Transforming Open-Source Documents to Terror Networks: The Arizona TerrorNet

Daniel M. McDonald, Hsinchun Chen, and Robert P. Schumaker

Artificial Intelligence Lab, University of Arizona
Department of Management Information Systems
1130 East Helen Street Tucson, AZ 85721
dmm,hchen,rschumak@eller.arizona.edu

Abstract

Homeland security researchers and analysts more than ever must process large volumes of textual information. Information extraction techniques have been proposed to help alleviate the burden of information overload. Information extraction techniques, however, require re-training and/or knowledge re-engineering when document types vary as in the homeland security domain. Also, while effectively reducing the volume of the information, information extraction techniques do not point researchers to unanticipated interesting relationships identified within the text. We present the Arizona TerrorNet, a system that utilizes less specified information extraction rules to extract less choreographed relationships between known terrorists. Extracted relations are combined in a network and visualized using a network visualizer. We processed 200 unseen documents using the TerrorNet which extracted over 200 relationships between known terrorists. An Al Qaeda network expert made a preliminary inspection of the network and confirmed many of the network links.

Introduction

Terrorism researchers and analysts must process large amounts of news and other journalistic data to understand terror networks and analyze events. However, the amount of information published by national and international journalistic sources is quickly outstripping the boundaries of manual effort. While basic searching tools are helpful, there is a lack of automated document analysis that is tailored to tasks performed by terrorism researchers. Recently homeland security researchers have benefited from understanding the relationships between terrorists and their cells (Sageman 2004). Information extraction techniques, however, have typically focused on fact finding and filling event templates as opposed to extracting terrorist networks.

In addition, relevant information about terrorism is spread in varied genre over the Internet. For example, sources of information vary between personal interviews with suspected terrorists to the 9/11 commission reports and domestic and foreign court transcripts. The varied sources of information reviewed by homeland security researchers and analysts would create a need for greater knowledge engineering efforts and/or algorithm training to create information extraction tools. This need would be even greater for the MUC-style template filling extraction techniques that are highly specified with lexical constraints. The requirement to re-train algorithms for every type and topic of document dims the promise of information extraction for reducing information overload.

We present the Arizona TerrorNet, a tool which automatically extracts terror networks from varied types of documents. The AZ TerrorNet utilizes less specified information extraction techniques than those found in systems participating in the Message Understanding Conference template tasks. The lower level of specification has two benefits. First, a greater variety of less anticipated relations can be extracted from the text. Once aggregated, relations can potentially reveal less obvious and more interesting relations. Second, the reduced use of specific lexical constraints in the Arizona TerrorNet makes it able to process a greater variety of document types from a greater number of domains. As a result, the costs associated with knowledge re-engineering are reduced.

In this paper we first present some research background in the field of information extraction (IE). The focus of the review is on different IE techniques that have different needs for knowledge re-engineering given new document types and domains. Next, we present the collection of information extraction algorithms utilized in the Arizona TerrorNet. We propose that the collection of algorithms and the aggregation process can discover relationships that are not entirely obvious in the text and thus are more interesting. We then briefly present the result of parsing 200 previously unseen documents from various International news sources. A subset of the relations is shown and comments from a terrorist network expert are included.

Background

Missed information and subsequent lost opportunities have been reported as previous homeland security shortcomings (Kean, Hamilton et al. 2004). Information extraction has been proposed as a method to manage vast amounts of information and thus avoid lost opportunities (Cowie and Lehnert 1996). Information extraction techniques attempt to whittle away large amounts of text leaving only the most relevant pieces of information in a structured format. The resulting information in its structured form can then be fed into databases for future analysis and mining.

Most previous information extraction tasks have focused primarily on incident and entity extraction, text summarization (Mani, House et al. 1998), and filling scenario templates (DARPA 1998). We are aware of one information extraction project where relationships between people were extracted. In the early 1980's, Zarri worked on extracting semantic relationships between French historical figures from relevant texts (Zarri 1983).

Using information extraction techniques in hopes of assisting national security effort, however, is not new. Information extraction research has been aided largely by seven Message Understanding Conferences (MUC) funded by DARPA. In the MUC-3 and MUC-4, newspaper and newswire texts were used that contained information about terrorist activities in Latin America (DARPA 1991; DARPA 1992). Templates were created for different incidents such as arson and kidnapping. Within each template, relevant information to the incident would be added such as the incident's date, location, and incident type. Use of the templates allowed precision and recall performance evaluations to be compared to human performance.

However information extraction algorithms are often designed to extract only specific information to fill templates. As users' tasks change over time, the need to engineer new information extraction routines is inevitable. This process of having to re-create information extraction routines to suit new domains and tasks is referred to as the knowledge engineering bottleneck (Cowie and Lehnert 1996). It is generally thought that the knowledge engineering bottleneck is the greatest impediment to more widespread adoption of information extraction (Ciravegna 2001). Research efforts in information extraction have attempted to reduce this bottleneck by lowering the knowledge engineering cost. If algorithms can capture "shallow knowledge" automatically, then knowledge engineering has no significant cost. With no significant cost, the knowledge invested in the extraction routines does not necessarily have to be reused to be justified. Besides work on automatically extracting shallow knowledge, work has been done trying to maximize the use of end-users to train systems that can learn rules behind the scenes (Ciravegna and Petrelli 2001). This line of research recognizes the cost of corpus annotation and tries to facilitate maximum end-user participation. We now separate our review into systems that are highly specified and those that are more portable between tasks.

Highly-Specified IE Template Systems

As mentioned, information extraction systems are engineered to fill the slots of well-defined templates. Systems have often relied on specific lexical semantic patterns in sentences to guide the extraction of template information. Autoslog (Riloff 1993), for example, utilized a trigger slot that was filled with a verb such as "bombed" or "robbed" to activate extraction. Other systems such as Liep (Huffman 1995), Palka (Kim and Moldovan 1995), and Crystal (Soderland, Fisher et al. 1995) also utilized an exact word or verb root constraint as part of the extraction process. While utilizing such narrow rules became a necessity to achieve high performance on the highly specified template filling tasks in MUC, such rules and template tasks in general have some drawbacks. First, the information that is extracted is not in a format that can be utilized for any but the intended task, which might be an alerting system and/or a historical record. In addition, there is no text mining or discovery taking place with the text processing (Tan 1999). Relations extracted are of a type that is anticipated and less interesting, despite being very relevant and accurate. Second, because systems use lexical constraints, the performance of the system is more dependent upon having training data of the type and domain of documents being processed.

More Portable IE Parsing Algorithms

While generally MUC systems have been task specialized and less portable, there are also examples of individual information extraction algorithms that have utilized more general and portable knowledge. Many of these algorithms involve syntax parsing. Ciravegna and Lavelli present a full parsing approximation approach using finite-state cascades that is well suited for information extraction tasks (Ciravegna and Lavelli 2001). Autoslog, mentioned earlier, also included a syntax module that was able to determine the functional role of phrases in sentences. Such a module can be more easily reused on different domains. Also, one of the first to perform syntax analysis via cascaded finite state parsers was the FASTUS system from SRI (Hobbs, Appelt et al. 1996). These systems all perform more syntax analysis than just phrasing and are able to do it using regular grammars and cascaded finite state automata, making the approaches feasible for large scale information extraction.

The Arizona TerrorNet Approach

As mentioned above, some information extraction systems are more tailored to specific tasks and thus less portable to different domains and document types. The Arizona TerrorNet is required to parse documents and extract networks from varied source documents, thus requiring fewer lexical constraints. Despite using fewer lexical constraints, semantic phrase analysis is still required as relations of interest include only those between people. To meet the requirements of our extraction task and remain capable of extracting relations from multiple document types, we utilize a two-pronged strategy. First, we utilize a hybrid syntax-semantic tag with inherited properties for tagging the words in each document. The proper assignment of a hybrid tag is crucial for our semantic analysis. Second, we utilize a large amount of syntax parsing in the network extraction process.

Hybrid Syntax-Semantic Tag

Typical information extraction systems conduct syntax and then semantic tagging/analysis in a pipelined approach. In the Arizona TerrorNet, we have combined that information into one tag. We generally followed a naming convention where the tag head carried the semantic information and the syntactic information was carried by the tag tail (i.e. SEMANTIC_SYNTAX or LOCATION_NNP). In addition, each hybrid tag is defined in a tag ontology where tags have "is a" relationships with multiple parent tags. For example, Figure 1 shows an example of a LOCATION_NNP tag and its inheritance path. LOCATION_NNP is a singular proper noun. The tag, however, also inherits from general noun phrase, general semantic noun, semantic location, general proper noun, and finally named entity. All algorithms that deal with the hybrid tags, such as transformation-based algorithms and parsing algorithms are aware of each tags inherited identities. Thus, the more general a tag used in a rule is the more word classes and thus words will be affected. There are over 7,500 entries in the ontology with 3800 different tags. The majority of the tags come from verb groupings made by Beth Levine (Levin 1993). For each verb class, for example WALTZ, there are corresponding syntax additions: WALTZ_INF, WALTZ_VB, WALTZ_VBD, WALTZ_VBG, WALTZ_VBN, WALTZ_VBP, and WALTZ_VBZ

Using hybrid tags allows the grammar to be expressed in terms of syntax, semantics, or both because some parents have entirely semantic properties while others have syntax properties. For example, a rule that says combine a determiner (i.e. the) with a noun phrase (i.e. bomb) to create a new noun phrase (DT+NP=> NP) would apply to all the tags beneath NP in the hierarchy (shown in Figure 1). For that rule to just apply to location nouns, however, the rule would need to state a determiner with a location tag equals a noun phrase (DT+LOCATION=>NP). In cases where more than one rule applies to a sequence of tags, the rule with the combined least number of hops to the actual tags is the one that is activated. Syntax and semantic constraints can of course be expressed together in a grammar without using hybrid tags. However, having the hybrid information in the tag allowed us to statistically learn many grammar rules from lexicon and corpora thus reducing the first-time knowledge engineering cost.

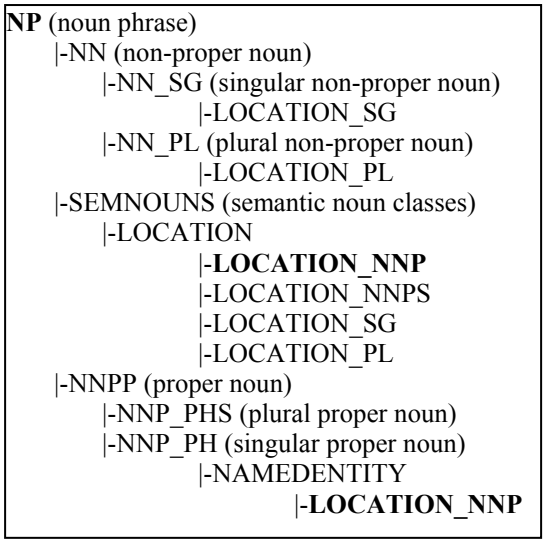

Figure 1 – Tag ontology inheritance for LOCATION_NNP

TerrorNet's semantic analysis relies on properly assigning the hybrid tag, particularly the semantic portion of the tag. There are differing views on how many word senses a lexical entry should contain (Wilks and Stevenson 1997). In our dictionary, we use fewer more vague tags more in line with research from Wierzbicka (Wierzbicka 1989). Our intention is to limit the number of semantic alternatives and thus improve tagging accuracy.

More Syntax Parsing

While full parse tree construction was limited in MUC, we have implemented additional syntax parsing to approximate fuller parsing. While syntax parsing cannot outperform lexically-based extraction rules, generally fewer rules are required to cover a domain (McDonald, Chen et al. 2004). In addition, syntax rules can be more readily reused between domains. By producing fewer rules that can potentially be reused on different types of documents, we reduce our knowledge engineering cost. We control the accuracy of our extraction rules by relying, when needed, on the semantic properties of the hybrid tag.

Our parsing algorithm uses a regular grammar that recognizes dependencies up to 20 tags away. Parsing rules are applied in six cascaded finite state automata (FSA). The first three levels of the FSA complete named entity recognition and noun phrasing, not including noun conjuncts. The last three FSA recognize embedded clauses, complementary phrases, and the functional role of the different phrases, such as subject, verb and object. In many cases we ignore prepositional attachment. While the first three levels rely heavily on rules using semantic word classes to recognize named entities, the final three levels utilize more syntax-related rules. We will now discuss the parsing steps in greater detail.

AZ TerrorNet Components

The Arizona TerrorNet consists of several natural language processing modules that are combined together to recognize relationships between known terrorists and then assemble those relationships into networks. The ordering of the components that make up the architecture of the TerrorNet is shown in Figure 2. We now explain in greater detail the function of each component.

Tokenization and Tagging

The steps of tokenization and tagging are shown in Figure 2 within the box labeled 1.The parsing begins by applying regular expressions that recognize dates, percents, address information, and monetary expressions in the text. Next, tokenization algorithms recognize word boundaries and sentence boundaries. The sentence splitting relies on a lexicon of 300 common abbreviations and rules to recognize new abbreviations. Documents are tokenized generally according to the PENN TREE BANK tokenizing rules for handling apostrophes and punctuation. In addition, words are also split on hyphens. After tokenization, phrases are recognized and tagged using finite state automata (FSA) so that each word in the text is visited only once. Approximately 18,000 phrases are tagged. Tagged phrases include locations, organizations, as well as discourse phrases such as "on the other hand" and multi-word prepositions, such as "by means of". Discourse phrases were adopted from research by Marcu (Marcu 2000). Tags applied to phrases are hybrid tags. Next hybrid word tags are applied to remaining words in the text using Brill's transformation-based algorithm (Brill 1993). The contextual rules, originally trained on the

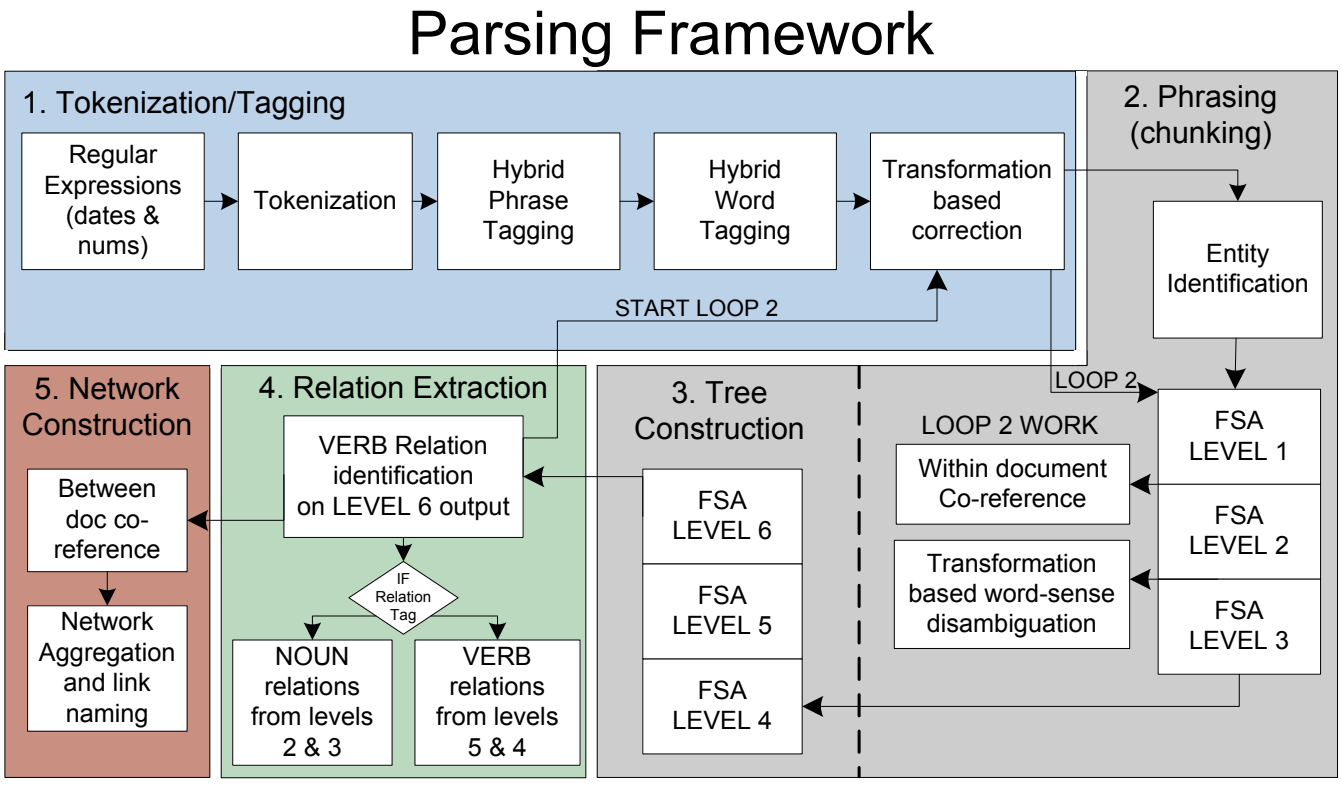

Figure 2 – The components of the Arizona TerrorNet

PENN TREE BANK and Brown corpora have been altered to reflect the hybrid tags. In addition, due to our use of hybrid syntax-semantic tags, many transformations are related to word sense disambiguation. Thus we are using the transformation-based algorithm for semantic and syntax tagging at the same time. The transformation-based algorithm has also previously been used for semantic tagging in research (Wilks and Stevenson 1997).

Phrasing (Chunking)

Following tokenization and hybrid tagging, we perform phrasing or chunking. Chunking, first proposed by Abney (Abney 1991), is an attempt to group words together recognizing linguistic theory as well as empirical studies of how users group words. Chunking has been used before to improve the parsing process (Ciravegna and Lavelli 1997). An additional benefit is that chunks can be recognized by computationally inexpensive methods. The process of recognizing chunks takes place at different levels. The smallest chunk we recognize is a named entity as defined by the MUC-7 conference (DARPA 1998). These entities include locations, people, organizations, dates, times, money, and percents. So at the entity stage the phrase "Microsoft employee Bill Gates" would be three chunks, Microsoft/ORGANIZATION_NNP, employee/ROLE_SG, Bill Gates/PERSON_NNP. Entity identification rules by default do not descend the tag ontology, though adding a flag to a tag forces it to descend the ontology. Level 1 of the FSA performs just as the entity identification step, with the only difference being that by default rules do use the ontology to match candidate tag sequences. For evaluation purposes, entities are considered named after level one of the FSA. After training the parser on 41 (of the 100 total) of the dry-run documents from MUC-7 we were able to achieve an 85 percent f-score (equal weighting for precision and recall) on the MUC-7 test set. Details of our entity extraction approach are beyond the scope of this paper. In general, sequences of hybrid tags are used to identify named entities. All the parsing rules (including both sections 3 and 4) use a binary rule format, with a rule pattern ($\gamma\alpha\delta$) and a corresponding transformation. From the rule pattern, α is the rule core and cannot be empty. The γ is prior context and the δ is future context and either can be empty. A transformation is the new hybrid tag that is added to the parse tree at the next higher level. An example parsing rule is shown in Figure 3. In Figure 3, the rule core is "NP CC NP" and is transformed to an NP when the rule core is preceded by a "BY" tag and followed by a "." tag.

At level 2 of the FSA, certain named entities are combined together to form new named entities. For example, the three tags "CITY_NNP , STATE_NNP" are transformed to the tag LOCATION_NNP, where at level 1 the locations are left separate. At level 3 of the FSA multiple entities of any kind can be combined together into a single noun phrase. For example, "Microsoft employee Bill Gates" would now all be a single phrase with the tag PERSONCOMPLEX_NNP. Unlike just the PERSON_NNP tag, the PERSONCOMPLEX_NNP lets the parser know that a noun relationship exists within the noun string, namely "Bill Gates - is employee of – Microsoft". After level 3, all noun phrases should be as long as possible without creating any noun phrases with coordinating conjunctions.

```
<GRAMMAR LEVEL= "4">
<RULE NUM=1>
<RULEPATTERN>
<PREVIOUSCONTEXT TAG="BY" />
<RULECORE>NP CC NP</RULECORE>
<FUTURECONTEXT TAG="." />
</RULEPATTERN>
<TRANSFORMATION>
NP
</TRANSFORMATION>
</RULE>
</GRAMMAR>
```

Figure 3 – A parsing rule with a rule pattern and transformation

Tree Construction

At this stage, phrase structure parsing takes place. Rules corresponding to this stage are primarily syntactic. As a result, fewer rules exist in levels 4, 5, and 6 than at the lower phrasing levels which tend to have more semantic rules. In level 4, conjuncts of noun phrases are recognized and transformed into new noun phrases, while conjunctions serving discourse purposes are left alone. Pronouns are also now combined with noun phrases and prepositional phrases are constructed but not attached. The rules at levels 5 and 6 identify embedded and relative clauses. The tag string after the 6th cascade should have no remaining embedded or relative clauses.

Relation Extraction

The relation extraction algorithm takes as input sequences of up to 20 hybrid tags from the output of the phrasing and tree construction stage. Relations at this stage are only recognized within single sentences. Relation extraction rule have a similar format as the parsing rules. The tag sequence is matched against a rule pattern ($\gamma\alpha\delta$) that has a corresponding transformation. Like the parsing rules, only the rule core (α) cannot be empty. The essence of the transformation is that a set of relation definitions are

returned that correspond to the input sequence of hybrid tags. Relations are roughly equivalent to subject, verb, object relations. Which phrases participate in which relations and in what capacity is then recorded to be utilized by the transformation-based algorithm and by the network construction algorithm.

After relation identification has occurred for the first time, AZ TerrorNet loops back to the transformation-based algorithm. In this iteration, the transformation-based algorithm transforms identified entities and/or their boundaries based on an expanded group of features. The feature set we use in this iteration includes the phrases' functional role in the sentence (subject or object), the verb class(es) corresponding to the phrase, and various combinations of the hybrid tags surrounding the entity. Some of the new features were made possible by the higher-level parsing and relation identification steps performed in the first loop. After the transformation-based algorithm is done, control passes to the cascade of FSA that perform chunking. All the parsing rules are rerun based on the sequence of tags that have been transformed for the second time. In this second loop, there are also additional components that are added to the phrasing stage. After level 1 of the FSA a within document co-reference algorithm is run. In this step primarily pronouns are resolved to their antecedents in the document. Because we are extracting relations between people, the resolution of pronouns is important. After level 2, the transformation-based algorithm is run for the third and final time. Different from the prior two times, the input into the algorithm is a sequence of tags that have been combined at a higher level in the tree. In addition, the transformations taking place rely primarily on corresponding verb classes and the functional role of the tags being transformed. Finally, all transformations are allowed to occur at this level regardless of whether the phrase has a history of ever being properly tagged as the new tag.

Network Construction

Once the second loop of processing is complete, the network is assembled in the network construction stage, which is a two step process involving between document co-reference and link classification. A between document co-reference algorithm must recognize when different noun strings in the nodes of the relations actually refer to the same person. Because we are interested only in known terrorists, the process of identifying the nodes begins by comparing the people listed in the relations to a list of 400 known terrorist identified in previous research (Sageman 2004). The name matching algorithm requires at least last names to match under all circumstances. If there are two names in the string, then the last name and first name must match. Heuristics are used to differentiate last names from names indicating lineage, such as "al-Zawahiri". If matches cannot be made at first pass, different transliterations of the names are generated and the matching process is repeated. Examples of different transliterations for the last name "Hussein" include "Hocine", "Husayn", and "Hussayn". We are currently using a lexicon containing nearly 1,000 Arab names with various transliterations.

Once the nodes have been matched up and interactions have been aggregated, the nature of the relationship between the two people is characterized. A link classification algorithm takes as input the verb classes of the interactions between two people. The output of the classification algorithm is one of seven relation categories. The seven relationship categories used include friend, kin, Imam, teacher, antagonistic type, operational/work type, and one that cannot be determined. We are using the verb classes identified by Beth Levine to train the classifier (Levin 1993). When verbs of the MURDER class are encountered, the classifier should return antagonistic type of relationship. When verbs of the ADMIRE class, however, are encountered the resulting classification should be either friend or teacher. The classification algorithm is currently being trained to improve performance.

An Example Network

Once the relationships have been aggregated and a label for the type of relationship generated, the network is generated in XML and passed to a network visualizer for display. A screen shot of the network visualizer is shown in Figure 4. As a test of the Arizona TerrorNet, we selected 200 previously unseen open-source news documents that had been identified as useful by a domain expert (Sageman 2004). We then ran all 200 documents and produced a network. A subset of that network is shown in Figure 4. As is seen in Figure 4, we have not yet applied the link classification to the multiple connectors between terrorists. As mentioned earlier, training is currently underway to test the potential of a classification into one of seven different relationship groups. The 200 documents used in the test came from such sources as the New York Times, Washington Post, Al-Sharq al-Awsat, The Boston Globe, Le Figaro, The Straits Times, Agence France Presse, and the Los Angeles Times. Over 200 relationships were extracted from the 200 documents. In Figure 4, the names that are capitalized have been matched to lists of known terrorists. An Al Qaeda network expert inspected the entire network produced from the 200 documents and confirmed many of the links.

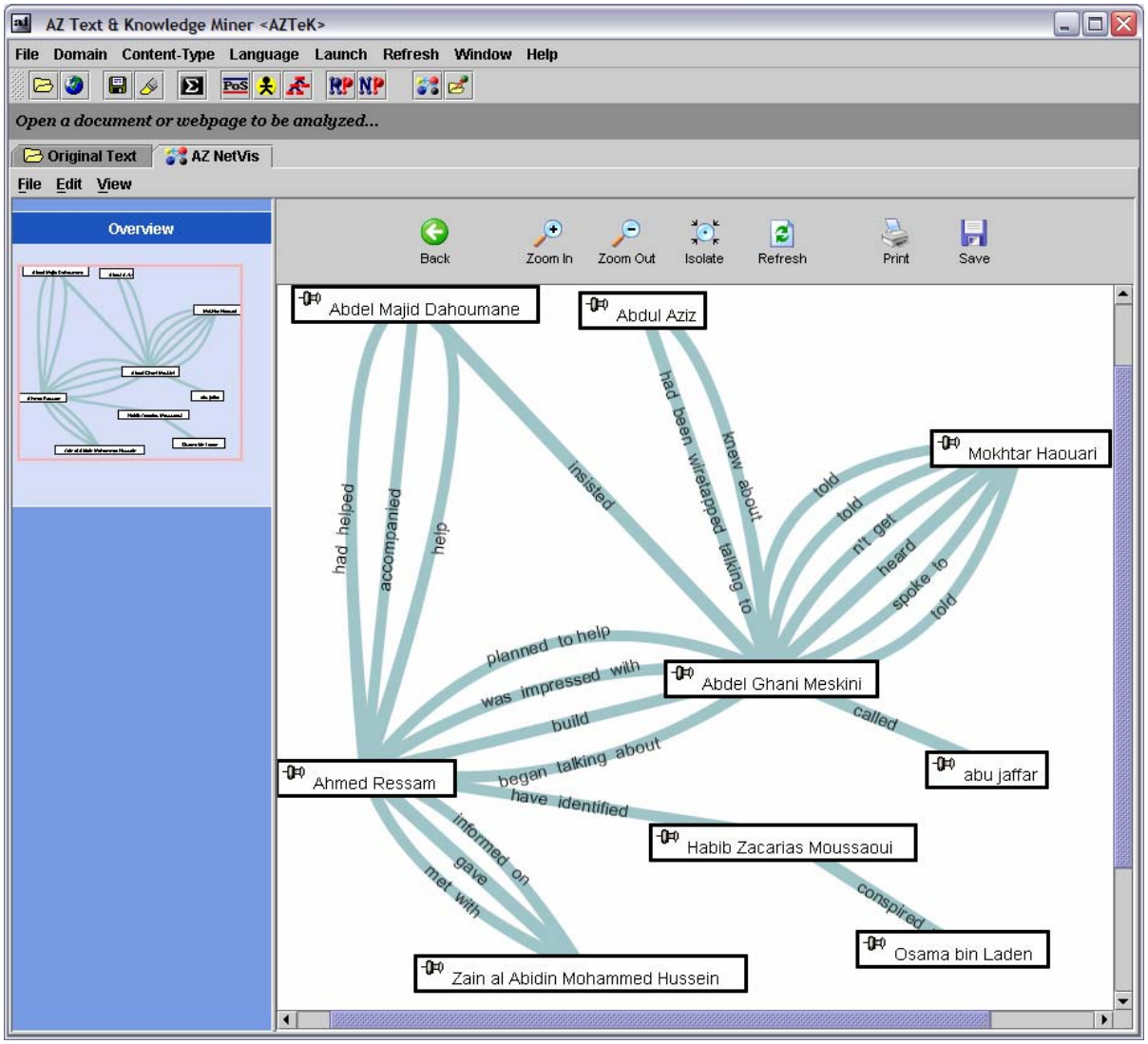

Figure 4: Automatically generated terrorist network from open source documents

Conclusions

We have presented an approach for extracting terrorist networks from open-source texts, the Arizona TerrorNet. The task requires the system to extract relationships from various types of documents and thus extraction rules have to be somewhat portable. In addition, more than just extracting reported relationships, the system has to aggregate relations and connect them in a network. The aggregation and analysis steps have the potential to alert analysts to relations that might not have been explicitly stated in the text. An example of such a situation may involve a transitive relation or a newly understood interaction that is the result of several aggregated interactions between the two terrorists. We ran the parser on 200 previously unseen documents and produced some promising results although very preliminary.

References

Abney, S. (1991). Parsing by chunks. *Principle-Based Parsing*. Dordrecht, Kluwer Academic Publishers.

Brill, E. (1993). A Corpus-Based Approach to Language Learning. Computer Science. Philadelphia, University of Pennsylvania.

Ciravegna, F. (2001). "Challenges in Information Extraction from Text for Knowledge Management." *IEEE Intelligent Systems and their Applications* **16**(6): 84.

Ciravegna, F. and A. Lavelli (1997). Controlling Bottom-Up Chart Parsers through Text Chunking. *5th International*

Workshop on Parsing Technologies (IWPT97), Boston, MA.

Ciravegna, F. and A. Lavelli (2001). "Full Parsing Approximation for Information Extraction via Finite-State Cascades." *Natural Language Engineering* **1**(1): 1-21.

Ciravegna, F. and D. Petrelli (2001). User Involvement in Adaptive Information Extraction: Position Paper. *IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, Seattle, WA, AAAI Working Notes.

Cowie, J. and W. Lehnert (1996). "Information Extraction." *Communications of the ACM* **39**(1): 80-91.

DARPA (1991). Proceedings of the 3rd Message Understanding Conference (MUC-3), San Diego, California, Morgan Kaufmann.

DARPA (1992). Proceedings of the 4th Message Understanding Conference (MUC-4), McLean, VA, Morgan Kaufmann.

DARPA (1998). Proceedings of the 7th Message Understanding Conference (MUC-7), Washington, D.C., Morgan Kaufmann.

Hobbs, J., D. Appelt, et al. (1996). FASTUS: Extracting Information from Natural Language Texts. *Finite State Devices for Natural Language Processing*. E. Roche and Y. Schabes, MIT Press.

Huffman, S. (1995). Learning information extraction patterns from examples. *IJCAI-95 Workshop on new approaches to learning for natural language processing*.

Kean, T. H., L. H. Hamilton, et al. (2004). The 9/11 Commission Report, The 9/11 Commission. **2005**.

Kim, J. and D. Moldovan (1995). "Acquisition of linguistic patterns for automatic information extraction." *IEEE Transactions on Knowledge and Data Engineering* **7**(5): 713-724.

Levin, B. (1993). *English Verb Classes and Alternations*. Chicago, The University of Chicago Press.

Mani, D., D. House, et al. (1998). The tipster summac text summarization evaluation: Final report, DARPA.

Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*. Boston, MA, MIT Press.

McDonald, D., H. Chen, et al. (2004). "Extracting Gene Pathway Relations Using a Hybrid Grammar: The Arizona Relation Parser." *Bioinformatics*.

Riloff, E. (1993). "Automatically constructing a dictionary for information extraction tasks." *Proceedings of the 11th National Conference on Artificial Intelligence*: 811-816.

Sageman, M. (2004). *Understanding Terror Networks*, University of Pennsylvania Press.

Soderland, S., D. Fisher, et al. (1995). Crystal: Inducing a conceptual dictionary. *14th International Joint Conference on Artificial Intelligence (IJCAI-95)*.

Tan, A.-H. (1999). Text Mining: The state of the art and the challenges. *PAKDD workshop on Knowledge Discovery from Advanced Databases*, Beijing, China.

Wierzbicka, A. (1989). *Semantics, Culture and Cognition*. Oxford, Oxford University Press.

Wilks, Y. and M. Stevenson (1997). Sense Tagging: Semantic Tagging with a Lexicon. *SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What and How?*, Washington D.C.