Chapter 11: Text Mining

Lecturer: Dr. *Nguyen Thi Ngoc Anh*
Email: ngocanhnt@ude.edu.vn

1

Data Mining in Text

- Association search in text corpuses provides suggestive information
  - Groups of related entities
  - Clusters that identify topics
- Flexibility is crucial
  - Describe what an interesting pattern would look like
  - What causes items to be considered associated
    same document, sequential associations, ?
  - Choice of techniques to rank the results
- Integrate with Information Retrieval systems
  - Common base preprocessing (e.g. Natural Language processing)
  - Need IR system to explore/understand text mining results

Why Text is Hard

- Lack of structure
  - Hard to preselect only data relevant to questions asked
  - Lots of irrelevant "data" (words that don't correspond to interesting concepts)
- Errors in information
  - Misleading/wrong information in text
  - Synonyms/homonyms: concept identification hard
  - Difficult to parse *meaning*
    I believe X is a key player vs. I doubt X is a key player
- Sheer volume of "patterns"
  - Need ability to focus on user needs
- Consequence for results:
  - False associations
  - Vague, dull associations

What About Existing Products? Data Mining Tools

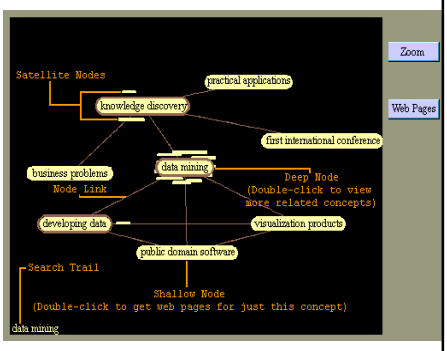
- Designed for particular types of analysis on structured data
  - Structure of data helps define known relationship
  - Small, inflexible set of "pattern templates"
- Text is "free flow of ideas", tough to capture precise meaning
  - Many patterns exist that aren't relevant to problem
- Experiments with COTS products on tagged text corpuses demonstrate these problems
  - "Discovery overload": many irrelevant patterns, density of actionable items too low
  - Lack of integration with Information Retrieval systems makes further exploration/understanding of results difficult

What About Existing Products? "Text Mining" Information Retrieval Tools

- "Text Mining" is (mis?)used to mean information retrieval
  - IBM TextMiner *(now called "IBM Text Search Engine")*
  - http://www.ibm.com/software/data/iminer/fortext/ibm_tse.html
  - DataSet http://www.ds-dataset.com/default.htm
- These are *Information Retrieval* products
  - Goal is *get the right document*
- May *use* data mining technology (clustering, association)
  - Used to improve retrieval, not discover associations among concepts
- No capability to discover patterns among *concepts in the documents*.
- May incorporate technologies such as concept extraction that ease integration with a Knowledge Discovery in Text system

What About Existing Products? Concept Visualization

- Goal: Visualize concepts in a corpus
  - SemioMap http://www.semio.com/
  - SPIRE http://www.pnl.gov/Statistics/research/spire.html
  - Aptex Convectis http://www.aptex.com/products-convectis.htm
- High-level concept visualization
  - Good for major trends, patterns
- Find concepts related to a particular query
  - Helps find patterns if you know some of the *instances* of the pattern
- Hard to visualize "rare event" patterns



What About Existing Products? Corpus-Specific Text Mining

- Some "Knowledge Discovery in Text" products
  - Technology Watch *(patent office)* http://www.ibm.com/solutions/businessintelligence/textmining/techwatch.htm
  - TextSmart *(survey responses)* http://www.spss.com/textsmart
- Provide limited types of analyses
  - Fixed "questions" to be answered
  - Primarily high-level (similar to concept visualization)
- Domain-specific
  - Designed for specific corpus and task
  - Substantial development to extend to new domain or corpus

What About Existing Products? Text Mining Tools

- True "Text Mining" just beginning to come to market
  - Associations: ClearForest http://www.clearforest.com
  - Semantic Networks: Megaputer's TextAnalyst™ http://www.megaputer.com/taintro.html
  - IBM Intelligent Miner for Text *(toolkit)* http://www.ibm.com/software/data/iminer/fortext
- Currently limited capabilities (but improving)
  - Further research needed
  - Directed research will ensure the *right* problems are solved
- Major Problem: Flood of Information
  - Analyzing results as bad as reading the documents

Scenario: Find Active Leaders in a Region

- Goal: Identify people to negotiate with prior to relief effort
  - Want "general picture" of a region
  - No expert that already knows the situation is available
- Problems:
  - No clear "central authority"; problems are regional
  - Many claim power/control, few have it for long
  - Must include *all* key players in a region
- Solution: Find key players *over time*
  - Who is key today?
  - Past players (may make a comeback)

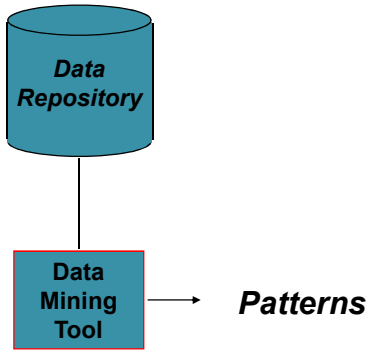
Example: Association Rules in News Stories

- Goal: Find related (competing or cooperating) players in regions
- Simple association rules (any associated concepts) gives too many results
- *Flexible* search for associations allows us to specify what we want: Gives fewer, more appropriate results

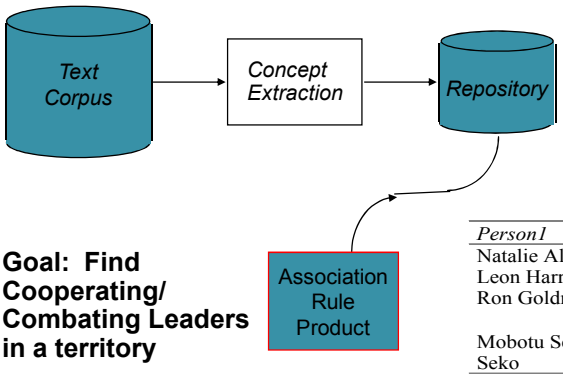
Person1	*Person2*	*Support*
Natalie Allen	Linden Soles	117
Leon Harris	Joie Chen	53
Ron Goldman	Nicole Simpson	19
	. . .	
Mobotu Sese Seko	Laurent Kabila	10

Person1	*Person2*	*Place*	*Support*
Mobuto Sese Seko	Laurent Kabila	Kinshasa	7

Conventional
Data Mining System Architecture



Using Conventional Tools:
Text Mining System Architecture

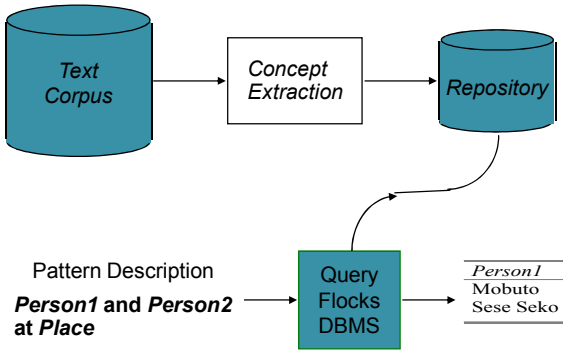


Goal: Find Cooperating/ Combating Leaders in a territory

Person1	*Person2*	
Natalie Allen	Linden Soles	117
Leon Harris	Joie Chen	53
Ron Goldman	Nicole Simpson	19
	. . .	
Mobotu Sese Seko	Laurent Kabila	10

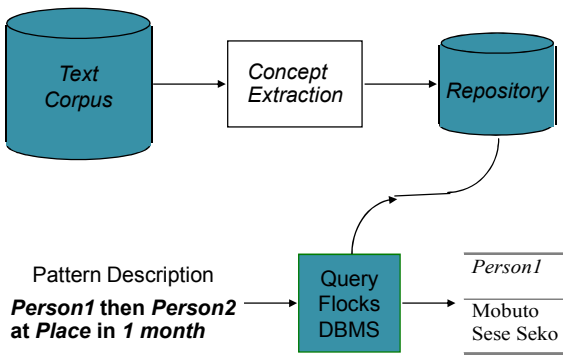
Too Many Results

Flexible
Text Mining System Architecture



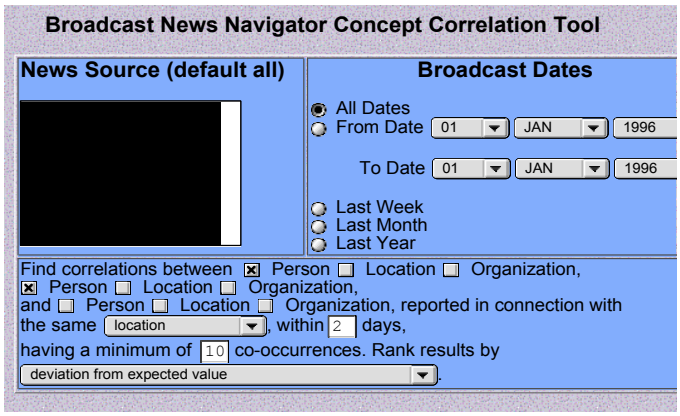
Person1	*Person2*	*Place*	
Mobuto Sese Seko	Laurent Kabila	Kinshasa	7

Flexible: Adapts to new tasks
Text Mining System Architecture



Person1	*Person2*	*Place*	*Avg. Days*
Mobuto Sese Seko	Laurent Kabila	Kinshasa	12

Example of Flexible Association Search



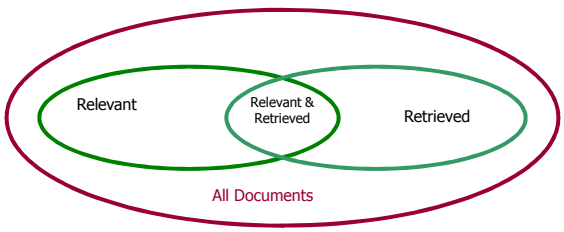
Text Databases and IR

- Text databases (document databases)
  - Large collections of documents from various sources: news articles, research papers, books, digital libraries, e-mail messages, and Web pages, library database, etc.
  - Data stored is usually *semi-structured*
  - Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data
- Information retrieval
  - A field developed in parallel with database systems
  - Information is organized into (a large number of) documents
  - Information retrieval problem: locating relevant documents based on user input, such as keywords or example documents

Information Retrieval

- Typical IR systems
  - Online library catalogs
  - Online document management systems
- Information retrieval vs. database systems
  - Some DB problems are not present in IR, e.g., update, transaction management, complex objects
  - Some IR problems are not addressed well in DBMS, e.g., unstructured documents, approximate search using keywords and relevance

Basic Measures for Text Retrieval



- Precision: the percentage of retrieved documents that are in fact relevant to the query (i.e., "correct" responses)
- Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved

$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{relevant\}|}$$

Information Retrieval Techniques(1)

- Basic Concepts
  - A document can be described by a set of representative keywords called index terms.
  - Different index terms have varying relevance when used to describe document contents.
  - This effect is captured through the assignment of numerical weights to each index term of a document. (e.g.: frequency, tf-idf)
- DBMS Analogy
  - Index Terms → Attributes
  - Weights → Attribute Values

Information Retrieval Techniques(2)

- Index Terms (Attribute) Selection:
  - Stop list
  - Word stem
  - Index terms weighting methods
- Terms ✕ Documents Frequency Matrices
- Information Retrieval Models:
  - Boolean Model
  - Vector Model
  - Probabilistic Model

Boolean Model

- Consider that index terms are either present or absent in a document
- As a result, the index term weights are assumed to be all binaries
- A query is composed of index terms linked by three connectives: not, and, and or
  - e.g.: car *and* repair, plane *or* airplane
- The Boolean model predicts that each document is either relevant or non-relevant based on the match of a document to the query

Boolean Model: Keyword-Based Retrieval

- A document is represented by a string, which can be identified by a set of keywords
- Queries may use expressions of keywords
  - E.g., car *and* repair shop, tea *or* coffee, DBMS *but not* Oracle
  - Queries and retrieval should consider synonyms, e.g., repair and maintenance
- Major difficulties of the model
  - Synonymy: A keyword *T* does not appear anywhere in the document, even though the document is closely related to *T*, e.g., data mining
  - Polysemy: The same keyword may mean different things in different contexts, e.g., mining

Similarity-Based Retrieval in Text Databases

- Finds similar documents based on a set of common keywords
- Answer should be based on the degree of relevance based on the nearness of the keywords, relative frequency of the keywords, etc.
- Basic techniques
- Stop list
  - Set of words that are deemed "irrelevant", even though they may appear frequently
  - E.g., *a, the, of, for, to, with*, etc.
  - Stop lists may vary when document set varies

Similarity-Based Retrieval in Text Databases (2)

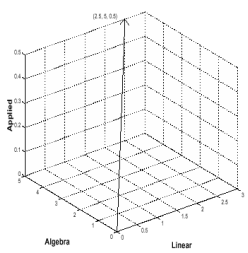
- Word stem
  - Several words are small syntactic variants of each other since they share a common word stem
  - E.g., *drug, drugs, drugged*
- A term frequency table
  - Each entry *frequent_table(i, j)* = # of occurrences of the word t_i in document d_i
  - Usually, the *ratio* instead of the absolute number of occurrences is used
- Similarity metrics: measure the closeness of a document to a query (a set of keywords)
  - Relative term occurrences
  - Cosine distance: $sim(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|}$

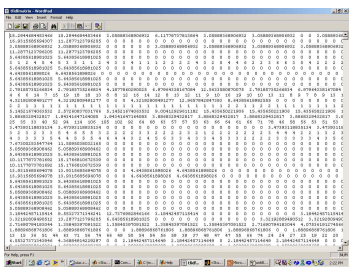
Indexing Techniques

- Inverted index
  - Maintains two hash- or B+-tree indexed tables:
    - document_table: a set of document records <doc_id, postings_list>
    - term_table: a set of term records, <term, postings_list>
  - Answer query: Find all docs associated with one or a set of terms
  - + easy to implement
  - – do not handle well synonymy and polysemy, and posting lists could be too long (storage could be very large)
- Signature file
  - Associate a signature with each document
  - A signature is a representation of an ordered list of terms that describe the document
  - Order is obtained by frequency analysis, stemming and stop lists

Vector Model

- Documents and user queries are represented as m-dimensional vectors, where m is the total number of index terms in the document collection.
- The degree of similarity of the document d with regard to the query q is calculated as the correlation between the vectors that represent them, using measures such as the Euclidian distance or the cosine of the angle between these two vectors.



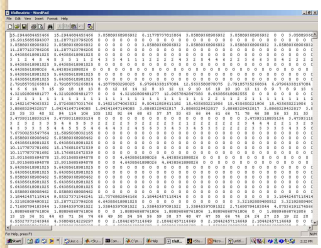


Latent Semantic Indexing (1)

- Basic idea
  - Similar documents have similar word frequencies
  - Difficulty: the size of the term frequency matrix is very large
  - Use a singular value decomposition (SVD) techniques to reduce the size of frequency table
  - Retain the *K* most significant rows of the frequency table
- Method
  - Create a term x document weighted frequency matrix A
  - SVD construction: A = U * S *V'
  - Define K and obtain U_k , S_k , and V_k.
  - Create query vector q' .
  - Project q' into the term-document space: Dq = q' * U_k * S_k^{-1}
  - Calculate similarities: cos α = Dq . D / ||Dq|| * ||D||

Latent Semantic Indexing (2)

Weighted Frequency Matrix



Query Terms:
- Insulation
- Joint

DOCUMENTS:

'CM031.txt'

'CM046.txt'

'CM001.txt'

'CM029.txt'

'CM040.txt'

K>> return

TERMS:

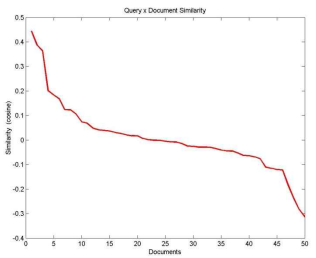
'joint'

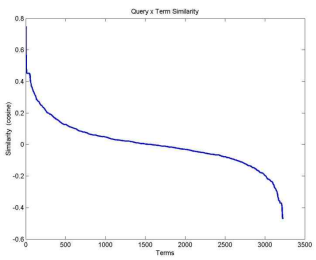
'insulation'

'roofing'

'expansion'

'saw'





Probabilistic Model

- Basic assumption: Given a user query, there is a set of documents which contains exactly the relevant documents and no other (ideal answer set)
- Querying process as a process of specifying the properties of an ideal answer set. Since these properties are not known at query time, an initial guess is made
- This initial guess allows the generation of a preliminary probabilistic description of the ideal answer set which is used to retrieve the first set of documents
- An interaction with the user is then initiated with the purpose of improving the probabilistic description of the answer set

Types of Text Data Mining

- Keyword-based association analysis
- Automatic document classification
- Similarity detection
  - Cluster documents by a common author
  - Cluster documents containing information from a common source
- Link analysis: unusual correlation between entities
- Sequence analysis: predicting a recurring event
- Anomaly detection: find information that violates usual patterns
- Hypertext analysis
  - Patterns in anchors/links
    - Anchor text correlations with linked objects

Keyword-Based Association Analysis

- Motivation
  - Collect sets of keywords or terms that occur frequently together and then find the association or correlation relationships among them
- Association Analysis Process
  - Preprocess the text data by parsing, stemming, removing stop words, etc.
  - Evoke association mining algorithms
    - Consider each document as a transaction
    - View a set of keywords in the document as a set of items in the transaction
  - Term level association mining
    - No need for human effort in tagging documents
    - The number of meaningless results and the execution time is greatly reduced

Text Classification(1)

- Motivation
  - Automatic classification for the large number of on-line text documents (Web pages, e-mails, corporate intranets, etc.)
- Classification Process
  - Data preprocessing
  - Definition of training set and test sets
  - Creation of the classification model using the selected classification algorithm
  - Classification model validation
  - Classification of new/unknown text documents
- Text document classification differs from the classification of relational data
  - Document databases are not structured according to attribute-value pairs

Text Classification(2)

- Classification Algorithms:
  - Support Vector Machines
  - K-Nearest Neighbors
  - Naïve Bayes
  - Neural Networks
  - Decision Trees
  - Association rule-based
  - Boosting

			#1	#2	#3	#4	#5
		# of documents # of training documents # of test documents # of categories	21,450 14,704 6,746 135	14,347 10,667 3,680 93	13,272 9,610 3,662 92	12,902 9,603 3,299 90	12,902 9,603 3,299 10
System	Type	Results reported by					
Word	(non-learning)	[Yang 1999]	150	310	290		
ProbBayes	probabilistic	[Dumais et al. 1998]				.752	.815
	probabilistic	[Joachims 1998]					.720
	probabilistic	[Lam et al. 1997]	.443 (MF_1)				
	probabilistic	[Lewis 1992a]	.650				
Bim	probabilistic	[Li and Yamanishi 1999]				.747	
	probabilistic	[Li and Yamanishi 1999]				.773	
Nb	probabilistic	[Yang and Liu 1999]				.795	
	decision trees	[Dumais et al. 1998]					.884
C4.5	decision trees	[Joachims 1998]					.794
Ind	decision trees	[Lewis and Ringuette 1994]	.670				
Swap-1	decision rules	[Apté et al. 1994]		.805			
Ripper	decision rules	[Cohen and Singer 1999]	.683	.811		.820	
SleepingExperts	decision rules	[Cohen and Singer 1999]	**.753**	.759		.827	
Dl-Esc	decision rules	[Li and Yamanishi 1999]				.820	
Charade	decision rules	[Moulinier and Ganascia 1996]		.738			
Charade	decision rules	[Moulinier et al. 1996]		.783 (F_1)			
Llsf	regression	[Yang 1999]		.855	.810		
Llsf	regression	[Yang and Liu 1999]				.849	
BalancedWinnow	on-line linear	[Dagan et al. 1997]	.747 (M)	.833 (M)			
Winnow-Ripp	on-line linear	[Lam and Ho 1998]				.822	
Rocchio	batch linear	[Cohen and Singer 1999]	.660	.748		.776	
FindSim	batch linear	[Dumais et al. 1998]				.617	.646
Rocchio	batch linear	[Joachims 1998]					.799
Rocchio	batch linear	[Lam and Ho 1998]				.781	
Rocchio	batch linear	[Li and Yamanishi 1999]				.625	
Classi	neural network	[Ng et al. 1997]		.802			
Nnet	neural network	[Yang and Liu 1999]				.838	
	neural network	[Wiener et al. 1995]			.820		
Cin.W	example-based	[Lam and Ho 1998]				.860	
k-NN	example-based	[Joachims 1998]					.823
k-NN	example-based	[Lam and Ho 1998]				.820	
k-NN	example-based	[Yang 1999]	.690	.852	**.820**		
k-NN	example-based	[Yang and Liu 1999]				.856	
SvmLight	SVM	[Dumais et al. 1998]				.870	**.920**
SvmLight	SVM	[Joachims 1998]					.864
SvmLight	SVM	[Li and Yamanishi 1999]				.841	
SvmLight	SVM	[Yang and Liu 1999]				.859	
AdaBoost.MH	committee	[Schapire and Singer 2000]		.860			
	committee	[Weiss et al. 1999]				**.878**	
	Bayesian net	[Dumais et al. 1998]				.800	.850
	Bayesian net	[Lam et al. 1997]	.542 (MF_1)				

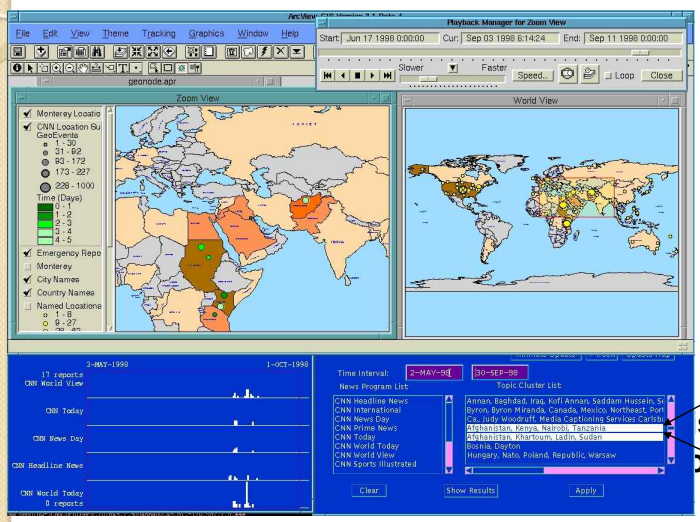
Document Clustering

- Motivation
  - Automatically group related documents based on their contents
  - No predetermined training sets or taxonomies
  - Generate a taxonomy at runtime
- Clustering Process
  - Data preprocessing: remove stop words, stem, feature extraction, lexical analysis, etc.
  - Hierarchical clustering: compute similarities applying clustering algorithms.
  - Model-Based clustering (Neural Network Approach): clusters are represented by "exemplars". (e.g.: SOM)

Goal: Automatically Identify Recurring Topics in a News Corpus

- Started with a user problem: Geographic analysis of news
- Idea: Segment news into ongoing topics/stories
  How do we do this?
- What we need:
  - Topics
  - "Mnemonic" for describing/remembering the topic
  - Mapping from news articles to topics
- Other goals:
  - Gain insight into collection that couldn't be had from skimming a few documents
  - Identify key players in a story/topic

User Problem: Geographic News Analysis



TopCat identified separate topics for U.S. embassy bombing and counter-strike.

Bombing

Counter-strike

A Data Mining Based Solution
Idea in Brief

- A topic often contains a number of recurring players/concepts
  - Identified highly correlated named entities (frequent itemsets)
  - Can easily tie these back to the source documents
  - *But there were too many to be useful*
- Frequent itemsets often overlap
  - Used this to cluster the correlated entities
  - But the link to the source documents is no longer clear
  - Used "topic" (list of entities) as a query to find relevant documents to compare with known mappings
- Evaluated against manually-categorized "ground truth" set
  - Six months of print, video, and radio news: 65,583 stories
  - 100 topics manually identified (covering 6941 documents)

TopCat Process

- Identify named entities (person, location, organization) in text
  - Alembic natural language processing system
- Find highly correlated named entities (entities that occur together with unusual frequency)
  - Query Flocks association rule mining technique
  - Results filtered based on strength of correlation and number of appearances
- Cluster similar associations
  - Hypergraph clustering based on hMETIS graph partitioning algorithm (based on (Han et. al. 1997))
  - Groups entities that may not appear together in a single broadcast, but are still closely related

Preprocessing

- Identify named entities (person, location, organization) in text
  - Alembic Natural Language Processing system
- Data Cleansing:
  - Coreference Resolution
    - Used intra-document coreference from NLP system
    - Heuristic to choose "global best name" from different choices in a document
  - Eliminate composite stories
    - Heuristic - same headline monthly or more often
  - High Support Cutoff (5%)
    - Eliminate overly frequent named entities (only provide "common knowledge" topics)

Named Entities vs. Full Text

- Corpus contained about 65,000 documents.
- Full text resulted in almost 5 million unique word-document pairs vs. about 740,000 for named entities.
- Prototype was unable to generate frequent itemsets at support thresholds lower than 2% for full text.
  - At 2% support, one week of full text data took 30 times longer to process than the named entities at 0.05% support.
- For one week:
  - 91 topics were generated with the full text, most of which aren't readily identifiable.
  - 33 topics were generated with the named-entities.

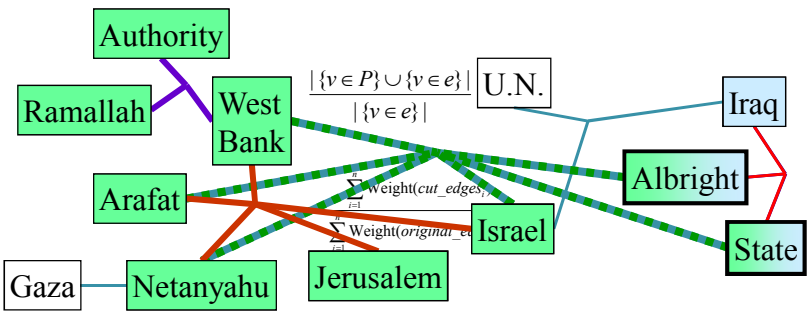
Frequent Itemsets

Israel	State	West Bank	Netanyahu	Albright	Arafat	*627390806*
Iraq	State	Albright				*479*
Israel	Jerusalem	West Bank	Netanyahu	Arafat		*4989413*
Gaza	Netanyahu					*39*
Ramallah	Authority	West Bank				*19506*
Iraq	Israel	U.N.				*39*

- Query Flocks association rule mining technique
  - ***22894 frequent itemsets with 0.05% support***
- Results filtered based on strength of correlation and support
  - ***Cuts to 3129 frequent itemsets***
- Ignored subsets when superset with higher correlation found
  - ***449 total itemsets, at most 12 items (most 2-4)***

Clustering

- Cluster similar associations
  - Hypergraph clustering based on hMETIS graph partitioning algorithm (adapted from (Han et. al. 1997))
  - Groups entities that may not appear together in a single broadcast, but are still closely related



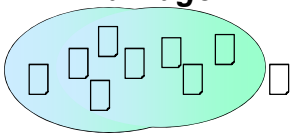
Mapping to Documents

- Mapping Documents to Frequent Itemsets easy
  - Itemset with support *k* has exactly *k* documents containing all of the items in the set.
- Topic clusters harder
  - Topic may contain partial itemsets
- Solution: Information Retrieval
  - Treat items as "keys" to search for
  - Use Term Frequency/Inter Document Frequency as distance metric between document and topic
- Multiple ways to interpret ranking
  - Cutoff: Document matches a topic if distance within threshold
  - Best match: Document only matches closest topic

Merging

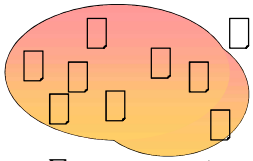
- Topics still to fine-grained for TDT
  - Adjusting clustering parameters didn't help
  - Problem was sub-topics
- Solution: Overlap in documents
  - Documents often matched multiple topics
  - Used this to further identify related topics

Marriage



$$\frac{\sum_{i \in documents} TFIDF_{ia} * TFIDF_{ib} / N}{\sum_{i \in documents} TFIDF_{ia} / N * \sum_{i \in documents} TFIDF_{ib} / N}$$

Parent/Child



$$\frac{\sum_{i \in documents} TFIDF_{ip} * TFIDF_{ic} / N}{\sum_{i \in documents} TFIDF_{ic} / N}$$

TopCat: Examples from Broadcast News

- LOCATION Baghdad
  PERSON Saddam Hussein
  PERSON Kofi Annan
  ORGANIZATION United Nations
  PERSON Annan
  ORGANIZATION Security Council
  LOCATION Iraq
- LOCATION Israel
  PERSON Yasser Arafat
  PERSON Walter Rodgers
  PERSON Netanyahu
  LOCATION Jerusalem
  LOCATION West Bank
  PERSON Arafat

TopCat Evaluation

- Tested on Topic Detection and Tracking Corpus
  - Six months of print, video, and radio news sources
  - 65,583 documents
  - 100 topics manually identified (covering 6941 documents)
- Evaluation results (on evaluation corpus, last two months)
  - Identified over 80% of human-defined topics
  - Detected 83% of stories within human-defined topics
  - Misclassified 0.2% of stories
- Results comparable to "official" Topic Detection and Tracking participants
  - Slightly different problem - retrospective detection
  - Provides "mnemonic" for topic (TDT participants only produce list of documents)

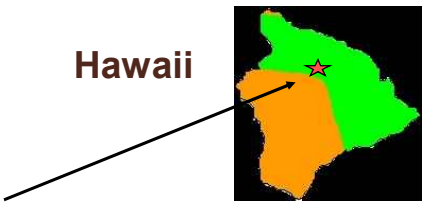
Project Participants

- MITRE Corporation
  - Modeling intelligence text analysis problems
  - Integration with information retrieval systems
  - Technology transfer to Intelligence Community through existing MITRE contracts with potential developers/first users
- Stanford University
  - Computational issues
  - Integration with database/data mining
  - Technology transfer to vendors collaborating with Stanford on other data mining work
- Visitors:
  - Robert Cooley (University of Minnesota, Summer 1998)
  - Jason Rennie (MIT, Summer 1999)

Where we're going now: Use of the Prototype

- MITRE internal:
  - Broadcast News Navigator
  - GeoNODE
- External Use:
  - Both Broadcast News Navigator and GeoNODE planned for testing at various sites
  - GeoNODE working with NIMA as test site
  - Incorporation in DARPA-sponsored TIDES Portal for Strong Angel/RIMPAC exercise this summer

Exercise Strong Angel
June 2000



The scenario… Humanitarian Assistance

- Increasing violence against Green minority in Orange
- Green minority refugees massing in border mountains
  - Ethnic Green crossing into Green, though Orange citizens
- Live bomblets found near roads
- Basics in short supply
  - water, shelter, medical care

What We've Learned: Recommendations/Thoughts for Further Work

- Want flexibility in describing patterns
  - What lends support to an association (e.g. across hyperlink; combining sequential, "standard" associations)
  - *Type* of associated entity important in describing pattern
- Major risk: density of "good stuff" in results too low
  - Problem isn't *wrong* results, but *uninteresting* results
  - Simple support/confidence rarely appropriate for text
  - Support a range of metrics - no single "proper measure"
- Cleaning and Mining as part of same process
  - Human cost of pre-mining cleansing too high
  - Human feedback on mining *results* (may alter results)

What we see in the Future: COTS support for Data Mining in Text

- Working with vendors to incorporate query flocks technology in DBMS systems
  - Stanford University working with IBM Almaden Research
- Working with vendors to incorporate text mining in information retrieval systems
  - MITRE discussing technology transition with Manning&Napier Information Services, Cartia
- More Research needed
  - What are the types of analyses that should be supported?
  - What are the right relevance measures to find *interesting* patterns, and how do we optimize these?
  - What additional capabilities are needed from concept extraction?

Potential Applications

- Topic Identification
  - Identify by different "types" of entities (person / organization / location / event / ?)
  - Hierarchically organize topics (in progress)
- Support for link analysis on Text
  - Tools exist for visualizing / analyzing links (e.g. NetMap)
  - Text mining *detects* links -- giving link analysis tools something to work with
- Support for Natural Language Processing / Document Understanding
  - Synonym recognition -- A and B may not appear together, but they each appear with X, Y, and Z -- A and B may be synonyms
- Prediction: Sequence analysis (in progress)

Similarity Search in Multimedia Data

- Description-based retrieval systems
  - Build indices and perform object retrieval based on image descriptions, such as keywords, captions, size, and time of creation
  - Labor-intensive if performed manually
  - Results are typically of poor quality if automated
- Content-based retrieval systems
  - Support retrieval based on the image content, such as color histogram, texture, shape, objects, and wavelet transforms

Queries in Content-Based Retrieval Systems

- Image sample-based queries
  - Find all of the images that are similar to the given image sample
  - Compare the feature vector (signature) extracted from the sample with the feature vectors of images that have already been extracted and indexed in the image database
- Image feature specification queries
  - Specify or sketch image features like color, texture, or shape, which are translated into a feature vector
  - Match the feature vector with the feature vectors of the images in the database