Abstract – This paper discusses the challenges of and possible methods for data association in the domain of counterinsurgency where “soft/linguistic” data is an important input data type. An overview of the processing operations from input to construction of fused estimates is described. The design issues that are discussed and require further exploration to yield a workable and efficient association process include developing an input batching logic, finding efficient ways to search between graphs, and the selection of appropriate semantic similarity metrics to associate nodes and arcs. Additionally, the solution to a multi-dimensional assignment problem and graph merging techniques will need to be defined. The application of data association in this type of environment has potential to yield an improved, comprehensive data graph which will aid in reducing search time and provide more accurate results for analysts making real time decisions in the real world.

Keywords: Data association, batching, semantic similarity, multi-dimensional assignment problem, graph merging, graph matching.

1 Introduction

Experiences in Iraq and Afghanistan in dealing with insurgency/counter-insurgency problems have required the (ongoing) formulation of entirely new paradigms of intelligence analysis and dynamic decision-making. Depending on the phases of counter-insurgency (“COIN”) operations, the nature of decision-making ranges from conventional military-like to socio-political. Because of this rather wide spectrum of action, the nature of information support required has an equally wide range. Since automated Information Fusion (IF) processes provide some of the support to such decision-making, requirements for IF process design must address these varying requirements, resulting in considerable challenges in IF process design. Further, these experiences have also shown that some of the key observational and intelligence data in COIN operations comes from dismounted soldiers reporting on their patrol activities. (From the US Army Field Manual on COIN [1]: “Intelligence for current operations comes from a variety of sources, but operations reports are particularly important. This is because current enemy activities are more often reported by patrols, units conducting raids, or observation posts than they are by dedicated intelligence collectors.”) These data are naturally communicated in language in the form of various military and intelligence reports and messages. Such data finds its way into IF processes as raw digitized text, and this input modality creates new challenges to IF process designs, contrasted with more traditional IF applications involving the use of highly-calibrated, numerically precise observational data. In this paper, we focus on the implications of this relatively new input stream on the design of a Data Association process for this “soft” type of data.

Within any automated or semi-automated IF process, the data association function is that which in effect partitions the data at hand (typically but not exclusively observational data) into subsets to be allocated or assigned to estimation algorithms that are able to fuse and exploit this information to estimate some attribute or quality of some entity. This function must be able to (a) nominate the hypothesized entities to which the data would be associated, (b) evaluate the competing alternatives of feasible data-to-entity assignments, and (c) select the “best” data assignments based on some objective function.

The function of data association can be applied in many different types of data and entity-rich environments. Some defense applications that have been studied previously include automated target recognition and guidance and control of autonomous vehicles. Data association may also be applied in the defense domain of “asymmetric” or “irregular” warfare, to include COIN problems, as has been mentioned. Here, field observation reports can come in the form of verbal, textual or sensor observations and relay information that can be used to assess the real world.

Currently, there is a great influx of information being received from these sources. There are many ways that have been developed to absorb this data and postulate answers to various posed questions. For example, if the information is represented in a graphical form, graph matching may be utilized to discover relevant patterns in the data (matching graphical templates of conditions of interest (we call these “target graphs”) to graphical structures representing the observational data). In regard to graph matching as a discovery-based inferencing technique, significant research has been performed by the
team at the Center for Multisource Information Fusion (CMIF) as well as by others; see [2] for examples. However, there is a lack of exploration into the processes and algorithms needed for data association or graph containment in these applications.

The focus of the research reported in this paper involves investigating the design and use of data association and graph merging methods prior to a graph matching process and assessing the benefits that are observed. The addition of these tools to graph matching will be tested to show their effectiveness in decreasing the amount of storage space required by the graphs, decreasing search time, as well as providing more comprehensive nodes for the matching.

2 System process

2.1 Streaming data environment

To understand how data association is applicable to this specific IF process, a view of the overall process flow is shown in Figure 1-A. The process first begins with the observations of events and situations as seen by multiple human sources. These observations can occur at some sample rate and humans undergo an “internal fusion” process as part of the perception-cognition-comprehension sequence. These observations are then transformed into utterances and are translated into digital text. Note that we are addressing the challenging IF environment involving streaming data.

The next step in the process is that the text is sent through some text extraction/entity-extraction process which identifies entity IDs, attributes, and inter-entity relationships. In this discussion, we assume that the data output from this process will already have undergone coreference resolution (in our project, a separate team is working on these functions and we have coordinated this partition). This step of the process is designed to remove pronouns from the data and replace them with the correct referent; it also reconciles variants of named entities in different sentences (within a given, presumed multisection message), as well as across different messages. It should be understood that this operation culminates in what we call a “propositional graph”, and that, as hinted at above; we are employing graphical methods both for representational purposes and as analytical purposes.

Once the information has been extracted, it undergoes a data association process (see Figure 1-B). Then, the associated nodes and arcs of the candidate graphs (of a batched-message set) can be merged together, this merged-data-graph representing the composite observational data of the current message-set. The message-batching logic (not discussed here) might include a window of time or some fixed quantity of messages. After the multi-message association is complete and an associated-message graph-merging is done, this associated/merged data graph will need to be input into the prior-time-slice existing data graph representing the observed world. This second step may require a second data association procedure to account for additional similarities between this data (i.e. the newly-formed merged/associated data graph and the prior-time cumulative data graph). Ideally, this process will add the optimal quantity and quality of information to the observed-data graph. This will assist in both size containment and the most favorable matches to the template graph, and thus the best discovered inferences from the associated observational data.

2.2 Deduction versus discovery

Since this research focuses on the unpredictable and irregular world of counterinsurgency, we judge that a deductive model based approach will be problematical. The traditional deductive model is developed in environments where reliable, stable knowledge can be employed to develop necessary models of a dynamic environment for IF process design. Based on prior literature searches and associated analyses [3], we have judged that such knowledge and model-development methods are still evolving and immature for the COIN case. In this case, the asymmetric nature of the adversaries requires the real time decision making abilities provided by a discovery based approach. In a discovery method, we assert that analysts can identify patterns of meaningful data that can be discovered by various automated pattern-discovery techniques that in turn may yield actionable information. In our approach, the pattern discovery approach involves graph-matching methodologies (see Section 4.1).

3 Data association within the prototype fusion process

While Figure 1-A provides a higher level view of the process as a whole, there are many sub-processes that are also being performed. One such sub-process includes data association (Figure 1-B). It is an intricate part of this system that involves many components.

3.1 Initial requirements of data association

Initially, the pertinent information is received from the enhanced text extraction process and is batched until some appropriate conditions have been satisfied. These conditions may depend on the current state of the world and the arrival rates of incoming messages from the field. The more dynamic the circumstances in the real world, the higher the required observational sampling rate will be and the more frequent data association may need to occur. Additionally, the batching rate may also be affected by the processing capacities of the various operations. Once they have met the batching conditions, data association can now be continued.
3.2 Data association operations

Data association can be broken down into three major steps, which consist of hypothesis generation, hypothesis evaluation and hypothesis selection. They should be followed by the handbook here.

3.2.1 Hypothesis generation

In hypothesis generation for the soft data of interest here, the hypothesis nomination process is self-generated as a result of the inherent semantic content of the messages, i.e. by the entities embodied in the language. For any set of “n” batched messages and thus “n” propositional-graphs, a graph-search procedure is one design challenge, as a technique to structure the “similarity” comparison among semantic content of the messages will have to be developed; this is Design Challenge 1. While a variety of graph-searching techniques exist [4], the method of choice will need to be one that is efficient at searching between graphs. This methodology leads to the formulation of an assignment matrix, populated by association scores that come from the Hypothesis Evaluation function, described next.

3.2.2 Hypothesis evaluation

Once the association/assignment matrix structure has been formulated (note that this may be “N-dimensional” if more than two messages are being compared), the next phase is to determine the degree of similarity between the nodal and arc elements according to their semantic content. In the design of the Data Association function, this is the hypothesis evaluation phase. Here, semantic similarity metrics will need to be defined and applied to calculate the degree of pair-or-N-wise similarity across the content of the candidate messages (this is Design Challenge 2). Both intra-message and intra-graph data association will be required and will initially be done at the word level. Examples of this include determining the relationship between words such as tall, big, and large. To accomplish this task, relationship scores will be assigned which determine the closeness ratings of the various node and arc pairs. Even with these scores, it is expected that an optimization-based assignment algorithm solution will need to be used to best associate the data.

Alternative data association process designs will be considered that will consider associations at the triple level or relation level in the message graph structures. This will be determined after a sufficient literature review has taken place. If the review is inconclusive, tests may be conducted using sample data to establish which, if any, of the alternative designs will yield the optimal results. Furthermore, an obstacle that may be encountered is transforming the output received into a usable input for the association process. It is thought that the output will be presented in the form of nodes and arcs, which not only represent entities but also their attributes. A process may need to be developed which is capable of converting these graphs into Directed Attributed Relational Graphs (DARG).

3.2.2.1 Semantic similarity metrics

When comparing messages, it is necessary to compare their respective graphs to one another in order to assess their degree of similarity. These comparisons may include determining if the graphs are identical to one another or have similar sub-graphs. If they are changed over time, a determination will need to be made to see if they originated from the same graph. One type of comparison in particular involves studying the similarity between nodes. There are many different types of similarity algorithms that have been developed to determine the degree of similarity between two graphs. This section will...
discuss some existing algorithms that have been established. Our Design Challenge 3 will involve reviewing and assessing which of the various options for such similarity scores we will use in prototyping an initial approach to Data Association processing.

In our ongoing analysis, we have explored the modern literature on semantic similarity; here we review some of the strategies in that literature. The development of machine readable dictionaries, lexicons, thesauri, and taxonomies has helped aid the organization of words into a semantic space. An article by Jiang and Conrath [5] discusses the use of a corpus-based approach and lexical taxonomies to calculate the semantic similarity between words. Calculating the semantic association can be estimated by the conceptual distance between nodes (similarity between words) in the taxonomy. A hierarchically structured taxonomy can be useful in estimating the semantic similarity between nodes in the network. Two specific approaches used to determine the conceptual similarity of two words in this type of network are known as node and edge based approaches. The node based approach relates to the information content approach while the edge based approach corresponds to the conceptual distance approach.

The information content approach developed by Resnik (in [5]) is one method to determine the conceptual similarity. The information value in terms of a hierarchal network is the information content value of a super-ordinate class (nodes that share common information). The information content of a certain class can be calculated by finding the probability of occurrence of that specific class in a large text corpus (Information Content (c) = log\(^{-1}\)P(c)), where P(c) is the probability of encountering an instance of concept c). The similarity between two words can be calculated using the formula

\[ \text{sim}(c_1, c_2) = \max_{c \in \text{Sup}(c_1,c_2)} \left[ \text{Information Content}(c) \right] \]

\[ = \max_{c \in \text{Sup}(c_1,c_2)} \left[ -\log * p(c) \right] \]  

(1)

Here, Sup(c\(_1\),c\(_2\)) represents the set of words that include both c\(_1\),c\(_2\). Another equation can be used where words can have more than one meaning

\[ \text{sim}(w_1, w_2) = \max_{c_1 \in \text{sen}(w_1), c_2 \in \text{sen}(w_2)} \left[ \text{sim}(c_1, c_2) \right] \]

(2)

The probability using a maximum likelihood estimation can be calculated using the equation

\[ P(C) = \frac{\text{freq}(c)}{n} \]  

(3)

where the \text{freq}(c)=\sum\text{freq}(w)/\text{classes}(w).

Jiang and Conrath [6] developed their own method to compare the sum of the information content of the individual concepts with the lowest common subsumer. Here, a subsumer is the point of the hierarchy at which the concepts share a parent. This method incorporates the use of a conditional probability of encountering an instance of a child-synset given an instance of a parent-synset. In this instance, a synset is a group of data that is considered semantically equal. The equation used to find this semantic similarity is notated as

\[ \text{sim}(c_1, c_2) = \frac{1}{\text{IC}(c_1) + \text{IC}(c_2) - 2 \cdot \text{IC}(\text{lcs}(c_1,c_2))} \]

(4)

where \text{IC} is the information content of concept c and lcs is the lowest common subsumer. The lowest common subsumer signifies the most specific concept that is shared by two concepts [7]. Although this approach is very similar to Resnik’s proposed computation upper bound and thus potentially yielding the same results, the advantage is that it does not rely on a corpora analysis. This will avoid the sparse data problem present in many corpus based approaches [8].

Lin (in [6]) uses a similar formula for computing the degree of similarity between two concepts. Lin’s method scales the information content of the lowest super-ordinate (iso, having the same meaning as lcs) with the sum of information content of two concepts through the function

\[ \text{sim}(c_1, c_2) = \frac{2 \cdot \text{log}\left(\text{IC}(c_1,c_2) \right)}{\log p(c_1) + \log p(c_2)} \]

(5)

Another measure of similarity that has been developed by Leacock and Chodorow (in [6]) uses the normalized path lengths between two concepts (c\(_1\) and c\(_2\)). The degree of similarity between two concepts can be calculated using the equation

\[ \text{sim}(c_1, c_2) = -\log\left(\frac{\text{length}(c_1,c_2)}{2 \cdot \text{MAX}}\right) \]

(6)

where the length of (c\(_1\),c\(_2\)) represents the number of edges on the shortest path between concepts and MAX is the depth of the taxonomy (in [7]). One limitation of this method is that it focuses on IS-A links and scales the path length by the depth of the taxonomy.

Wu and Palmer evaluate similarity by determining their closeness in terms of structural relationship which can be defined by:

\[ \text{sim}(c_1, c_2) = \frac{2 \cdot N_1}{N_1 + N_2 + 2 \cdot N_3} \]

(7)

To find the values of N\(_i\), c\(_3\) must first be found. In this case, c\(_3\) is the lowest common subsumer of c\(_1\) and c\(_2\). N\(_1\) and N\(_2\) can be calculated as the number of nodes between c\(_1\) and c\(_3\) and c\(_2\) and c\(_3\) respectively. N\(_3\) is then determined based on the number of nodes between c\(_3\) and the root [9].

Hirst and St. Onge (in [6]) calculate semantic similarity based on the semantic relatedness of two lexicalized concepts. Here, concepts are considered closely related if and only if their WordNet synsets are linked by a path that is not too long and does not frequently change in direction. This relatedness is calculated using the equation
\[ r_{HS}(c_1, c_2) = c - \text{path length} - k \cdot d. \quad - (8) \]

Here, \( c \) and \( k \) are constants and \( d \) represents the number of changes in direction of the path. It is noted that if the result of the above calculation is zero then the synsets are said to be unrelated.

As noted above, an extensive literature review is being conducted to determine which algorithm would be the most appropriate for our COIN-based IF Data Association application. This will involve analyzing the benefits and limitations of the above algorithms. An article by Pedersen et al. [10] analyzes some of these metrics along with advantages and disadvantages. Table 1 below, from Pedersen, summarizes these findings. The best method will be chosen and implemented to calculate the semantic similarity between nodes. The accuracy of these calculations are pertinent to merging graphs to not only make the process more efficient but accurate.

### 3.2.3 Hypothesis selection

The last phase of the data association process within fusion is that of hypothesis selection. After similarity scores are assigned to each node and arc comparison, it is likely that an assignment problem will need to be formulated. Comparing more than two messages may result in a multi-dimensional NP-hard assignment problem, depending on how a solution is framed. This is a traditionally difficult problem and the best solution to this formulation needs to be found (this is Design Challenge 4). One option that has been used for these types of problems is a Lagrangian relaxation algorithm, the application of which yields a sub-optimal solution. Of course, the fusion community has studied assignment problems and solutions extensively, and we are aware of most if not all modern solution methods that we will review in regard to making this design choice.

### 3.3 Graph Merging

After the hypothesis selection is complete, a data graph needs to be generated that is representative of both the associated, merged and unmerged data. To be considered for merging, the scores will likely need to meet some given minimum association threshold similarity value to minimize inappropriate data fusion. Another factor to be considered is what identifying information is to be kept from each of the merged nodes and arcs and how much is removed. This gives rise to the question of the loss of possibly valuable data, so care must be given to choose an appropriate method of merging. Once a merged data graph has been created, it must be compared with the existing, prior-time cumulative observed data graph. A graph matching technique must also be applied here to determine if existing nodes and arcs are already represented within the cumulative data graph. This check will assist to keep the cumulative data graph as small as possible. The details of these thresholding and message-handling procedures comprise Design Challenge 5. Note that associated nodes and arcs in the final cumulative data graph will have fused sets of attributes, concatenated from the contributing nodes or arcs, and so will be informationally-richer, yielding the best foundation for subsequent graph-matching operations.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Concept</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Content (IC)</td>
<td>Resnik</td>
<td>IC of the least common subsumer (LCS)</td>
<td>• Uses empirical information from corpora</td>
<td>• Doesn’t use IC of individual concepts (only LCS)</td>
</tr>
<tr>
<td></td>
<td>Jiang and Conrath</td>
<td>Scales LCS by IC of concepts</td>
<td>• Accounts for IC of individual concepts</td>
<td>• WordNet nouns only</td>
</tr>
<tr>
<td></td>
<td>Lin</td>
<td>Scales LCS by IC of concepts</td>
<td>• Accounts for IC of individual concepts</td>
<td>• WordNet nouns only</td>
</tr>
<tr>
<td></td>
<td>Leacock and Chodorow</td>
<td>Finds shortest path between concepts; log smoothing</td>
<td>• Simplicity</td>
<td>• WordNet nouns only</td>
</tr>
<tr>
<td></td>
<td>Wu and Palmer</td>
<td>Path length to subsumer; scaled by subsumers path to root</td>
<td>• Simplicity</td>
<td>• IS-A relations only</td>
</tr>
<tr>
<td></td>
<td>Hirst and St.Onge</td>
<td>Relies on synsets in WordNet</td>
<td>• Measures relatedness of all parts of speech</td>
<td>• WordNet specific</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• More than IS-A relations</td>
<td>• Relies on synsets and relations not available in UMLS</td>
</tr>
</tbody>
</table>
4 State estimation

4.1 Graph matching as a selected discovery technique

Once the data association and graph merging operations are complete, a cumulative, current-time observed data graph exists that is the most compact and efficient representation of the observations on the real world. Comparisons can then be made to questions posed by the data analyst. As noted previously, these questions are converted into graphical representations (“Target Graphs”) and are then compared to the cumulative data graph. To accomplish this task, a method of graph matching needs to be applied. TruST, an efficient graph-matching algorithm developed here at CMIF, is one example of a graph matching algorithm that can be applied to cases such as these [11]. The State Estimation that results here is in the form of a discovered pattern/graph in the observed-data-graph that best matches the target graph or user’s state of interest. The methods employed in our research are in effect “multiple-hypothesis” techniques, as they provide ranked lists of “best” matches for the user to consider.

4.2 TruST

The TruST algorithm utilizes a truncated search tree approach to solve the graph matching problem. TruST is similar to the beam search algorithm. In this algorithm, only the x most promising nodes are explored at each step. This constraint prevents the exponential expansion of the search tree; yet in doing so does not guarantee that an optimal solution will be obtained. The sub-optimal solution is generally quite good, and with manipulation of the parameters of the algorithm, varying levels of efficiency can be acquired.

Zhang [15] in his paper has shown that truncated branch-and-bound gives better results than most of the heuristics. TruST is based on a similar truncated branch-and-bound approach which works on graphs. The search tree is developed dynamically during the search and initially consists of only the root. At each iteration of this algorithm, a sub-problem is selected for exploration from the pool of live subproblems using the scores of the current match. We use a strategy which is similar to the breadth first search strategy found in the literature. The basic principle is to process all the nodes at one level of the search tree before any node at a deeper level.

In what follows, we consider the branching rule for the selected sub-problem. Each sub-problem is developed by adding one pair to its parent problem. Topology is the most important factor considered in this step. Those (template, data graph) node pairs which are qualified to be added should be connected respectively to at least one template and one data graph node in the parent problem. The way in which a newly added template node connects with the existing template node must be exactly the same as the way that connects the data graph nodes in the corresponding pairs. When a new data graph node is added, the score at that level is calculated using the average of the scores of all the node pairs and edge pairs connecting them, at that level. So as the level increases the new nodes are added and the score is revised in correspondence with the new pair score. Further details on the working of TruST algorithm are provided in [11][13][14].

5 Preliminary thoughts on testing

5.1 Test phase

Prior to designing a program that merges nodes and arcs that are associated, it is beneficial to test the effectiveness of data association on graph matching. This analysis is important to ensure that the system output contains viable and accurate results. This experiment can be conducted by comparing the results of various methods of data association with those achieved without it. Another factor that would need to be taken into account during this analysis is how the size of graphs correlates to the effectiveness of merging graphs.

The similarity analysis and methods of scoring will also need to be accounted for during this experiment. For example, through the literature review a study of node-to-node, triple-to-triple, and relation level association could be done to examine effects on graph matching processes. An analysis should also be performed on how choices in similarity scores could contribute to the efficiency of merging nodes and arcs that are associated.

6 Program development

Once an optimal method of association and merging has been found, a program will be developed to implement this process. This program is likely to utilize many of the features implemented within the TruST code, since graph-matching operations also involve consideration of graph similarity. Given that TruST has been written using C++ and Java, it is expected that these languages will also be used to build the data association program.

After the program has been created, it will be necessary to test and validate the system using existing data. This data could be retrieved from several different sources. There will be an official data set referred to as “HASTEN-1” which is expected to come from the US Army Research Office. Additional sources include a 100 and 220 message data set existing from previous research; all of these data are from COIN problems. When the program has been validated, the size of the graphs and the quality of the matches can be compared to a model that has not used data association. This will demonstrate how the application of this program can improve system performance.
6.1 Example results

While no calculations have actually been performed yet, an ideal result in a simple form will be described in this section. Assume a message, Message 1, comes in that says “John Smith met with Frank Jones on Monday January 5, 2010”. The original message is sent through a text extractor which forms a triple that contains additional metadata. This triple can be seen in figure 2.

Figure 2. Example Message 1: John Smith met with Frank Jones on Monday January 5, 2010.

Assume a second message arrives, Message 2, says “Jon met with Frank on Monday at 5:00pm”. Once the triple has been extracted and metadata has been added, it can be shown in the following graphical nature (figure 3).

Figure 3. Example Message 2: Jon met with Frank on Monday at 5:00pm.

Assuming that the TruST algorithm enhancements were implemented into the program, the two messages illustrated above would be tested for similarity. Provided that node A and node C yielded the best match that met the given threshold level, they may be merged to form a single node combining several of the attribute features from both nodes. The same method can be done with node B and node D. Further comparisons can be made on the arcs that connect these nodes. Figure 4 demonstrates how a triple would appear after merging the two messages have been completed.

Figure 4. Merged Example: John Smith met with Frank Jones on Monday January 5, 2010 at 5:00pm

This example illustrates the issue of the loss of information due to graph merging. For the example above, when forming node AC it was arbitrarily chosen which attribute would be picked. For instance, the first name provided in node A was John while the first name in node C was Jon. Although the nodes met some threshold to be merged, there is still a possibility that the nodes were actually different. Once node AC is developed, there is a risk of losing important information that may be relevant in the future. This type of problem may occur in real life due to commonalities between entities.

Another important concern that is presented with merging graphs together is the error that could be produced. For example, if John and Jon were not the same individual, the merged node AC may assign John an eye color of blue when in fact that may not be the case. Additionally, there may be error when comparing less descriptive attributes. This can be seen with node A having a height of 6’1” and node C having a fuzzy height of tall. While the uncertainty comparisons and association values will account for this, a possibility for incorrect merging still exists.

7 Conclusion

In the field of Information Fusion, it is often said that Data Association is at the heart of the process, as it is the function that allocates the available observational data to state estimation processes. For traditional electronic sensor-based systems, there have been many advancements made toward achieving association algorithm efficiency and effectiveness. However, the formulation of a process design and the specification of algorithmic details for the implementation of a Data Association process for soft data within the area of counterinsurgency results in a variety of challenges as described herein. Decisions on batching logic, methods of graph searching and merging and proper semantic similarity metrics need to be made. Additionally, an appropriate method to solve the multi-dimensional assignment problem, as well as a technique to merge graph features and attributes together will need to be found. Once these issues have been resolved, the analyst will be presented with an efficient, comprehensive, and condensed representation of the real world from which to make real-time decisions. Finally, these operations are computationally complex, and even if workable and effective designs can be found, computationally efficient implementations will also need to be developed. As workable solutions evolve, the fusion community will be able to provide automated support to analysts and decision-makers in these critically-important applications.

Acknowledgement

We gratefully acknowledge that this research activity is supported by a Multidisciplinary University Research Initiative (MURI) grant (Number W911NF-09-1-0392) for “Unified Research on Network-based Hard/Soft Information Fusion”, issued by the US Army Research Office (ARO) under the program management of Dr. John Lavery.
References


