Visual Analytics for Supporting Entity Relationship Discovery on Text Data

Hanbo Dai[1] , Ee-Peng Lim[1], Hady Wirawan Lauw[1], and Hweehwa Pang[2]

[1] School of Computer Engineering, Nanyang Technological University
[2] School of Information Systems, Singapore Management University

Abstract. To conduct content analysis over text data, one may look out for important named objects and entities that refer to real world instances, synthesizing them into knowledge relevant to a given information seeking task. In this paper, we introduce a visual analytics tool called **ER-Explorer** to support such an analysis task. ER-Explorer consists of a data model known as **TUBE** and a set of data manipulation operations specially designed for examining entities and relationships in text. As part of TUBE, a set of interestingness measures is defined to help exploring entities and their relationships. We illustrate the use of ER-Explorer in performing the task of finding associations between two given entities over a text data collection.

1 Introduction

1.1 Motivation

Information synthesis and analysis can be facilitated by a visual interface designed to support analytical processing and reasoning. Such an interactive visualization approach is also known as **visual analytics**[1]. In this research, we specifically focus on designing and implementing a visual analytics system to support the entity relationship discovery task that involves identifying entities and relationships from a document or a collection of documents so as to create a network of entities that are relevant to an entity relationship discovery task.

Consider the task of finding the person and organization entities that connect two terrorists from a given document collection. A domain expert will need an interactive visual tool to help in extracting entities from the documents and the relationships among these entities, judging the relevance of these entities and relationships by checking them up in documents containing them, and selecting the relevant ones to be included in the results.

For a visual analytics system to support the above retrieval task, the following system features are required.

- *Network representation of information*: Entity and relationship instances are best represented using a graph or network, especially when path and connectivity properties of these instances are to be studied and visualized along with the documents containing them.

C.C. Yang et al. (Eds.): ISI 2008 Workshops, LNCS 5075, pp. 183–194, 2008.

- *Interactive refinement of results*: The above retrieval task, like many others that require expert judgement, will involve much user interaction in multiple iterations. Hence the visual analytics system will have to incorporate user operations that may include or exclude entities and relationships from the retrieval results.
- *Intelligent user assistance*: Given the possibly large volume of document data and many entity and relationship instances embedded in documents, users will expect some intelligent assistance from the visual analytics system to help them gain more insight into the data. The exact form of assistance may very much depend on the task at hand. For example, entities (or relationships) may have to be ranked by their closeness to the two given terrorists so as to help user decision making.

The above are also the system features that distinguish visual analytic systems from the other visual interface systems for analyzing networks of entity and relationship instances. In social network analysis, the state-of-art visual interface systems often assume that networks of entity and relationship instances have already been identified and verified, as well as can be studied separately from the documents containing them[2,6,7]. This assumption clearly does not hold for documents which are not pre-annotated. Even if the documents are already pre-annotated, it is still challenging to determine the relevant entity and relationship instances. This often requires users to interpret text content in documents containing these instances.

1.2 Research Objectives and Contributions

In this research, we therefore aim to design a visual analytic framework for entity relationship discovery under the assumption that (a) user judgement on document content is required for identifying relevant entity and relationship instances, and (b) the discovery is an iterative process with user involvement.

Our contribution in this paper can be summarized as follows:

- We present a visual analytics framework for discovering a network of related entities found in text data. This framework consists mainly of a multi-dimensional data model and a visual interface tool for representing and manipulating entity and relationship instances.
- We design a text cube representation of the entity and relationship instances in document data. This representation, known as **TUBE**, supports semantic entity types, conceptual entity representation, inter-entity relationships and other data constructs useful for information analysis and synthesis.
- We develop a visual analytics system known as **ER-Explorer** to realize a set of user operations on a network of entities derived from a set of text documents so as to conduct entity relationship discovery.
- We illustrate our visual analytics system prototype using a case study where the entities and relationships linking two given entities can be discovered through an interactive process.

1.3 Paper Organization

We organize the rest of the paper as follows. In Section 2, we cover the related research. In Section 3, our framework for entity relationship discovery using visual analytics is presented. In Section 4, we describe the **ER-Explorer**, a visualization tool implemented based on our proposed framework. This is followed by a case study analysis in Section 5. We finally conclude the paper in Section 6.

2 Related Work

Visually analyzing social networks has been receiving growing attention and several visualization tools have been developed for this purpose. *Vister*[3] provides an environment to explore and analyze online social network, supporting automatical identification and visualization of connections and community structures. *SocialAction*[4] allows users to explore different social network analysis measures to gain insights into the network properties, to filter nodes (representing entities), and to find outliers. Users can interactively aggregate nodes to reduce complexity, find cohesive subgroups, and focus on communities of interest. However, the measures used in these systems are topological-oriented.

Xu and Chen [8] proposed a framework for automatic network analysis and visualization. Their *CrimeNet Explorer* identifies relationships between persons based on frequency of co-occurrence in crime incident summaries. Hierarchy clustering algorithm is then applied to partition the network based on relational strength.

The above systems while supporting network visualization, lack the measures for discovering associations among nodes. Their way of grouping entities is based on centrality measure or relational strength, which does not allow user judgement and may fail to group semantically identical entities.

A visual analytic system *Jigsaw*[9] represents documents and their entities visually in multiple views to illustrate connections between entities across the different documents. It takes an incremental approach to suggest relevant reports to examine next by inspecting the co-occurred entities. However, it does not use measures other than frequency of entities in documents. When the list of co-occurred entities becomes very large, it would be quite cumbersome for an analyst to find the interesting entities or documents, since considering the frequency measure alone may be restrictive. Moreover, in cases where co-occurrence relationship between entity are not semantically meaningful, the analytics ability of Jigsaw will be ineffective.

There is much research literature on path finding. *Transitive association discovery* was proposed to detect conceptual association graph in a text dataset[10]. Interestingness measures based on co-occurrence are designed. A dynamic programming algorithm was developed to compute interesting paths of various lengths from source to target entities. Document contexts of the paths are also provided. People association finding in the *ArnetMiner* project[11] also aims to detect the good associations. Since the above approaches rely on algorithms to

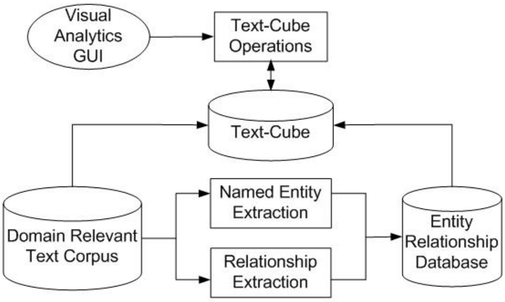


Fig. 1. System Architecture

compute paths, users have no control over the paths and entities he or she may want to explore. Moreover, semantically identical entities are also not considered.

3 Framework of Visual Entity Relationship Discovery from Text Data

3.1 Architecture of Visual Analytics Tool

As shown in Figure 1, our proposed visual analytics system architecture consists of a domain relevant text corpus on which the entity relationship discovery task is to be performed. From the text corpus, named entities and relationships will be extracted into an *entity relationship database*. A *text cube database* is then constructed from the entities and relationships. It consists of one or more text cube instances and each text cube instance is a multidimensional table with entities as dimension values and relationships as cells. Unlike a database table, a text cube also provides document evidence of the entities and relationships so as to facilitate cross checking entities and relationships with their text sources. A set of *text cube operations* are provided to manipulate the content of text cubes. These are also the data operations to be invoked by the *visual analytics GUI tool*. Users will interact with the GUI tool performing visual analysis operations on the text data without having to know the underlying text cube representations and operations.

3.2 TUBE: Text Data Cube Model and Operations

TUBE Representation. In our TUBE model, a domain relevant document collection $\mathcal{D}$ provides the raw data for analysis[12]. For each document $d \in \mathcal{D}$, the set of named objects extracted from d is denoted by $A(d)$. The entire set of named objects extracted from $\mathcal{D}$ is denoted by $A(\mathcal{D}) = \bigcup_{d \in \mathcal{D}} A(d)$. We define a mapping function $f_d : A(\mathcal{D}) \rightarrow 2^{\mathcal{D}}$ to map from a named object to its supporting document set, consisting of documents that contain the named object.

In TUBE, we introduce the notion of entity e, which is defined as a named object or a set of other entities as follows:

$$e = \begin{cases} a, & a \in A(\mathcal{D}) \\ \{e_1, e_2, \ldots, e_n\}, & e_i \textit{ is an entity.} \end{cases}$$

We say a is a component of e, $a \prec e$, if $a = e$ or $\exists e_i \in e$ s.t. $a \prec e_i$. e is said to be a *conceptual entity* if it is not a named object. The *document evidence* of e is defined as $f_d(e) = \bigcup_{a \prec e} f_d(a)$.

We now define a n-dimensional TUBE as a tuple $T = \langle S, B, M, D \rangle$. S represents the *schema* and $S = \{s_1, s_2, \ldots, s_n\}$ where s_i denotes the list of entities of dimension i. B is a *mask* with 0 or 1 values. $M = \{m_1, m_2, \ldots, m_{|M|}\}$ is a set of measures. Each m_j is associated with a measure function $mf_j()$. D represents a document collection and $D \subseteq \mathcal{D}$.

The TUBE T has $|s_1| \times |s_2| \times \ldots \times |s_n|$ cells. Each cell is denoted by c $(e_1, e_2, \ldots, e_n)$ where $e_i \in s_i$ for $1 \leq i \leq n$. Without causing any ambiguity, we may use c to denote a cell. A cell c is said to be *present* if $B(c) = 1$ or *hidden* if $B(c) = 0$. The document evidence of c is defined by $f_d(c) = \bigcap_{i=1}^{n} f_d(e_i)$. When $f_d(c)$ is not empty, we say that $e_1, \ldots, e_n$ *co-occur*. This *co-occurrence relationship* can be represented by c We also define the *named object set* of c as $A(c) = \bigcup_{i=1}^{n} \bigcup_{a \prec e_i} \{a\}$. The *support value* for a d_k in $f_d(c)$ with respect to c is defined by:

$$Sup(c, d_k) = \textstyle\sum_{a \in A(c)} tf_{d_k,a} \times idf_a$$

where $tf_{d_k,a}$ is the a's frequency in d_k

$$idf_a = \tfrac{|\mathcal{D}|}{|f_d(a)|}$$

Given a cell c, c has a measure value $c.m_j = mf_j(c)$ derived by applying the measure function mf_j.

TUBE Operations. We have also designed a set of operations on TUBE. Given a TUBE instance $T = \langle S, B, M, D \rangle$,

- Insert operation adds an entity to a selected dimension.
- Remove operation removes an existing entity from a dimension.
- SelectCell operation assigns 0 or 1 to a specified entry in B which corresponds to a cell in T.
- Cluster operation groups a subset of entities in a specified dimension into a new conceptual entity and add this conceptual entity to that dimension.

TUBE Instances For Entity Relationship Discovery. Our entity relationship discovery uses two TUBE instances T_1 and T_2. $T_1 = \langle S^1, B^1, M^1, D \rangle$ and $T_2 = \langle S^2, B^2, M^2, D \rangle$ are 1-D and 2-D TUBE instances respectively. We initialize T_1 to have $S^1 = \{s_1^1\}$, $s_1^1 = A(\mathcal{D})$ by Insert operation. In other words, T_1 has a dimension consisting of all named objects. T_2 is initialized to have $S^2 = \{s_1^2, s_2^2\}$ where $s_1^2 = s_2^2 = A(\mathcal{D})$. In other words, T_1 is designed to maintain information about named objects and T_2 for information about the relationships of pairs of entities. Also note that any operations on one dimension of T_2 will affect the other dimension the same way.

The masks B^1 and B^2 are initialized to return 0's for all cells, making all named objects and relationships initially hidden from the network view of our visual tool.

3.3 Entity Relationships Exploration Using T_1 and T_2

Given two entities of interest known as *source entity* (s) and *target entity (t)*, a typical entity relationship discovery task would be finding interesting paths between them. Each path denoted by $e_1 \leftrightarrow \ldots \leftrightarrow e_p$ represents a chain of relationships. Each relationship denoted as $e_{i-1} \leftrightarrow e_i$, for $1 < i \leq p$, and e_1 and e_p are entities semantically equivalent to s and t respectively. Note that in this task, the relationships are non-directional. The roles of source and target are therefore exchangeable. Nevertheless, we just distinguish them for easy discussion.

Our visual tool can incrementally add named objects and relationships into the entity network presentation window as nodes and edges respectively by invoking TUBE operations on the two TUBE instances T_1 and T_2, To display an entity in the visual tool, we set the respective cell in T to have $B = 1$. To display a relationship, we set the corresponding cell in T_2 to have $B = 1$. Hiding entities and relationships can be performed in a similar way. This interactive approach to construct entity networks can be assisted by interestingness measures defined for the entity relationship discovery task.

3.4 Interestingness Measures for Entity Relationship Discovery

In this section, we define several measures to be used in T_1 and T_2 to support entity relationship discovery. For T_1, there is only one measure, i.e., $M^1 = \{m_{path\_strength}\}$. For T_2, we define $M^2 = \{m_{name\_sim}, m_{strength}, m_{d\_entity}\}$.

- $m_{path\_strength}$: the length of shortest path(s) between s and t going through an entity (a named object, since it is defined on T_1).
- m_{name_sim}: the similarity score between two entity names.
- $m_{strength}$: the relationship strength between two entities.
- m_{d_entity}: the dominance of one entity over another.

Given a cell $c(e_i, e_j)$ in T_2,

$$mf_{name\_sim}(c) = Avg_{a_u \prec e_i, a_v \prec e_j}\ NameSimilarity(a_u, a_v)$$

where *NameSimilarity* is a name comparison function that returns a value between 0 (unrelated name objects) and 1 (synonym). If e_i and e_j are conceptual entities, the measure value returned is an average of over name similarities between named objects of e_i and e_j. With this measure, we now derive a set of synonyms for an entity e_i, as denoted by

$$Synonym(e_i) = \{e_j | mf_{name\_sim}(c(e_i, e_j)) > \lambda\}.$$

The synonym entities of e_i are entities whose names are within λ edit distance from that of e_i. The function *Synonym* is helpful to detect different spellings of an entity. Grouping synonym entities together may discover new associations, since they may have different relationships with other entities.

The measure function of $m_{strength}$ for a cell $c(e_i, e_j)$ is denoted by $mf_{strength}$ $(c(e_i, e_j))$ which computes strength using *Dice Coefficient*, i.e.,

$$mf_{strength}(c(e_i, e_j)) = \log\left(1 + 2 \cdot \frac{|f_d(c(e_i,e_j))|}{|f_d(e_i)+|f_d(e_j)|}\right)$$

The strength of a cell representing a pair of entities captures the likelihood of a relationship between them. The more documents they co-occur in, the higher the strength.

Given two entities e_i and e_j, the *d_entity* measure determines if the documents containing e_i are also those containing e_i and e_j. This happens when e_i always appears together with e_j(This implies whenever e_i appears, e_j is always there), and we say that e_j *dominates* over e_i.

$$m_{d\_entity}(c(e_i, e_j)) = \begin{cases} 1 \; if \; f_d(c(e_i)) = f_d(c(e_i, e_j)) \\ 0 \text{ otherwise} \end{cases}$$

For example, m_{d_entity}(“9-11”,“New York”) = 1 when “9-11” appears in only those documents containing both “9-11” and “New York”.

For T_1, the measure $m_{path\_strength}(c(e_i))$ returns the strength of shortest path(s) between s and t going through e_i. Let $s\_path(e_i)$ denote this set of shortest paths, $m_{path\_strength}$ is defined as:

$$m_{path\_strength}(c(e_i)) = Max_{p_{ik} \in s\_path(e_i)} strength(p_{ik})$$

where

$$strength(p_{ik}) = \prod_{(c(e_x,e_y) \in p_ik} mf_{strength}(c(e_x, e_y))$$

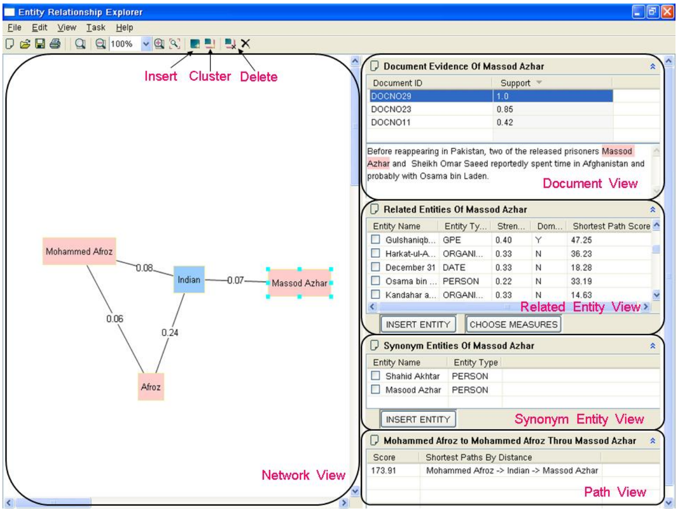
When multiple shortest paths between s and t pass through entity e_i, $m_{path\_strength}(e_i)$ will take the maximum path strength among them. A large $m_{path\_strength}(e_i)$ suggests that there exists a path with edges that represent strong relationships. Hence, e_i may be a good entity to explore to establish useful linkages between s and t.

4 Visual Analytics Tool for Entity Relationship Discovery

In this section, we describe our Visual Analytics Tool, **ER-Explorer** (Entity Relationship Explorer) can be utilized. The named entity extraction in our system is performed by **BBN Identifinder** [13], which can extract entities of 24 types including *person*, *organization*, *GPE* (Geo-political entities), *date* and others. After extracting co-occurrence relationship extraction, *Lucene* is used to index all documents by their extracted named entities. The visualization part of our tool is built upon *Chisio*[1], a free Compound or Hierarchical Graph Visualization Tool based on eclipse Graphical Editing Framework.

Overview of User Interface
ER-Explorer is mainly made up of five views (see Figure 2), namely, a *Network View*, a *Document View*, a *Related Entity View*, a *Synonym Entity View* and a *Path View.*

[1] http://www.cs.bilkent.edu.tr/ ivis/chisio.html



Document Evidence Of Massod Azhar

Document ID	Support
DOCNO29	1.0
DOCNO23	0.85
DOCNO11	0.42

Before reappearing in Pakistan, two of the released prisoners Massod Azhar and Sheikh Omar Saeed reportedly spent time in Afghanistan and probably with Osama bin Laden.

Related Entities Of Massod Azhar

Entity Name	Entity Ty...	Stren...	Dom...	Shortest Path Score
□ Gulshaniqb...	GPE	0.40	Y	47.25
□ Harkat-ul-A...	ORGANI...	0.33	N	36.23
□ December 31	DATE	0.33	N	18.28
□ Osama bin ...	PERSON	0.22	N	33.19
□ Kandahar a...	ORGANI...	0.33	N	14.63

Synonym Entities Of Massod Azhar

Entity Name	Entity Type
□ Shahid Akhtar	PERSON
□ Masood Azhar	PERSON

Mohammed Afroz to Mohammed Afroz Throu Massod Azhar

Score	Shortest Paths By Distance
173.91	Mohammed Afroz -> Indian -> Massod Azhar

Fig. 2. ER-Explorer Interface

Network View is where the user visualizes the network and manipulates it with visual analytics operations. We visually display entities in TUBE as nodes and relationships as edges. Entities are shown as boxes in different colors associated with their entity types. Edges are weighted by the $m_{strength}$ of T_2. Each conceptual entity is visualized as a compound box drawn to enclose its component entities.

Document View shows the supporting documents of a selected entity relationship. These can be read by users to understand the context of entities and relationships. This view consists of two parts. The upper part lists the document IDs and their support values with respect to selected entity/relationship. The lower part displays the content of a document, once that document is selected. All named objects in the document semantically represented by the selected entity/relationship are highlighted.

When an entity/relationship is selected in the Network View, the **Related Entity View** displays all entities co-occurred with the entity selected or entities of the selected relationship. The *co-occurring entities* of an entity e_i is defined as $e_i.CoEntSet = \{a_j | a_j \in A(f_d(e_i)), a_j \neq e_i\}$. Co-occurring entities are shown with measures chosen by users using the "CHOOSE MEASURES" button. These measures includes $m_{strength}$, m_{d_entity} from T_2, and $m_{path\_strength}$ from T_1. A co-occurring entity can be added to the Network View by using the "INSERT ENTITY" button. When no entity/relationship is selected, this view lists all entities in the Network view with values of $m_{path\_strength}$ from T_1.

When an entity is selected in the Network View, this view displays synonym entities derived from T_2. A "INSERT ENTITY" button is also provided to add synonym entities into the Network View. When an relationship in the Network View is selected, the Synonym Entity View will be empty.

Path View displays shortest path(s) linking the source entity and the target entity. When an entity is selected in the Network View, It lists all shortest paths through this selected entity. When no entity is selected, this view displays all shortest paths through all entities in the Network view. When a relationship is selected, this view will be empty.

4.1 Visual Analytic Operations

The visual analytics operations including *Insert*, *Delete* and *Cluster* visually implements TUBE operations. Other operations supporting visualization requirements including highlighting, zooming, dragging are also provided. These visual analytic operations can be found on the toolbar and in the Edit menu of ER-Explorer.

The visual analytics operation *Insert* corresponds to $\mathsf{SelectCell}$ in T_1 and T_2. Suppose a user inserts an entity e, the mask value will be changed by setting $B^1(c(e)) = 1$ in T_1. As for the mask value in T_2, we set $B^2(c(e, e_i)) = 1$, where $c(e_i) = 1$ in T_1. This reveals all relationships this entity has with all entities in the Network View.

ER-Explorer provides two ways of inserting an entity. One is using the "INSERT ENTITY" button in the Related Entity View and the Synonym Entity View. The other is utilizing the Insert button on the toolbar, which opens a window where all entities existing in the dataset can be retrieved and inserted. This is helpful when a user knows some entity of interest but does not know where to find it in any Views.

The *Delete* operation on a node representing a named object a corresponds to $\mathsf{SelectCell}$ operation on T_1. The mask value in T_1 will be changed by setting $B^1(c(e)) = 0$, which visually removes this node from the Network View. However, the same operation on a node representing a conceptual entity e corresponds to Remove in T_1 and T_2. T_2 will be changed by $S_1^2 = S_1^2 - \{e\}$, $B^2(c(e_i, e_j)) = 1$, where $e_i \in e,\ c(e_j) = 1$. The schema part of T_1 will also be changed by $S_1^1 = S_1^1 - \{e\}$. As a result, the conceptual entity is decomposed and its elements are displayed along with their edges connecting entities in the Network View. The *Delete* operation on a relationship $c(e_i, e_j)$ corresponds to the $\mathsf{SelectCell}$ operation on T_2. $B^2(c(e_i, e_j)) = 0$, which visually hides this edge.

The *Cluster* operation corresponds to $\mathsf{Cluster}$ in T_1 and T_2. Given a new conceptual entity e created by this operation, T_2 will be changed as $S_1^2 = S_1^2 \cup \{e\}$, $B^2(c(e, e_i)) = 1\ and\ B^2(c(e_k, e_i)) = 0$, where $c(e_i) = 1$ in T_1, $e_k \in e$. T_1 is changed as $S_1^1 = S_1^1 \cup \{e\}$. To use *Cluster*, a user first selects the intended entities in Network View for grouping. He/she then clicks on the cluster button. Visually, all selected nodes are framed by a box representing the new conceptual entity, which can be renamed for easy reference. After this, all edges linking to the selected entities are replaced by edges linking the new conceptual entity and other entities.

5 Case Study

To demonstrate how our ER-Explorer can help to discover entities and relationships that are relevant to association between two given entities, we describe a case study where it is used to find the linkage between two terrorists, Mohammed Afroz and Massod Azhar from the **IC814** dataset. The dataset was derived from a report titled "The Hijacking of IC-814: Al Qaeda, Taliban and Pakistani Factors" which gives a detailed description and analysis about the hijacking of the Indian aircraft IC-814, a well known terrorist incident in year 1999. We extracted entities of types *person*, *organization*, *event*, *GPE*, *product* and *date* as they are more relevant to our discovery task. We then extracted relationships by identifying sentences containing at least two named entities and considered each sentence as a document.

We now describe the process a user will be involved to derive the entity network shown in Figure 3. The user may begin the entity relationship discovery task by first adding the two entities "Mohammed Afroz" and "Massod Azhar" into the Network View. With no other entity selected, the user will see a list of shortest paths between source and target nodes in the Path View. Suppose the user notices that the path $MohammedAfroz \leftrightarrow Indian \leftrightarrow MassodAzhar$ which suggests the two entities are somehow linked by "Indian". She adds "Indian" into the Network View. Next, the user may refer to Related Entity View as she selects "Mohammed Afroz" in the Network View. The Related Entity View shows a list of candidate entities sorted by interestingness measures including $m_{strength}$, m_{d_entity} and $m_{path\_strength}$. The entity "Afroz", a high $m_{strength}$ value in the view looks very similar to "Mohammed Afroz". It may then be inserted into the Network View.

As "Indian" and "Afroz" get inserted into the Network View, several new edges between them also show up in the view. In order to understand the relationships in these edges, the user refers to the Document View of each edge. She may find the only document containing both "Mohammed Afroz" and "Indian" in the sentence "After the confession of Mohammed Afroz was made public by a statement of the Indian minister" which does not imply any meaningful relationships. Hence, the corresponding edge linking the two entities is deleted. The user can also find out that "Afroz" and "Mohammed Afroz" refer to the same person. She therefore uses the *Cluster* operation to group them together and names the new conceptual entity as "C_Mohammed Afroz".

The user subsequently uses the Related Entity View and Path View to explore other entities co-occuring with "C_Mohammed Afroz" or linked to it by shortest paths. She subsequently inserted "Sandy", "Osama bin Laden", and "SIMI" into the Network View. She will also find "India" as a synonym of "Indian" and group them into a conceptual entity "C_India". By reading the document containing "Sandy", she can also find that the latter is one of the hijackers and has an alias "Shahid Akhtar Sayeed". "Shahid Akhtar Sayeed" is then inserted into the Network View. The Synonym Entity View also suggests "Shahid Akhtar Sayed" as another similar entity. Subsequent document verification concludes that they are the same and are grouped into the conceptual entity "C_Shahid

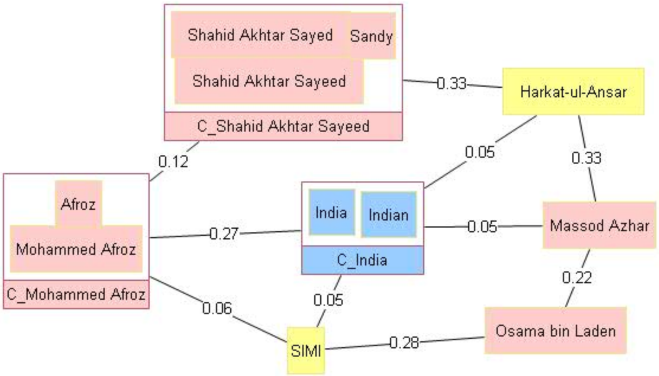


Fig. 3. The Result Network of Our Case Study

Akhtar Sayeed". After checking the supporting document of "C_Shahid Akhtar Sayeed" and "Massod Azhar", the user may find out that the two entities are indirectly linked by "Harkat-ul-Ansar", an organization.

At this point, several entities and relationships have been found while the semantics of the links among them can be summarized in three story threads between Mohammed Afroz and Massod Azhar. The first involves Mohammed Afroz's trainning sponsored by SIMI group, which has a close relation with Osama bin Laden. The latter has ever spent some time with Massod Azhar. The second conveys the information that Mohammed Afroz was active in several places in India and was also arrested there, and so was Massod Azhar. The third says that Mohammed Afroz was trained as a pilot together with Shahid Akhtar Sayeed, who is a member of Harkat-ul-Ansar organization, of which Massod Azhar was the general secretary.

6 Conclusions

In this paper, we propose an interactive visual approach to discover entity and relationships embedded in text data. We have developed a visual analytics tool called ER-Explorer which is equipped with a versatile data model known as TUBE to manipulate entity and relationship information and their supporting documents. We have demonstrated its capability on a hijacking event dataset to discover relationships between two terrorists. For our future research, we plan to extend ER-Explorer to discover associations between more than two entities and to automate some of the exploration subtasks through some tunable parameters. We are also interested to study how concise textual summary of the constructed entity network can be generated from the supporting documents for easy reading.

Acknowledgments

This work was supported in part by A*STAR Public Sector R&D, Project Number 062 101 0031.

References

1. Thomas, J., Cook, K.: A Visual Analytics Agenda. IEEE Computer Graphics and Applications 26(1), 10–13 (2006)
2. Shen, Z., Ma, K.-L., Eliassi-Rad, T.: Visual Analysis of Large Heterogeneous Social Networks by Semantic and Structural Abstraction. IEEE Transactions on Visualization and Computer Graphics 12(6), 1427–1439 (2006)
3. Jeffrey Heer, D.B.: Vizster: Visualizing Online Social Networks. In: Proceedings of the IEEE Symposium on Information Visualization (October 2005)
4. Adam Perer, B.S.: Balancing Systematic and Flexible Exploration of Social Networks. IEEE Transactions on Visualization and Computer Graphics 12(5), 693–700 (2006)
5. Krebs, V.: Mapping networks of terrorist cells. Connections: the Journal of the International Network of Social Network Analysts 24(3), 43–52 (2002)
6. Bilgic, M., Licamele, L., Getoor, L., Shneiderman, B.: D-Dupe: An Interactive Tool for Entity Resolution in Social Networks. In: Proceedings of the IEEE Symposium on Visual Analytics Science And Technology, October 2006, pp. 43–50 (2006)
7. Yang, C.C., Liu, N., Sageman, M.: Analyzing the Terrorist Social Networks with Visualization Tools. In: Proceedings of the IEEE International Conference on Intelligence and Security Informatics (May 2006)
8. Xu, J., Chen, H.: CrimeNet Explorer: A Framework for Criminal Network Knowledge Discovery. ACM Transactions on Information Systems 23(2), 201–226 (2005)
9. Stasko, J., Gorg, C., Liu, Z., Singhal, K.: Jigsaw: Supporting Investigative Analysis through Interactive Visualization. In: Proceedings of the IEEE Symposium on Visual Analytics Science And Technology, October 2007, pp. 131–138 (2007)
10. Jin, W., Srihari, R.K., Wu, X.: Mining Concept Associations for Knowledge Discovery Through Concept Chain Queries. In: Proceedings of the Pacific Asia Conference on Knowledge Discovery and Data Mining (April 2007)
11. Tang, J., Zhang, J., Zhang, D., Yao, L., Zhu, C.: ArnetMiner: An Expertise Oriented Search System for Web Community. In: Proceedings of the 6th International Conference of Semantic Web (November 2007)
12. Lauw, H.W., Lim, E.-P., Pang, H.: TUBE (TextcUBE) for Discovering Documentary Evidence of Associations among Entities. In: Proceedings of the ACM Symposium of Applied Computing (March 2007)
13. Bikel, D., Schwartz, R., Weischedel, R.M.: An Algorithm that Learns What's in a Name. Machine Learning 34(1-3), 211–231 (1999)