sells medical supplies throughout the Middle East and represents a company in Moscow called Medikat.

Figure 7: Example of coreferencing.

ument in order, and for each mention either links it to a previous linked chain of mentions, or begins a new chain consisting only of the current mention. The resolution of a mention is guided by matchers and anti-matchers that score candidate antecedent mention chains based on properties such as the closest matching alias (using a complex comparison allowing for missing tokens), known alias associations, discourse proximity (how far away the last mention in a chain is and how many are intervening), and entity type. Figure 7 shows a simple example where "Kolokov" and some pronouns are replaced by "Igor Kolokov" using this approach.

4.4.4 *How does the Analyst Know where to Start?*

The question of how to find a starting point is very subjective and depends on the analyst's objectives. Some systems choose use graph-theoretic metrics such as betweenness centrality to identify key players in an entity network. We found significant benefits in classifying documents and using entities extracted from specific classes as candidate starting points. We use the AlchemyAPI [38] to assign each document to its most likely topic category (news, sports, business, law and crime, etc.). The analyst's further perusal of documents in each category helps narrow down the number of documents that might contain potential start and end entities. In section 5.6.2, we describe how we can reduce the start set of documents to solve a particular task with the VAST 2011 dataset.

4.4.5 *Modeling Issues*

So far, we have discussed our framework in the entity space. That is, the network we consider has entities as nodes and each entity is modeled as a vector of documents. Storytelling provides different explanations of a relationship between two entities as form of clique chains where each node of the clique chains is an entity. It is possible to reverse this model and traverse in the document space so that we can obtain clique chains of entities. In this reverse case, documents are modeled as vectors of entities and we can find explanation clique chains for a pair of documents instead of a pair of entities. This reverse modeling is sometimes useful where the investigation is less evidence based and the task is to find a small plot from a large volume of documents. In section 5.6.2, we explain how when we used documents as objects and entities as features, we can identify an imminent threat to a hypothetical city.

5 Experimental Results

Due to confidentiality reasons, our group is unable to disclose many intelligence analysis datasets that we have applied our algorithm on. For this paper, we demonstrate results on many public domain datasets, including i) the Atlantic Storm dataset developed in the Joint Military Intelligence College [39], ii) two datasets from the annual VAST challenge contests in 2010 and 2011, and iii) a dataset of politicians that we harvested from Wikipedia text. These datasets range from having 779 to 230,627 entities.

5.1 Evaluation measures

We use both quantitative and qualitative measures for our evaluation. To compare the performance of different successor generation strategies, we assess the number of nodes explored against the length of the story, and whether this growth is linear or exponential. For a numerical measure of story quality, we assess the pairwise Soergel distances between entities in a story (both consecutive

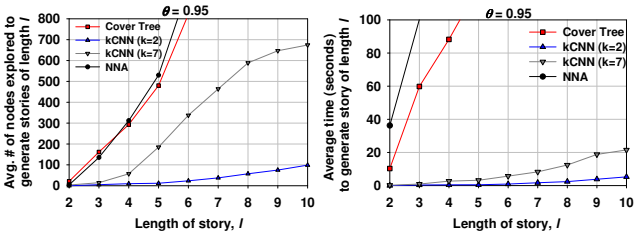


Figure 8: Comparison of successor generation approaches.

and non-consecutive ones) and distill these distances into a dispersion coefficient. The dispersion coefficient assesses the overlap of the features of the entities of a story (not the graph structure of the clique chain) and, thus, evaluates the story, not the clique chain. An ideal story is one that meets the Soergel distance threshold only between consecutive pairs whereas a non-ideal story "oversatisfies" the distance threshold and meets it even between non-consecutive pairs. If n entities of a story are $e_0, e_1, \ldots, e_{n-1}$, then we quantify dispersion coefficient as:

$$\vartheta = 1 - \frac{1}{n-2} \sum_{i=0}^{n-3} \sum_{j=i+2}^{n-1} \text{disp}\,(e_i, e_j)$$

where

$$\text{disp}\,(e_i, e_j) = \begin{cases} \frac{1}{n+i-j}, & \text{if } \mathbb{D}\,(e_i, e_j) > \theta \\ 0, & \text{otherwise} \end{cases}$$

Thus, the dispersion coefficient is 1 for an ideal story and 0 in the worst case when every pair of entities of the story satisfies the distance threshold. Finally, we describe below the analysts interpretations and conclusions from the discovered stories. All the results presented in this paper were obtained using a regular desktop computer with Intel Core2 Quad CPU Q9450 @ 2.66GHz and 8 GB physical memory.

5.2 Evaluating Successor Generation Strategies

The goal of this experiment is to assess the number of nodes explored by the A* search and the time taken as a function of the discovered path length, and as a function of different successor generation strategies. For this purpose, we used the VAST11 dataset and aimed to generate 10,000 stories between randomly selected entity pairs, with a distance threshold of 0.95. Figure 8 depicts the results of the successful searches. As Figure 8 (left) shows, the cover tree and NNA approaches require much more number of node exploration than kCNN with k=2 and 7. The runtime trends shown in Figure 8(right) also mirror the number of nodes explored in Figure 8(left). This result is not surprising, as the cover tree algorithm does not factor into the clique constraint, thus preventing it from taking advantage of the search space reduction that this constraint provides. NNA does take advantage of this constraint, however, and it generates a strict ordering on the Jaccard's coefficient over the cliques, whereas kCNN simply generates some b candidate cliques. In practice, the kCNN relaxation results in the discovery of candidate cliques more rapidly than the NNA algorithm, while still remaining accurate. In both cases a post processing step is necessary to determine if a given candidate does in fact meet the search threshold. Through the remainder of this paper, we thus use the kCNN algorithm as it provides the best performance of the three algorithms.

5.3 Interplay between Clique and Distance Thresholds

We use the VAST11 dataset to study the effects of varying distance and clique size thresholds. As expected, the number of possible stories decreases monotonically with stricter distance and clique size requirements. Figure 9 shows the relationship between the largest

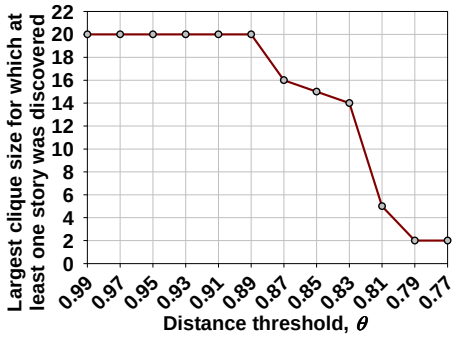


Figure 9: VAST 2011 Text Dataset (Entities are objects and documents are features): Relation between clique chain and distance threshold.

available clique size as distance thresholds become progressively stricter.

5.4 Comparison between Syntactic and non-syntactic Storytelling

To explore this phenomena, we apply apply our algorithms on a dataset of politicians harvested from Wikipedia.org (contains 230,627 entities and 49,612 documents). We allow only person/organization type of entities as junction points of a clique chain in syntactic storytelling. On the other hand, non-syntactic storytelling can use any type of entities as junction nodes. To evaluate the performance of these two approaches, we generated 10,000 random pairs of politicians and invoked syntactic and non-syntactic storytelling with a range of distance threshold and clique size. Figure 10 compares the results of syntactic and non-syntactic storytelling with those start and end entity pairs for which at least one story was found by both the approaches. Figure 10 (left) shows that the average dispersion is better (i.e., higher) for syntactic storytelling with any distance threshold. This plot also depicts an increase in average dispersion with stricter (lower) distance thresholds for both syntactic and non-syntactic approaches.

Figure 10(middle) compares syntactic and non-syntactic versions of storytelling in terms of average clique size as a function of distance threshold. It shows that the average clique size is smaller with the syntactic approach at any distance threshold. This indicates that the extra syntactic constraint applied to the successor generator reduces the size of the neighborhood. As a result, the explanations (with cliques) are generally smaller than the non-syntactic version.

Figure 10(right) shows the distributions of the costs of the paths of the discovered stories for both syntactic and non-syntactic strategies. It shows that the syntactic approach discovers costlier paths than the non-syntactic approach. The extra syntactic restrictions at the junction nodes result in costlier and hence rare stories that the non-syntactic approach might not discover. Around 60% of the discovered stories for the non-syntactic approach have cost between 4.0 to 8.0 where only 36.5% of the stories with the syntactic approach fall into this range. Note that both the approaches yield a right skewed distribution, but the syntactic approach has a longer right skew than the non-syntactic approach indicating generation of costlier stories.

5.5 Comparison between Storytelling and an uninformed Search

We used the VAST 2011 dataset to compare storytelling against an uninformed search. Figure 11 (a) shows that the use of Soergel distance heuristic improves the average branching factor over the vanilla BFS (h=0). Overall improvement of average branching factor is more than 50%. Figure 11(b) shows that the use of the straight line Soergel distance for the heuristic exhibits lower runtime than exploration by the A* procedure over BFS (h=0), for a range of different clique sizes. The average time saved is more than 60%, even with the smallest clique size k=2. The heuristic tends to save additional time as larger clique size requirements are imposed.

Figure 11 (c) and (d) respectively depict the average effective branching factor and average time to generate stories as functions of clique size, for different approaches. Despite the use of the truncated dataset, Figure 11 (c) and (d) show that the mixed mode gains due to the heuristic over the BFS have a similar trend to the normal mode of Figure 11 (a) and (b). Therefore, the mixed mode analysis offers a practical mechanism to provide the best possible gains from lossy datasets without time consuming remodeling of the vector space (e.g., [10] uses a costly remodeling as a post processing step).

5.6 Use Case Studies

In this subsection we describe some of the illustrative results we obtained using some benchmark intelligence analysis datasets.

5.6.1 Atlantic Storm Dataset

The Atlantic Storm dataset was developed in the Joint Military Intelligence College as part of an evidence-based case study. We extracted 779 entities from the Atlantic Storm text dataset. An illustrative example of some stories are already shown in Figure 2 which provides multiple explanations of the relationship between two persons: Jose Escalante and Omar Hanif. Escalante and Hanif play an important role in transferring biological agents from Europe to the Caribbean.

After analyzing some stories, the user obtains a mental model of the terrorist attack. An illustrative mental model is shown in Figure 12. It depicts that the conspiracy starts from Osama Bin Laden in Afghanistan. Two of his close contacts are Saeed Hasham and Fahd al Badawi. Once these two contacts are discovered it becomes easy to find stories involving Igor Kolokov, Pyotr Safrygin, and Boris Bugarov. All these three people are former employees of Russian State Research Center of Virology and Biotechnology (Vector). Among them Boris Bugarov is known to be a bioweapon scientist. Another person named Abdellah Atmani, who works for Holland Orange Ship Lines, helps in smuggling the biological agents to the Caribbean from Morocco. After discovering the plot up to this point, a story involving four key entities (Escalante, Arze, Morales, and Sufaat) reveals that the biological agents will be transferred to the USA from the Bahamas. Omar Hanif and Adnan Hijaji are two key players who recruited Al Qaeda field agents to tansport the biological agents to the USA by several cruise ships.

5.6.2 VAST 2011 Dataset

The VAST11 dataset contains 4,447 documents. The task is to find any imminent threat to the imaginary city named "Vastopolis". We extracted a total of 55,109 entities from 4,447 documents. To find a starting set to work with, we first classify each of the documents using AlchemyAPI (Section 4.4.4). We reduced the set of 4,474 documents down to 122 by selecting the articles that were classified as "law and crime" or were considered "unclassifiable". This narrows down the potential number of terminal entities (start and end) to a few hundred from 55,109 entities. (Note that once given a start and end point, our storytelling algorithm still uses the entire collection.)

Since the number of entities is too large for an analyst to work with and the possible imminent threat is hidden in the news articles, we modeled the documents as vectors of entities (as described in Section 4.4.5). In the actual solution plot there are only around ten entities and thirteen documents involved. There are other stories that can be discovered from this dataset but they do not pose any imminent threats to Vastopolis.

An example story from the dataset is: [03435.txt → 00783.txt → 02566.txt → 01785.txt → 03212.txt]. The start document contains an entity named "Paramurderers of Chaos" which is the name

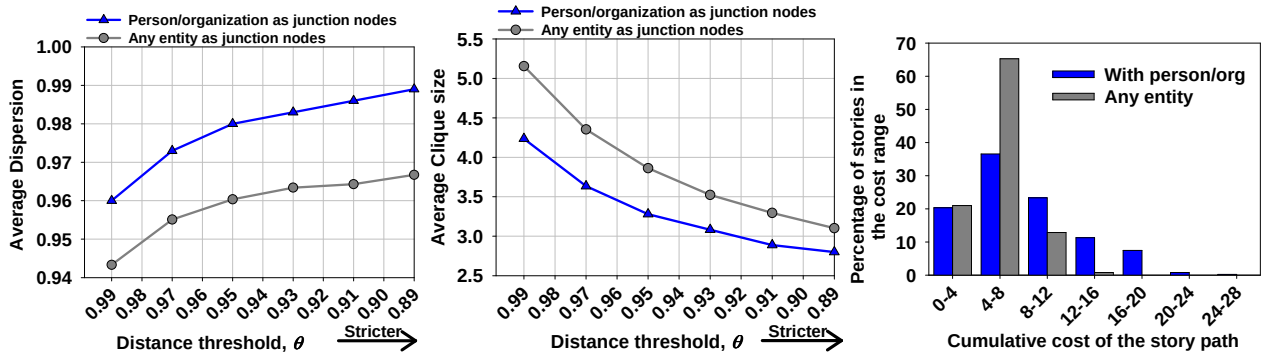
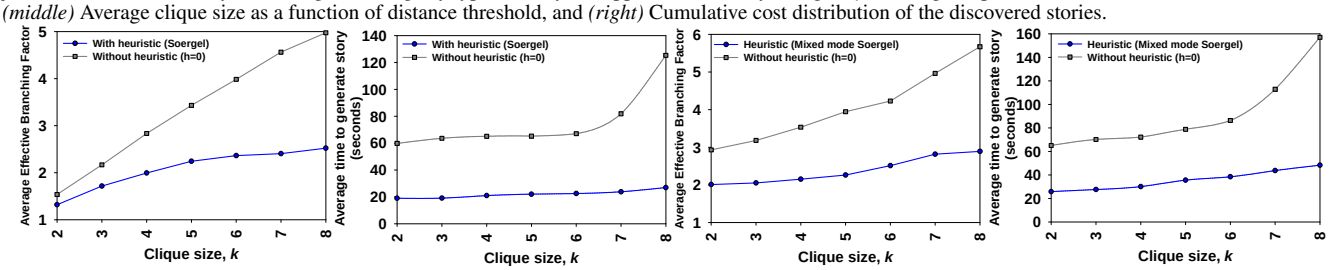


Figure 10: Wikipedia dataset(Entities are objects and documents are features). Comparison between syntactic (i.e., allowing only person/organization as junction nodes) and non-syntactic (e.g., allowing any type of entity as) approaches of storytelling: *(left)* Average dispersion as a function of distance threshold, *(middle)* Average clique size as a function of distance threshold, and *(right)* Cumulative cost distribution of the discovered stories.



(a) (b) (c) (d)

Figure 11: VAST 2011 Text Dataset (Entities are objects and documents are features). Effectiveness of the heuristic in terms of average effective branching factor and average runtime. (a) and (b) show a comparison between our storytelling and an uninformed search. (c) and (d) also depict similar plots but in the mixed operational mode.

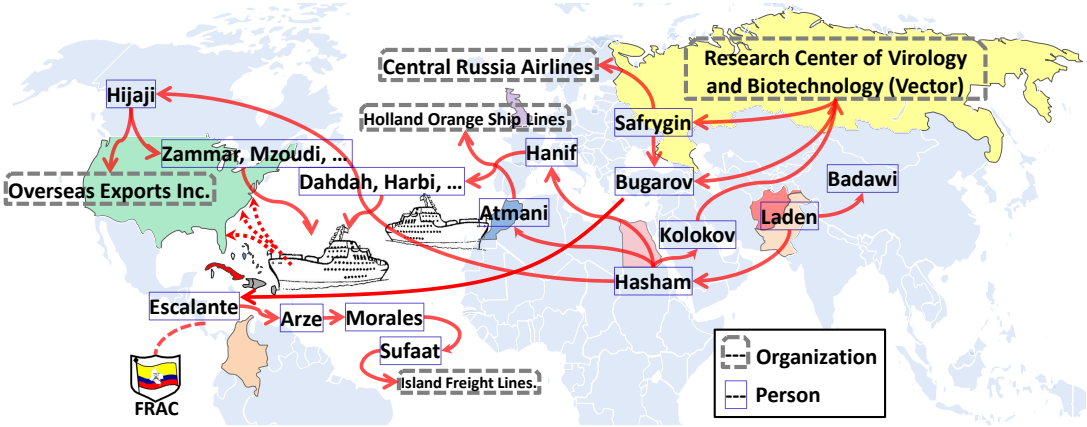


Figure 12: A sketch of the solution of the Atlantic Storm dataset.

of a terror group and the end document contains a report regarding a talk of professor (Edward Patino), another important entity in this plot. The algorithm selects documents 00783.txt, 02566.txt and 01785.txt as junction documents. The first two junction documents (00783.txt, 02566.txt) are related to computer viruses, worm threats, and security flaws in the Vast University. The next document (01785.txt) is related to a Robbery at Vast University (equipments of Professor Edward Patino were stolen). 01785.txt is one of the key documents in the solution which is properly picked by our algorithm. Note that document 00783.txt is considered a "false lead" in the solution planted by the VAST 2011 contest organizers. This study demonstrates that out algorithm was able to identify several key elements hidden inside a large corpora of news articles as well as a few false leads related to the actual plot. Several stories combined together and explanations provided by the cliques lead an analyst to the mental model of the whole plot.

5.6.3 VAST 2010 Dataset

Unlike the VAST 2011 dataset, VAST 2010 dataset is composed of transcribed phone calls, email messages, and reports from fields agents. The task is to investigate arms dealing. We extracted a total of 621 entities from this dataset, of which 102 are person entities. After reading a few documents, the user becomes interested in two people: Igor Sviatoslavich and Viktor Bout. The user first uses syntactic storytelling so that the junction points are people or organizations, but syntactic storytelling reported that there is no story between the given entities. A broader search reveals the result shown in Figure 13, where the junction nodes are locations. Both the two intermediate cliques of the chain are validated in the solution provided by the VAST 2010 Contest organizers. This story is part of an arms deal in the Iran region of the entire plot (called Iran network). Some places used to transport arms from North Korea to Iran were Colombo (Sri Lanka), Pyongyang, and Don Mueang. The story successfully reveals these locations. The story also reveals another person "Soltanzadeh" who is part of the Iran arms dealing network.

6 Discussion

We have presented a novel approach to storytelling between entities in large document collections, with applications to intelli-

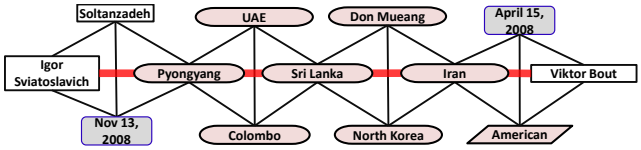


Figure 13: An example clique chain connecting Igor Sviatoslavich with Viktor Bout. Both of them are involved in arms dealing in the Iran network.

gence analysis. Our algorithms are both efficient at handling large datasets and effective at teasing out complicated plots from textual corpora. In the future, we aim to explore ways to automatically incorporate user feedback about presented stories, so as to dynamically adjust the weightings of documents and/or the similarity functions. This will enable analysts to preferentially explore certain types of stories featuring preferred entity types or subplots. We will also explore incorporating additional sources of data, besides text documents, e.g., social media. Finally, we plan to continue our dialog with intelligence analysts to further develop compelling software tools for analytics.

7 References

[1] R. J. Heuer, *Psychology of Intelligence Analysis*. Center for the Study of Intelligence, CIA, 1999. [Online]. Available: https://www.cia.gov/library/

[2] R. Clark, *Intelligence Analysis: a Target-centric Approach*. CQ Press, 2003.

[3] i2group, "The Analyst's Notebook," Last accessed: May 26, 2011, http://www.i2group.com/us.

[4] FMS Advanced Systems Group, FMS Inc., "Sentinel Visualizer," Last accessed: May 26, 2011, http://www.fmsasg.com/.

[5] E. Bier, E. Ishak, and E. Chi, "Entity Workspace: An Evidence File That Aids Memory, Inference, and Reading," in *ISI '06*, 2006, pp. 466–472.

[6] PNNL, "Pacific Northwest National Laboratory, INSPIRE visual document analysis," Last accessed: May 26, 2011, http://in-spire.pnl.gov.

[7] H. Khurana, J. Basney, M. Bakht, M. Freemon, V. Welch, and R. Butler, "Palantir: a Framework for Collaborative Incident Response and Investigation," in *IDtrust*, 2009.

[8] Information Interfaces Research Lab., Georgia Tech, "Jigsaw: Visual Analytics for Exploring and Understanding Document Collections," Last accessed: February 9, 2012, http://www.cc.gatech.edu/gvu/ii/jigsaw/.

[9] Y. Kang and J. Stasko, "Characterizing the Intelligence Analysis Process: Informing Visual Analytics Design through a Longitudinal Field Study," in *VAST*, 2011.

[10] D. Kumar, N. Ramakrishnan, R. Helm, and M. Potts, "Algorithms for Storytelling," *TKDE*, vol. 20, no. 6, pp. 736–751, 2008.

[11] H. Kang, C. Plaisant, B. Lee, and B. B. Bederson, "NetLens: Iterative Exploration of Content-actor Network Data," *Info. Vis.*, vol. 6, no. 1, pp. 18–31, 2007.

[12] R. Alonso and H. Li, "Model-guided Information Discovery for Intelligence Analysis," in *CIKM '05*, 2005, pp. 269–270.

[13] A. Koltuksuz and S. Tekir, "Intelligence Analysis Modeling," in *ICHIT '06*, 2006.

[14] K. Chopra and C. Haimson, "Information fusion for intelligence analysis," in *HICSS '05*, 2005, p. 111a.

[15] E. Lindahl, S. O'Hara, and Q. Zhu, "A Multi-agent System of Evidential Reasoning for Intelligence Analyses," in *AAMAS '07*, 2007, pp. 279:1–279:6.

[16] J. Gersh, B. Lewis, J. Montemayor, C. Piatko, and R. Turner, "Supporting Insight-based Information Exploration in Intelligence Analysis," *Commun. ACM*, vol. 49, April 2006.

[17] T. Coffman, S. Greenblatt, and S. Marcus, "Graph-based Technologies for Intelligence Analysis," *Commun. ACM*, vol. 47, pp. 45–47, March 2004.

[18] P. Crossno, B. Wylie, A. Wilson, J. Greenfield, E. Stanton, T. Shead, L. Ice, K. Moreland, J. Baumes, and B. Geveci, "Intelligence Analysis Using Titan," in *IEEE Symposium on Visual Analytics Science and Technology*, 2007, pp. 241–242.

[19] G. Chin, O. A. Kuchar, P. D. Whitney, M. Powers, and K. E. Johnson, "Graph-based comparisons of scenarios in intelligence analysis," in *SMC '04*, vol. 4, 2004.

[20] E. Bier, S. Card, and J. Bodnar, "Principles and Tools for Collaborative Entity-Based Intelligence Analysis," *TVCG*, vol. 16, no. 2, pp. 178–191, 2010.

[21] M. Chau, J. J. Xu, and H. Chen, "Extracting Meaningful Entities from Police Narrative Reports," in *Annual National Conference on Digital Government Research*, 2002, pp. 1–5.

[22] P. Pirolli and S. Card, "The Sensemaking Process and Leverage Points for Analyst Technology as Identified through Cognitive Task Analysis," in *ICIA '05*, 2005.

[23] L. Fang, A. D. Sarma, C. Yu, and P. Bohannon, "Rex: explaining relationships between entity pairs," *Proc. VLDB Endow.*, vol. 5, no. 3, pp. 241–252, 2011.

[24] K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and L. Guibas, "Image Webs: Computing and Exploiting Connectivity in Image Collections," in *CVPR*, 2010.

[25] J.-P. Brassard and J. Gecsei, "Path Building in Cellular Partitioning Networks," in *ISCA '80*, 1980, pp. 44–50.

[26] C. Faloutsos, K. S. McCurley, and A. Tomkins, "Fast Discovery of Connection Subgraphs," in *KDD '04*, 2004.

[27] M. S. Hossain, J. Gresock, Y. Edmonds, R. Helm, M. Potts, and N. Ramakrishnan, "Connecting the Dots between PubMed Abstracts," *PLoS ONE*, vol. 7, no. 1, p. e29509, 2012.

[28] D. Shahaf and C. Guestrin, "Connecting the Dots between News Articles," in *KDD '10*, 2010, pp. 623–632.

[29] D. R. Swanson, "Complementary Structures in Disjoint Science Literatures," in *SIGIR '91*, 1991, pp. 280–289.

[30] M. S. Hossain, C. Andrews, N. Ramakrishnan, and C. North, "Helping Intelligence Analysts Make Connections," in *AAAI '11 Workshop on Scalable Integration of Analytics and Visualization*, 2011.

[31] M. J. Zaki and N. Ramakrishnan, "Reasoning About Sets Using Redescription Mining," in *KDD '05*, 2005.

[32] A. Beygelzimer, S. Kakade, and J. Langford, "Cover Trees for Nearest Neighbor," in *ICML '06*, 2006, pp. 97–104.

[33] A. Leach and V. Gillet, *Introduction to Chemoinformatics*. Springer, 2007.

[34] Alias-i., "LingPipe 4.1.0," Last accessed: Jan 31, 2012, http://alias-i.com/lingpipe, 2008.

[35] Apache Software Foundation, "OpenNLP," Last accessed: Jan 31, 2012, http://incubator.apache.org/opennlp.

[36] Stanford Natural Language Processing Group, "Stanford NER: http://nlp.stanford.edu/software/CRF-NER.shtml," Last accessed: Jan 31, 2012.

[37] B. Baldwin, "CogNIAC: High Precision Coreference with Limited Knowledge and Linguistic Resources," in *ACL/EACL'97*, Spain, July 1997, pp. 38–45.

[38] AlchemyAPI., "AlchemyAPI: http://www.alchemyapi.com/api/categ/."

[39] F. J. Hughes, "Discovery, Proof, Choice: The Art and Science of the Process of Intelligence Analysis, Case Study 6, "All Fall Down", Unpublished report," 2005.