42

Text Mining and Information Extraction

Moty Ben-Dov[1] and Ronen Feldman[2]

[1] MDX University, London
[2] Hebrew university, Israel

Summary. Text Mining is the automatic discovery of new, previously unknown information, by automatic analysis of various textual resources. Text mining starts by extracting facts and events from textual sources and then enables forming new hypotheses that are further explored by traditional Data Mining and data analysis methods. In this chapter we will define text mining and describe the three main approaches for performing information extraction. In addition, we will describe how we can visually display and analyze the outcome of the information extraction process.

Key words: text mining, content mining, structure mining, text classification, information extraction, Rules Based Systems.

42.1 Introduction

The information age has made it easy for us to store large amounts of texts. The proliferation of documents available on the Web, on corporate intranets, on news wires, and elsewhere is overwhelming. However, while the amount of information available to us is constantly increasing, our ability to absorb and process this information remains constant. Search engines only exacerbate the problem by making more and more documents available in a matter of a few key strokes; So-called "push" technology makes the problem even worse by constantly reminding us that we are failing to track news, events, and trends everywhere. We experience information overload, and miss important patterns and relationships even as they unfold before us. As the old adage goes, "we can't see the forest for the trees."

Text-mining (TM), also known as Knowledge discovery from text (KDT), refers to the process of extracting interesting patterns from very large text database for the purposes of discovering knowledge. Text-mining applies the same analytical functions of data-mining but also applies analytic functions from natural language (NL) and information retrieval (IR) techniques (Dorre *et al.*, 1999).

The text-mining tools are used for:

O. Maimon, L. Rokach (eds.), *Data Mining and Knowledge Discovery Handbook*, 2nd ed.,
DOI 10.1007/978-0-387-09823-4_42, © Springer Science+Business Media, LLC 2010

- Extracting relevant information from a document – extract the features (entities) from a document by using NL, IR and association metrics algorithms (Feldman *et al.*, 1998) or pattern matching (Averbuch *et al.*, 2004).
- Finding trend or relations between people/places/organizations etc. by aggregating and comparing information extracted from the documents.
- Classifying and organizing documents according to their content (Tkach, 1998)
- Retrieving documents based on the various sorts of information about the document content.
- Clustering documents according to their content (Wai-chiu and Fu 2000).

A Text Mining system is composed of 3 major components (See Figure 42.1):

Information Feeders enable the connection between various textual collections and the tagging modules. This component connects to any web site, streamed source (such a news feed), internal document collections and any other types of textual collections.

Intelligent Tagging A component responsible for reading the text and distilling (tagging) the relevant information. This component can perform any type of tagging on the documents such as statistical tagging (categorization and term extraction), semantic tagging (information extraction) and structural tagging (extraction from the visual layout of documents).

Business Intelligence Suite A component responsible for consolidating the information from disparate sources, allowing for simultaneous analysis of the entire information landscape.

The TM task can be separated into two major categories according to their task and according to the algorithms and formal frameworks that they are using.

The first is the Task-oriented preprocessing approaches that envision the process of creating a structured document representation in terms of tasks and sub-tasks and usually involve some sort of preparatory goal or problem that needs to be solved.

The second is the preprocessing approaches that rely on techniques that derive from formal methods for analyzing complex phenomena and can be also applied to natural language texts. Such approaches include classification schemes, probabilistic models, rule-based systems approaches and other methodologies.

In this chapter we will first talk about the differences between TM and Text retrieval. Second we will describe the two approaches in TM - the task and the formal frameworks process.

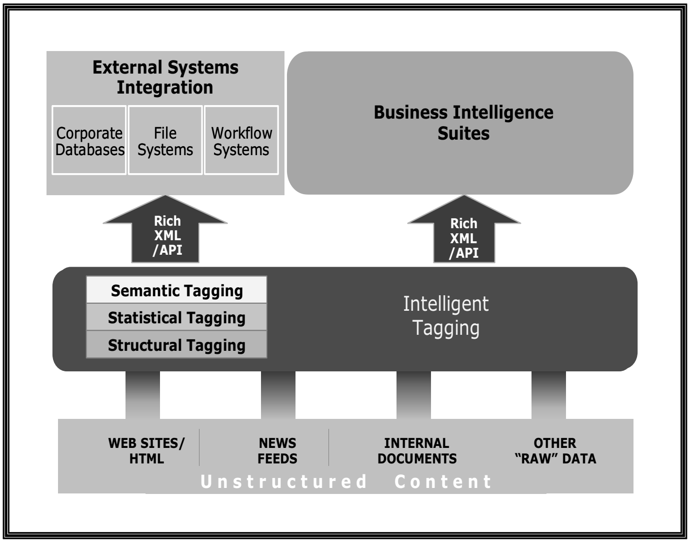


Fig. 42.1. Architecture of Text-Mining systems

42.2 Text Mining vs. Text Retrieval

It is important to differentiate between Text Mining (TM) and Text Retrieval (or information retrieval, as it is more widely known).

The goal of information retrieval is to help users find documents that satisfy their information needs (Baeza-Yates and Ribeiro-Neto, 1999). The standard procedure is analogous to looking for needles in a needle stack - the problem isn't so much that the desired information is not known, but rather that the desired information coexists with many other valid pieces of information (Hearst, 1999). The outcome of information retrieval process is documents.

The goal of TM is to discover or derive new information from text, finding patterns across datasets, and/or separating signal from noise. The fact that an information retrieval system can return a document that contains the information a user requested does not imply that a new discovery has been made: the information had to have already been known to the author of the text; otherwise the author could not have written it down.

On the other hand, the results of certain types of text processing can yield tools that indirectly *aid* in the information access process. Examples include text clustering to create thematic overviews of text collections (Cutting *et al*., 1992), automatically categorizing search results (Chen and Dumais, 2000), automatically generating term associations to aid in query expansion (Xu and Croft., 1996), and using co-citation analysis to find general topics within a collection or identify central web pages (Kleinberg, 1999).

The most important distinction between TM and Information Retrieval is the output of each process. In the IR process the out is documents, some it's clustered or ordered or scored but at the end to get the information we have to read the documents. In contrast the results of TM process can be features, patterns, connections, profiles or trends and to find the information we need we don't necessary have to read the documents.

42.3 Task-Oriented Approaches vs. Formal Frameworks

Two clear ways of categorizing the totality of preparatory document structuring techniques are according to their task and according to the algorithms and formal frameworks that they use.

Task-oriented preprocessing approaches envision the process of creating a structured document representation in terms of tasks and sub-tasks and usually involve some sort of preparatory goal or problem that needs to be solved. Other preprocessing approaches rely on techniques that derive from formal methods for analyzing complex phenomena that can be also applied to natural language texts. Such approaches include classification schemes, probabilistic models, rule-based systems approaches and other methodologies.

42.4 Task-Oriented Approaches

A document is an abstract which has a variety of possible actual representations. The task of the document structuring process is to take the most "raw" representation and convert it to the representation where the meaning of the document is understandable.

In order to cope with this extremely difficult problem, a "divide-and-conquer" strategy is typically employed. The problem is separated into a set of smaller subtasks, each of which is solved separately. The subtasks can broadly be divided into three classes (see Figure 42.2) – preparatory processing, general-purpose natural language processing ("NLP") tasks, and problem-dependent tasks.

- Preparatory processing converts the raw representation into a structure suitable for further linguistic processing. For example, the raw input may be a PDF document, a scanned page, or even recorded speech. The task of the preparatory processing is to convert the raw input into a stream of text, possibly labeling the internal text zones, such as paragraphs, columns, or tables. It is sometimes also possible for the preparatory processing to extract some document-level fields, such as ¡Author¿ or ¡Title¿, in case where the visual position of the fields allows their identification.

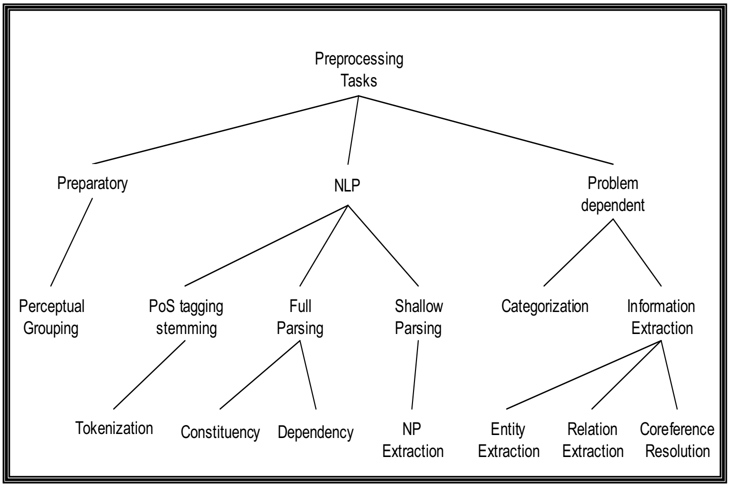


Fig. 42.2. Text Preprocessing Tasks.

- NLP process - The general-purpose NLP tasks process text documents using the general knowledge about natural language. The tasks may include tokenization, morphological analysis, part-of-speech tagging, and syntactic parsing, either shallow or deep.
  - Tokenization - The first step in information extraction from text is to identify the words in sentences in the text. The tokenizer performs this function. In English text, this process is fairly simple since white spaces and punctuation marks separate words. In other languages (e.g., Chinese, Japanese), in which spaces do not separate words, this process is more complex. Also, in some languages, use of hyphens to compound words (e.g., German, Dutch), this is a crucial step.
  - Part-of-speech Tagging - Part-of-speech tagging is the process of identifying a word's part of speech in a sentence (e.g., noun, verb, etc.) by its context. Tagging serves as the basis for the information retrieval system to perform syntax-sensitive filtering and analysis. Usually, PoS taggers at some stage of their processing perform morphological analysis of words. Thus, an additional output of a PoS tagger is a sequence of stems (also known as "lemmas") of the input words (Brill, 1992; Kupiec, 1992; Brill, 1995).
  - Syntactical parsing components perform a full syntactical analysis of sentences, according to a certain grammar theory. The basic division is between the constituency and dependency grammars (Keller, 1992; Pollard and Sag, 1994; Rambow and Joshi, 1994; Neuhaus and Broker, 1997).

    Shallow Parsing (Hammerton *et al.*, 2002) is the task of diving documents into non overlapping word sequences or phrases such that syntactically related words are grouped together. Each phrase is then tagged by one of a set of predefined tags such as: Noun Phrase, Verb Phrase, Prepositional Phrase, Adverb Phrase, Subordinated clause, Adjective Phrase, Conjunction Phrase, and List Marker. Shallow parsing is generally useful as a preprocessing step, either for bootstrapping, extracting information from corpora for use by more sophisticated parsers, or for end-user applications such as information extraction. Shallow parsing allows morphological analysis and the identification of relationships between the object, subject and/or spatial/temporal location within a sentence.

42.4.1 Problem Dependant Task - Information Extraction in Text Mining

Information Extraction (Hobbs *et al.*, 1992; Riloff, 1993; Riloff, 1994; Riloff and Lehnert, 1994; Huffman, 1995; Grishman, 1996; Cardie, 1997; Grishman, 1997; Leek, 1997; Wilks, 1997; Freitag, 1998), is perhaps the most prominent technique currently used in text mining pre-processing operations. Without IE (Information Extraction) techniques, text mining systems would have much more limited knowledge discovery capabilities.

IE technology would allow one to rapidly create extraction systems for new tasks whose performance was on par with human performance. However, even systems that do not have anything near perfect recall and precision can be of real value. In such cases, the results of the IE system would need to be fed into an auditing environment that would allow auditors to fix the system's precision (easy) and recall (much harder) errors. These types of systems would also be of value in cases when the vast amount of information does not enable the users to read all of it, and hence even a partially correct IE system would do much better than the option of not the getting any potentially relevant information at all.

In general, IE systems are useful if the following conditions are met:

- The information to be extracted is specified explicitly and no further inference is needed.
- A small number templates are sufficient to summarize the relevant parts of the document.
- The needed information is expressed relatively locally in the text.

As a first step in tagging documents for text mining systems, each document is processed to find (i.e., extract) entities and relationships that are likely to be meaningful and content-bearing. With respect to relationships, what are referred to here are facts or events involving certain entities.

By way of example, a possible **event** may be that a company has entered into a joint venture to develop a new drug. A **fact** may be that a gene causes a certain disease. **Facts** are static in nature and usually do not change; **events** are more dynamic in nature and generally have a specific time stamp associated with them. The extracted information provides more concise and precise data for the mining process than the more naive word-based approaches such as those used for text categorization, and tends to represent concepts and relationships that are more meaningful and relate directly to the examined document's domain.

From text we can extract four basic elements:

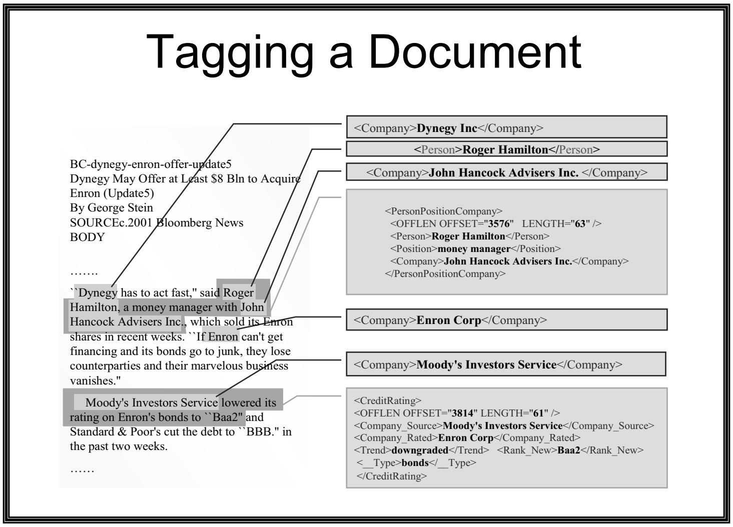
Entities Entities are the basic building blocks that can be found in text documents. Examples include people, companies, locations, genes, drugs, etc.

Attributes Attributes are features of the extracted entities. Examples of attributes might include the title of a person, the age of a person, the type of an organization, etc.

Facts Facts are the relations that exist between entities. Examples could include an employment relationship between a person and a company, Phosphorylation between two proteins, etc.

Event An event is an activity or occurrence of interest in which entities participate. Examples could include a terrorist act, a merger between two companies, a birthday, etc.

Figure 42.4.1 demonstrates of tagged entities and relations from a document:



Tagging a Document

BC-dynegy-enron-offer-update5
Dynegy May Offer at Least $8 Bln to Acquire
Enron (Update5)
By George Stein
SOURCEc.2001 Bloomberg News
BODY

…….

``Dynegy has to act fast," said Roger Hamilton, a money manager with John Hancock Advisers Inc., which sold its Enron shares in recent weeks. ``If Enron can't get financing and its bonds go to junk, they lose counterparties and their marvelous business vanishes."

Moody's Investors Service lowered its rating on Enron's bonds to ``Baa2" and Standard & Poor's cut the debt to ``BBB." in the past two weeks.

……

<Company>**Dynegy Inc**</Company>

<Person>**Roger Hamilton**</Person>

<Company>**John Hancock Advisers Inc.** </Company>

<PersonPositionCompany>
<OFFLEN OFFSET="**3576**" LENGTH="**63**" />
<Person>**Roger Hamilton**</Person>
<Position>**money manager**</Position>
<Company>**John Hancock Advisers Inc.**</Company>
</PersonPositionCompany>

<Company>**Enron Corp**</Company>

<Company>**Moody's Investors Service**</Company>

<CreditRating>
<OFFLEN OFFSET="**3814**" LENGTH="**61**" />
<Company_Source>**Moody's Investors Service**</Company_Source>
<Company_Rated>**Enron Corp**</Company_Rated>
<Trend>**downgraded**</Trend> <Rank_New>**Baa2**</Rank_New>
<__Type>**bonds**</__Type>
</CreditRating>

Fig. 42.3. Tagged Document.

42.5 Formal Frameworks And Algorithm-Based Techniques

42.5.1 Text Categorization

There are two main approaches to the categorization problem. The first approach is the *knowledge engineering* approach where the user is defining manually a set of rules encoding expert knowledge how to classify documents under given categories. The other approach is the *machine learning* approach where a general inductive process automatically builds an automatic text classifier by learning from a set of pre classified documents.

Knowledge Engineering Approach

An example of knowledge engineering approach is the CONSTRUE system (Hayes *et al.*, 1988; Hayes *et al.*, 1990; Hayes and Weinstein 1990; Hayes 1992) built by the Carnegie group for Reuters. A typical rule in the CONSTRUE system:

```
if DNF (disjunction of conjunctive clauses) formula then category else ¬ category
```

An example of this rule being applied might look like the following:

```
If ((wheat & farm) or (wheat & commodity) or (bushels & export) or (wheat & tonnes) or
(wheat & winter & ¬ soft)) then Wheat else ¬ Wheat
```

The main drawback of this approach is what might be referred to as the knowledge acquisition bottleneck. The rules must be manually defined by a knowledge engineer interviewing a domain expert. If the set of categories is modified, then these two professionals must intervene again.

The Machine Learning Approach

The machine learning approach, on the other hand, is based on the existence of a training set of document that are already pre-tagged using the predefined set of categories.

There are two main methods for performing ML based categorization. One method is to perform "Hard" (fully automated) classification where for each pair of category and document we assign a truth value (either TRUE if the document belongs to the category or FALSE otherwise). The other approach is to perform a ranking (semi-automated) based classification. In this approach rather than returning a truth value the classifier return a Categorization Status Value (CSV), i.e. a number between 0 and 1 that represents the evidence for the fact that the document belongs to the category. Documents are then ranked according to their CSV value. Specific text categorization algorithms are discussed below. We will use the following definitions:

- $D = \{d_1, d_2 \ldots .., d_n\}$, the training document collection
- $C = \{c_1, c_2 \ldots .., c_k\}$, the set of possible categories to be assigned to the documents
- $T = \{t_1, t_2 \ldots .., t_m\}$, the set of terms appearing in the documents
- w_{ij}: The weight of the jth term of the ith document
- $CSV_i(d_j)$: A number between 0 and 1 that represents the certainty that a category c_i should be assigned to document d_j
- $DisD_i, D_j)$: The distance between document D_i and D_j. This number represents the similarity between the documents.
- Probabilistic Classifiers view $CSV_i(d_j)$ in terms of $P(c_i|d_j)$, i.e. the probability that a document represented by a vector $\overrightarrow{d_j}$=¡w_{1j}, . . . , w_{mj}¿ of (binary or weighted) terms belongs to c_i, and compute this probability by an application of Bayes' theorem:

  $$P(c_i|\overrightarrow{d_j}) = \frac{P(c_i)P(\overrightarrow{d_j}|c_i)}{P(\overrightarrow{d_j})}.$$

  In order to compute $P(d_j)$ and $P(d_j|\mathrm{C}_i)$ we need to make the assumption that any two coordinates of the document vector are, when viewed as random variables, statistically independent of each other; this independence assumption is encoded by the equation:

  $$P(\overrightarrow{d_j}|c_i) = \prod_{k=1}^{|T|} p(w_{kj}|c_i).$$

- Example-based Classifiers do not build an explicit, declarative representation of the category c_i, but rely on the category labels attached to the training documents similar to the test document. These methods have thus been called lazy learners, since they defer the decision on how to generalize beyond the training data until each new query instance is encountered. The most prominent example of example-based classifier is KNN (K-Nearest-Neighbor).

  For deciding whether $d_j \in c_i$, k-NN looks at whether the k training documents most similar to d_j also are in c_i; if the answer is positive for a large enough proportion of them, a positive decision is taken, and a negative decision is taken otherwise. Distance-weighted version of k-NN is a variation of K-NN such that we weight the contribution of each neighbor by its similarity with the test document. Classifying d_j by means of k-NN thus comes down to computing

  $$CSV_i(d_j) = \sum_{d_z \in Tr_k(d_j)} Dis(d_j, d_z) \cdot C_i(d_z).$$

  One interesting problem is how to pick the best value for k. Larkey and Croft (1996) use k = 20, while Yang (2001) has found 30 $\leq$ k $\leq$ 45 to yield the best effectiveness. Various experiments have shown that increasing the value of k does not significantly degrade the performance.
- Propositional Rules Learners – There is a family of algorithms that try to learn the propositional definition of the category. One of the prominent examples of

this family of algorithms is Ripper (Cohen, 1995a; Cohen, 1995b, Cohen and Singer, 1996). Ripper learns rules that are disjunctions of conjunctions.

- Support Vector Machines – The support vector machine (SVM) algorithm was proven to be very fast and effective for text classification problems (Drucker *et al.*, 1999; Taira and Haruno, 1999; Joachims, 2000; Takamura and Matsumoto, 2001). SVMs were introduced by Vapnik in his work on structural risk minimization (Vapnik, 1995).

  A linear SVM is a hyperplane that separates with the maximum margin a set of positive examples from a set of negative examples. The margin is the distance from the hyperplane to the nearest example from the positive and negative sets.

Further Reading: Text Categorization

Sebastiani (2002) provides an excellent tutorial on text categorization.

- Papers that discuss various algorithms to text categorization include (Apte *et al.*, 1994; Cavnar and Trenkle, 1994; Iwayama and Tokunaga, 1994; Lewis and Ringuette, 1994; Yang and Chute, 1994; Cohen, 1995a; Goldberg, 1995; Cohen and Singer, 1996; Lam *et al.*, 1997; Ruiz and Srinivasan, 1997; Attardi *et al.*, 1998; Dumais *et al.*, 1998; Joachims, 1998; Kwok, 1998; Lam and Ho, 1998; Yavuz and Guvenir, 1998; Jo, 1999; Ruiz and Srinivasan, 1999; Taira and Haruno, 1999; Weigend *et al.*, 1999; Yang and Liu, 1999; Chen and Ho, 2000; D'Alessio *et al.*, 2000; Frank *et al.*, 2000; Junker *et al.*, 2000; Ko and Seo, 2000; Lewis, 2000; Siolas and d'Alche-Buc, 2000; Bao *et al.*, 2001; Ferilli *et al.*, 2001; Sable and Church, 2001; Soucy and Mineau, 2001; Tan, 2001; Vert, 2001; Yang, 2001; Zhang and Oles, 2001; Ko *et al.*, 2002; Ko and Seo, 2002; Leopold and Kindermann, 2002; Tan *et al.*, 2002; Bigi, 2003; Zhang *et al.*, 2003; Zhang and Yang, 2003).
- Approaches that combine several algorithms by using committees of algorithms or by using boosting are described in (Larkey and Croft, 1996; Liere and Tadepalli, 1997; Liere and Tadepalli, 1998; Forsyth, 1999; Ruiz and Srinivasan, 1999; Schapire and Singer, 2000; Sebastiani *et al.*, 2000; Al-Kofahi *et al.*, 2001; Bao *et al.*, 2001; Lam and Lai, 2001; Taira and Haruno, 2001; Nardiello *et al.*, 2003)
- Approaches that integrate linguistic knowledge and background knowledge into the categorization process can be found in ..(Jacobs 1992; Rodriguez, Gomez-Hidalgo *et al.*, 1997; Aizawa, 2001; Benkhalifa *et al.*, 2001a; Benkhalifa *et al.*, 2001b)
- Applications of text categorization are described in (Hayes *et al.*, 1988; Ittner *et al.*, 1995; Larkey, 1998; Lima *et al.*, 1998; Attardi *et al.*, 1999; Drucker *et al.*, 1999; Moens and Dumortier, 2000; Yang *et al.*, 2000; Gentili *et al.*, 2001; Krier and Zacc'a, 2002; Fall *et al.*, 2003; Giorgetti and Sebastiani, 2003; Giorgetti and Sebastiani, 2003)

42.5.2 Probabilistic models for Information Extraction

Probabilistic models often show better accuracy and robustness against the noise than categorical models. The ultimate reason for this is not quite clear, and can be an excellent subject for a philosophical debate.

Nevertheless, several probabilistic models have turned out to be especially useful for the different tasks in extracting meaning from natural language texts. Most prominent among these probabilistic approaches are Hidden Markov Models ("HMMs"), Stochastic Context-Free Grammars ("SCFG"), and Maximal Entropy ("ME").

Hidden Markov Models

HMM is a finite state automaton with stochastic state transitions and symbol emissions (Rabiner 1989). The automaton models a probabilistic generative process. In this process a sequences of symbols is produced by starting in an initial state, transitioning to a new state, emitting a symbol selected by the state and repeating this transition/emission cycle until a designated final state is reached.

Leek used the HMM for IE of gene names and location from scientific abstracts (Leek, 1997). In his work, leek build a HMM that classifies and parses natural language assertions about genes being located at a particular position on chromosomes. The HMM was trained on a small set of sentences fragments chosen from the collected scientific abstracts. Leek got precision of 80%. The HMM approach, in contrast with the traditional NLP methods, make no use of part-of-speech taggers or dictionaries, just using non-emitting states to assemble modules roughly corresponding to noun, verb and prepositional phrase.

The NYMBLE system, which was build for the MUC-6 task, (Bikel *et al.*, 1997) used the HMM approach for extracting names out of text. They created a HMM with only eight internal states (the name classes, including the NOT-A-NAME class), with two special state, the START-OF-SENTENCE and the END-OF-SENTENCE state. They trained the model for extracting names in two languages – English and Spanish. They results was the F-measure between 90% and 93%.

Another use of HMM for extracting the names of genes was done by Collier *et al.* (2000). Their research was to automatically extract facts from scientific abstracts and full papers in the molecular-biology domain. They trained the HMM entirely with bigrams based on lexical and character features in a relatively small corpus of 100 MEDLINE abstracts that were marked up by domain experts with term classes such as proteins and DNA. They get 0.73 f-score.

Seymore *et al.* (1999) explore the use of HMM models for IE tasks. They used their model for extracting important fields from the headers of computer science research papers. Their experiments show that HMM models do well at extracting important information from the headers of research papers. They achieved an accuracy of 90.1% over all fields of the header, and 98.3% for titles and 93.2% for authors. They found that the use of distantly-labeled data improved the results of the model by 10.7% accuracy in extracting for headlines.

The above are examples of the researches which has been done to implement the HMM for IE tasks. The results we get for IE by using the HMM are good comparing to other techniques but there are few problems in using HMM.

The main disadvantage of using an HMM for Information extraction is the need for a large amount of training data the more training data we have the better results we get. To build such training data it a time consuming task. We need to do lot of manually tagging which must to be done by experts of the specific domain we are working with.

The second one is that the HMM model is a flat model, so the most it can do is assign a tag to each token in a sentence. This is suitable for the tasks where the tagged sequences do not nest and where there are no explicit relations between the sequences. Part-of-speech tagging and entity extraction belong to this category, and indeed the HMM-based PoS taggers and entity extractors are state-of-the-art. Extracting relationships is different, because the tagged sequences can (and must) nest, and there are relations between them which must be explicitly recognized.

Stochastic Context-Free Grammars

A stochastic context-free grammar (SCFG) (Lari and Young, 1990; Collins, 1996; Kammeyer and Belew, 1996; Keller and Lutz, 1997a; Keller and Lutz, 1997b; Osborne and Briscoe, 1998) is a quintuple $G = (T,N,S,R,P)$, where T is the alphabet of terminal symbols (tokens), N is the set of nonterminals, S is the starting nonterminal, R is the set of rules, and $P : R \rightarrow [0..1]$ defines their probabilities. The rules have the form $n \rightarrow s_1 s_2 \ldots s_k$, where n is a nonterminal and each s_i either token or another nonterminal. As can be seen, SCFG is a usual context-free grammar with the addition of the P function.

Similarly to a canonical (non-stochastic) grammar, SCFG is said to *generate* (or *accept*) a given string (sequence of tokens) if the string can be produced starting from a sequence containing just the starting symbol S, and one by one expanding nonterminals in the sequence using the rules from the grammar. The particular way the string was generated can be naturally represented by a *parse tree* with the starting symbol as a root, nonterminals as internal nodes and the tokens as leaves.

The semantics of the probability function P is straightforward. If r is the rule $n \rightarrow s_1 s_2 \ldots s_k$, then $P(r)$ is the frequency of expanding n using this rule. Or, in Bayesian terms, if it is known that a given sequence of tokens was generated by expanding n, then $P(r)$ is the apriori likelihood that n was expanded using the rule r. Thus, it follows that for every nonterminal n the sum $\sum P(r)$ of probabilities of all rules r headed by n must equal to one.

Maximal Entropy Modelling

Consider a random process of an unknown nature which produces a single output value y, a member of a finite set Y of possible output values. The process of generating y may be influenced by some contextual information x, a member of the set X

of possible contexts. The task is to construct a statistical model that accurately represents the behavior of the random process. Such a model is a method of estimating the conditional probability of generating y given the context x.

Let $P(x,y)$ be denoted as the unknown true joint probability distribution of the random process, and $p(y|x)$ the model we are trying to build, taken from the class $\wp$ of all possible models. In order to build the model we are given a set of training samples, generated by observing the random process for some time. The training data consists of a sequence of pairs $(x_i$, $\mathrm{y}_i)$ of different outputs produced in different contexts.

In many interesting cases the set X is too large and underspecified to be directly used. For instance, X may be the set of all dots "." in all possible English texts. For contrast, the Y may be extremely simple, while remaining interesting. In the above case, the Y may contain just two outcomes: "SentenceEnd" and "NotSentenceEnd". The target model $p(y|x)$ would in this case solve the problem of finding sentence boundaries.

In cases like that it is impossible to directly use the context x to generate the output y. However, there are usually many regularities and correlations, which can be exploited. Different contexts are usually similar to each other in all manner of ways, and similar contexts tend to produce similar output distributions (Berger *et al.*, 1996; Ratnaparkhim, 1996; Rosenfeld, 1997; McCallum *et al.*, 2000; Hopkins and Cui, 2004).

42.6 Hybrid Approaches - TEG

The knowledge engineering (mostly rule based) systems traditionally were the top performers in most IE benchmarks, such as MUC (Chinchor *et al.*, 1994), ACE (ACE, 2002) and the KDD CUP (Yeh *et al.*, 2002). Recently though, the machine learning systems became state-of-the-art, especially for simpler tagging problems, such as named entity recognition (Bikel, *et al.*, 1999; Chieu and Ng, 2002), or field extraction (McCallum *et al.*, 2000).

Still, the knowledge engineering approach retains some of its advantages. It is focused around manually writing patterns to extract the entities and relations. The patterns are naturally accessible to human understanding, and can be improved in a controllable way. Whereas, improving the results of a pure machine learning system, would require providing it with additional training data. However, the impact of adding more data soon becomes infinitesimal while the cost of manually annotating the data grows linearly.

TEG (Rosenfeld *et al.*, 2004) is a hybrid entities and relations extraction system, which combines the power of knowledge-based and statistical machine learning approaches. The system is based upon SCFGs. The rules for the extraction grammar are written manually, while the probabilities are trained from an annotated corpus. The powerful disambiguation ability of PCFGs allows the knowledge engineer to write very simple and naive rules while retaining their power, thus greatly reducing the required labor.

In addition, the size of the needed training data is considerably smaller than the size of the training data needed for pure machine learning system (for achieving comparable accuracy results). Furthermore, the tasks of rule writing and corpus annotation can be balanced against each other.

Although the formalisms based upon probabilistic finite-state automata are quite successful for entity extraction, they have shortcomings, which make them harder to use for the more difficult task of extracting relationships.

One problem is that a finite-state automaton model is flat, so its natural task is assignment of a tag (state label) to each token in a sequence. This is suitable for the tasks where the tagged sequences do not nest and where there are no explicit relations between the sequences. Part-of-speech tagging and entity extraction tasks belong to this category, and indeed the HMM-based PoS taggers and entity extractors are state-of-the-art.

Extracting relationships is different in that the tagged sequences can and must nest, and there are relations between them, which must be explicitly recognized. While it is possible to use nested automata to cope with this problem, we felt that using more general context-free grammar formalism would allow for a greater generality and extendibility without incurring any significant performance loss.

42.7 Text Mining – Visualization and Analytics

One of the crucial needs in text mining process is the ability enables the user to visualize relationships between entities that were extracted from the documents. This type of interactive exploration enables one to identify new types of entities and relationships that can be extracted and, better explore the results of the information extraction phase. There are tools that can do the analytic and visualization task, the first is Clear Research (Aumann *et al.*, 1999; Feldman*et al.*, 2001; Feldman *et al.*, 2002).

42.7.1 Clear Research

Clear Research has five different visualization tools to analyze the entities and relationships. The following subsections present each one of them.

Category Connection Map

Category Connection Maps provide a means for concise visual representation of connections between different categories, e.g. between companies and technologies, countries and people, or drugs and diseases. The system finds all the connections between the terms in the different categories. To visualize the output, all the terms in the chosen categories are depicted on a circle, with each category placed on a separate part on the circle. A line is depicted between terms of different categories which are related. A color coding scheme represents stronger links with darker colors. An example of a Category Connection Map is presented in Figure 42.4. In this chapter we used a text collection (1354 documents) from yahoo-news about Bin Laden organization. In Figure 42.4 we can see the connection between Persons and Organizations.

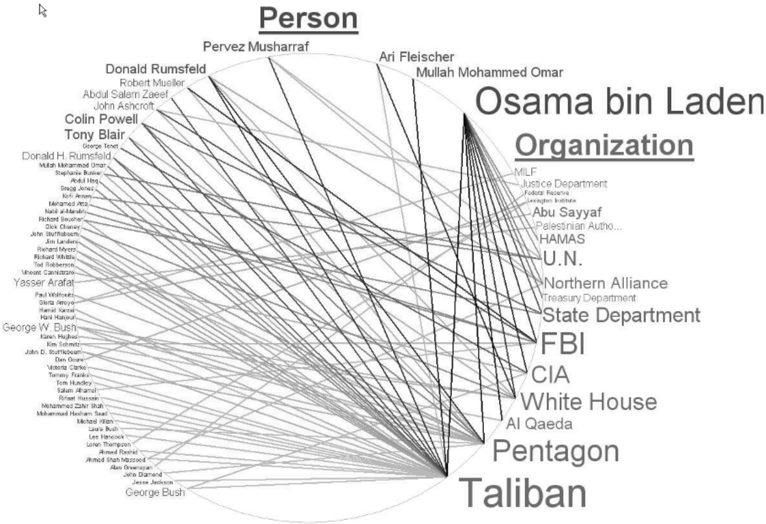


Fig. 42.4. Category map – connections between Persons and Organizations

Relationship Maps

Relationship maps provide a visual means for concise representation of the relationship between many terms in a given context. In order to define a relationship map the user defines:

- A taxonomy category (e.g. “companies”), which determines the nodes of the circle graph (e.g. companies)
- An optional context node (e.g. “joint venture”): which will determine the type of connection we wish to find among the graph nodes.

In Figure 42.5 we can see an example of relations map between Persons. The graph gives the user a summary of the entire collection in one view. The user can appreciate the overall structure of the connections between persons in this context, even before reading a single document!

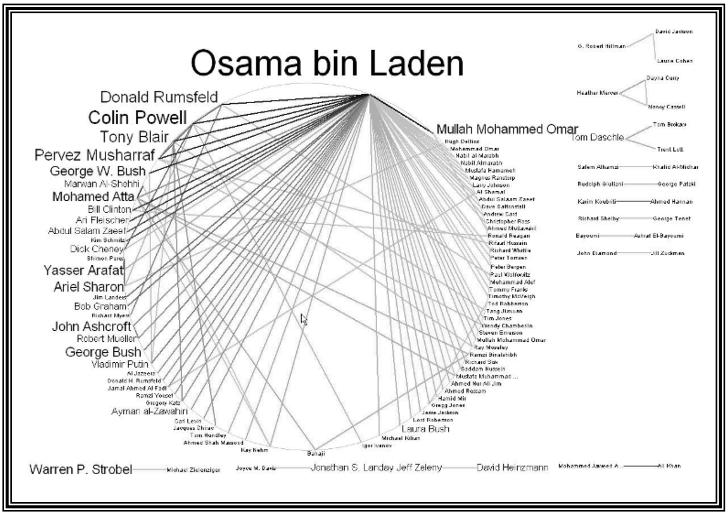


Fig. 42.5. Relationship map– relations between Persons

Spring Graph

A spring graph is a 2D graph where the distance between 2 elements should reflect the strength of the relationships between the elements. The stronger the relationship the closer the two elements should be. An example of a spring graph is shown in Figure 42.6. The graph represents the relationships between the people in a document collection. We can see that Osama Bin Laden is at the center connected to many of the other key players related to the tragic events.

Link Analysis

This query enables users to find interesting but previously unknown implicit information within the data. The Links Analysis query automatically organizes links (associations) between entities that are not present in individual documents. The results of a link analysis query can give new insight into the data and interprets the relevant interconnections between entities.

The Links Analysis query results graphically illustrate the links that indicate the associations among the selected entities. The results screen arranges the source and

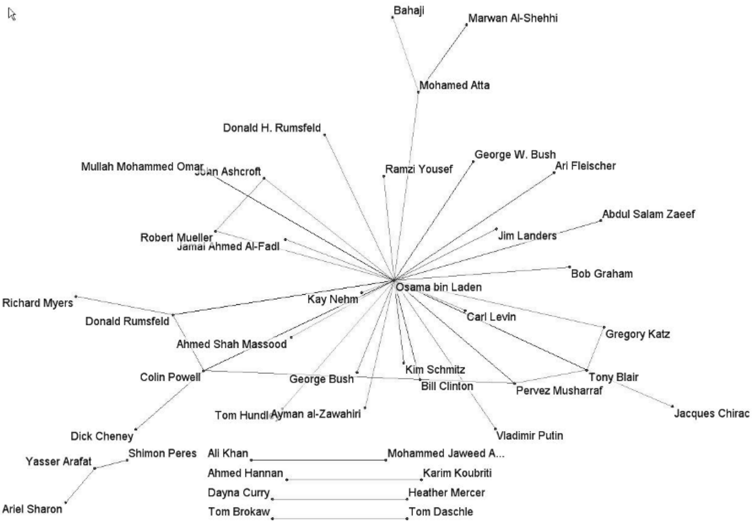


Fig. 42.6. Spring Graph

destination nodes at opposite ends and places the connecting nodes between them enabling users to follow the path that links the nodes together. The Links Analysis query is useful to users that require a graphical analysis that charts the interconnections among entities through implicit channels.

The Link Analysis query implicitly illustrates inter-relationships between entities. Users define the query criterion by defining the: source, destination and connection through entities. In this manner - the results, if any relations are found, will display the defined entities and the paths that show how they connect to one another, e.g. through third party or more entities.

In Figure 42.7 we can see a link analysis query about relation between Osama Bin Laden and John Paul II. We can see that there is no direct connection between the two but we can find indirect connection between them.

For more information regarding Link Analysis please refer to Chapter 17.5 in this volume.

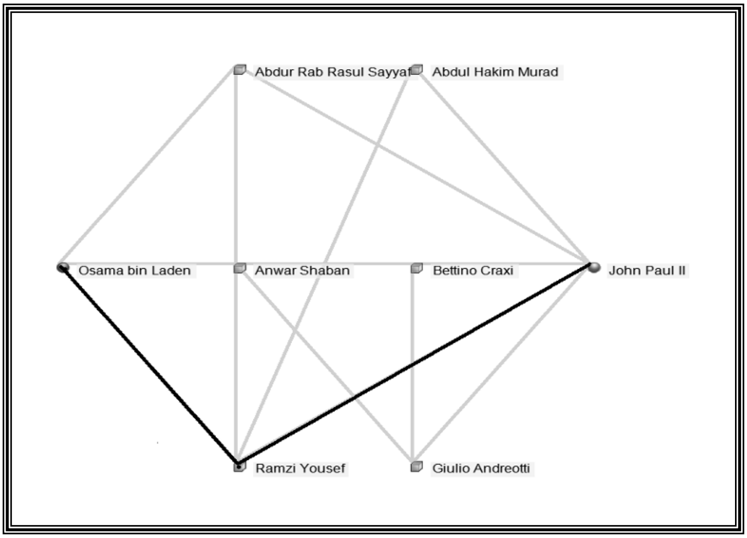


Fig. 42.7. Link Analysis – relations between Bin Laden and John Paul II.

42.7.2 Other Visualization and Analytical Approaches

The BioTeKS is an IBM prototype system for text analysis, search, and text-mining methods to support problem solving in life science, which was build by several groups in the IBM Research Division. The system is called “BioTeKS” (“Biological Text Knowledge Services”), and it integrates research technologies from multiple IBM Research labs (Mack *et al.*, 2004)

The SPIRE text visualization system, which images information from free text documents as natural terrains, serves as an example of the “ecological approach” in its visual metaphor, its text analysis, and its specializing procedures (Wise, 1999).

The ThemeRiver visualization depicts thematic variations over time within a large collection of documents. The thematic changes are shown in the context of a time line and corresponding external events. The focus on temporal thematic change within a context framework allows a user to discern patterns that suggest relationships or trends. For example, the sudden change of thematic strength following an external event may indicate a causal relationship. Such patterns are not readily accessible in other visualizations of the data (Havre *et al.*, 2002).

An approach for visualization technique of association rules is described in the following article (Wong *et al.*, 1999). We can find a technique for visualizing Sequential Patterns was describe in the work done by the Pacific Northwest National Laboratory (Wong *et al.*, 2000).

References

ACE (2002). http://www.itl.nist.gov/iad/894.01/tests/ace/. ACE - Automatic Content Extraction.

Aizawa, A. (2001). Linguistic Techniques to Improve the Performance of Automatic Text Categorization. Proceedings of NLPRS-01, 6th Natural Language Processing Pacific Rim Symposium. Tokyo, JP: 307-314.

Al-Kofahi, K., A. Tyrrell, A., Vachher, A., Travers, T., and Jackson (2001). Combining Multiple Classifiers for Text Categorization. Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management. H. P. a. L. L. a. D. Grossman. Atlanta, US, ACM Press, New York, US: 97-104.

Apte, C., Damerau, F. J., and Weiss, S. M. (1994). Automated learning of decision rules for text categorization. ACM Transactions on Information Systems, 12(3): 233-251.

Attardi, G., Gulli, A., and Sebastiani, F. (1999). Automatic Web Page Categorization by Link and Context Analysis. In C. H. a. G. Lanzarone (Ed.), Proceedings of THAI-99, 1st European Symposium on Telematics, Hypermedia and Artificial Intelligence: 105-119. Varese,

Attardi, G., Marco, S. D., and Salvi, D. (1998). Categorization by context. Journal of Universal Computer Science, 4(9): 719-736.

Aumann Y., Feldman R., Ben Yehuda Y., Landau D., Lipshtat O., and Y, S. (1999). Circle Graphs: New Visualization Tools for Text-Mining. Paper presented at the PKDD.

Averbuch, M., Karson, T., Ben-Ami, B., Maimon, O., and Rokach, L. (2004). Context-sensitive medical information retrieval, MEDINFO-2004, San Francisco, CA, September. IOS Press, pp. 282-262.

Bao, Y., Aoyama, S., Du, X., Yamada, K., and Ishii, N. (2001). A Rough Set-Based Hybrid Method to Text Categorization. In M. T. O. a. H.-J. S. a. K. T. a. Y. Z. a. Y. Kambayashi (Ed.), Proceedings of WISE-01, 2nd International Conference on Web Information Systems Engineering: 254-261. Kyoto, JP: IEEE Computer Society Press, Los Alamitos, US.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). Modern Information Retrieval, Addison-Wesley.

Benkhalifa, M., Mouradi, A., and Bouyakhf, H. (2001a). Integrating External Knowledge to Supplement Training Data in Semi-Supervised Learning for Text Categorization. Information Retrieval, 4(2): 91-113.

Benkhalifa, M., Mouradi, A., and Bouyakhf, H. (2001b). Integrating WordNet knowledge to supplement training data in semi-supervised agglomerative hierarchical clustering for text categorization. International Journal of Intelligent Systems, 16(8): 929-947.

Berger, A. L., Della Pietra, S. A., and Della Pietra, V. J. (1996). A maximum entropy approach to natural language processing. Computational Linguistics, 22.

Bigi, B. (2003). Using Kullback-Leibler distance for text categorization. Proceedings of ECIR-03, 25th European Conference on Information Retrieval. F. Sebastiani. Pisa, IT, Springer Verlag: 305-319.

Bikel, D. M., S. Miller, *et al.* (1997). Nymble: a high-performance learning name-finder. Proceedings of ANLP-97: 194-201.

Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: a high-performance learning name-finder, Proceedings of ANLP-97: 194-201.

Brill, E. (1992). A simple rule-based part of speech tagger. Third Annual Conference on Applied Natural Language Processing, ACL.

Brill, E. (1995). ”Transformation-based Error-driven Learning and Natural Language Processing: A Case Study in Part-Of-Speech Tagging.” Computational Linguistics, 21(4): 543-565.

Cardie, C. (1997). ”Empirical Methods in Information Extraction.” AI Magazine, 18(4): 65-80.

Cavnar, W. B. and J. M. Trenkle (1994). N-Gram-Based Text Categorization. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval. Las Vegas, US: 161-175.

Chen, H. and S. T. Dumais (2000). Bringing order to the Web: automatically categorizing search results. Proceedings of CHI-00, ACM International Conference on Human Factors in Computing Systems. Den Haag, NL, ACM Press, New York, US: 145-152.

Chen, H. and T. K. Ho (2000). Evaluation of Decision Forests on Text Categorization. Proceedings of the 7th SPIE Conference on Document Recognition and Retrieval. San Jose, US, SPIE - The International Society for Optical Engineering: 191-199.

Chieu, H. L. and H. T. Ng (2002). Named Entity Recognition: A Maximum Entropy Approach Using Global Information. Proceedings of the 17th International Conference on Computational Linguistics.

Chinchor, N., Hirschman, L., and Lewis, D. (1994). Evaluating Message Understanding Systems: An Analysis of the Third Message Understanding Conference (MUC-3). Computational Linguistics, 3(19): 409-449.

Cohen, W. and Y. Singer (1996). Context Sensitive Learning Methods for Text categorization. SIGIR’96.

Cohen, W. W. (1995a). Learning to classify English text with ILP methods. Advances in inductive logic programming. L. D. Raedt. Amsterdam, NL, IOS Press: 124-143.

Cohen, W. W. (1995b). Text categorization and relational learning. Proceedings of ICML-95, 12th International Conference on Machine Learning. Lake Tahoe, US, Morgan Kaufmann Publishers, San Francisco, US: 124-132.

Collier, N., Nobata, C., and Tsujii, J. (2000). Extracting the names of genes and gene products with a Hidden Markov Model.

Collins, M. J. (1996). A neew statistical parser based on bigram lexical dependencies. 34 th Annual Meeting of the Association for Computational Linguistics., university of California, Santa Cruz USA.

Cutting, D. R., Pedersen, J. O., Karger, D., and Tukey., J. W. 1992. Scatter/Gather: A cluster-based approach to browsing large document collections. Paper presented at the In Proceedings of the 15th Annual International ACM/SIGIR Conference, pages 318-329, Copenhagen, Denmark.

D’Alessio, S., Murray, K., Schiaffino, R., and Kershenbaum, A. 2000. The effect of using Hierarchical classifiers in Text Categorization, Proceeding of RIAO-00, 6th International Conference “Recherche d’Information Assistee par Ordinateur”: 302-313

Dorre, J., Gerstl, P., and Seiffert, R. (1999). Text mining: finding nuggets in mountains of textual data, Proceedings of KDD-99, 5th ACM International Conference on Knowledge Discovery and Data Mining: 398-401. San Diego, US: ACM Press, New York, US.

Drucker, H., Vapnik, V., and Wu, D. (1999). Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 10(5): 1048-1054.

Dumais, S. T., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. Paper presented at the Seventh International Conference on Information and Knowledge Management (CIKM’98).

Fall, C. J., Torcsvari, A., Benzineb, K., and Karetka, G. (2003). Automated Categorization in the International Patent Classification. SIGIR Forum, 37(1).

Feldman, R., Aumann, Y., Finkelstein-Landau, M., Hurvitz, E., Regev, Y., and Yaroshevich, A. (2002). A Comparative Study of Information Extraction Strategies, CICLing: 349-359.

Feldman, R., Aumann, Y., Liberzon, Y., Ankori, K., Schler, J., and Rosenfeld, B. (2001). A Domain Independent Environment for Creating Information Extraction Modules., CIKM: 586-588.

Feldman, R., Fresko, M., Kinar, Y., Lindell, Y., Liphstar, O., Rajman, M., Schler, Y., and Zamir, O. (1998). Text Mining at the Term Level. Paper presented at the In Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery, Nantes, France.

Ferilli, S., Fanizzi, N., and Semeraro, G. (2001). Learning logic models for automated text categorization. In F. Esposito (Ed.), Proceedings of AI*IA-01, 7th Congress of the Italian Association for Artificial Intelligence: 81-86. Bari, IT: Springer Verlag, Heidelberg, DE.

Forsyth, R. S. (1999). New directions in text categorization. Causal models and intelligent data management. A. Gammerman. Heidelberg, DE, Springer Verlag: 151-185.

Frank, E., Chui, C., and Witten, I. H. (2000). Text Categorization Using Compression Models. In J. A. S. a. M. Cohn (Ed.), Proceedings of DCC-00, IEEE Data Compression Conference: 200-209.

Freitag, D. (1998). Machine Learning for Information Extraction in Informal Domains. Computer Science Department. Pittsburgh, PA, Carnegie Mellon University: 188.

Gentili, G. L., Marinilli, M., Micarelli, A., and Sciarrone, F. 2001. Text categorization in an intelligent agent for filtering information on the Web. International Journal of Pattern Recognition and Artificial Intelligence, 15(3): 527-549.

Giorgetti, D. and F. Sebastiani (2003). ”Automating Survey Coding by Multiclass Text Categorization Techniques.” Journal of the American Society for Information Science and Technology, 54(12): 1269-1277.

Giorgetti, D. and F. Sebastiani (2003). Multiclass Text Categorization for Automated Survey Coding. Proceedings of SAC-03, 18th ACM Symposium on Applied Computing. Melbourne, US, ACM Press, New York, US: 798-802.

Goldberg, J. L. (1995). CDM: an approach to learning in text categorization. Proceedings of ICTAI-95, 7th International Conference on Tools with Artificial Intelligence. Herndon, US, IEEE Computer Society Press, Los Alamitos, US: 258-265.

Grishman, R. (1996). The role of syntax in Information Extraction. Advances in Text Processing: Tipster Program Phase II, Morgan Kaufmann.

Grishman, R. (1997). Information Extraction: Techniques and Challenges. SCIE: 10-27.

Hammerton, J., Miles Osborne, Susan Armstrong, and Daelemans, W. 2002. Introduction to the Special issue on Machine Learning Approaches to Shallow Parsing. Journal of Machine Learning Research, 2(Special Issue Website): 551-558.

Havre S., Hetzler E., Whitney P., and Nowell L., (2002). ”ThemeRiver: Visualizing Thematic Changes in Large Document Collections.” IEEE Transactions on Visualization and Computer Graphics, 8(1): 9-20.

Hayes, P. (1992). Intelligent High-Volume Processing Using Shallow, Domain-Specific Techniques. Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval: 227-242.

Hayes, P. J., Andersen, P. M., Nirenburg, I. B., and Schmandt, L. M. (1990). Tcs: a shell for content-based text categorization, Proceedings of CAIA-90, 6th IEEE Conference on Artificial Intelligence Applications: 320-326. Santa Barbara, US: IEEE Computer Society Press, Los Alamitos, US..

Hayes, P. J., Knecht, L. E., and Cellio, M. J. (1988). A news story categorization system, Proceedings of ANLP-88, 2nd Conference on Applied Natural Language Processing: 9-17. Austin, US: Association for Computational Linguistics, Morristown, US.

Hayes, P. J. and S. P. Weinstein (1990). Construe/Tis: a system for content-based indexing of a database of news stories. Proceedings of IAAI-90, 2nd Conference on Innovative Applications of Artificial Intelligence. AAAI Press, Menlo Park, US: 49-66.

Hearst, M. A. (1999). Untangling Text Data Mining. Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics, University of Maryland.

Hobbs, J. R., Appelt, D. E., John Bear, D. I., Kameyama, M., and Tyson, M. (1992). FASTUS: A System for Extracting Information from Text. Paper presented at the Human Language Technology.

Hopkins, J. and J. Cui (2004). Maximum Entropy Modeling in Sparse Semantic Tagging, NSF grant numbers IIS- 0121285.

Huffman, S. B. (1995). Learning information extraction patterns from examples. Learning for Natural Language Processing: 246-260.

Ittner, D. J., Lewis, D. D., and Ahn, D. D. (1995). Text categorization of low quality images, Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval: 301-315. Las Vegas, US.

Iwayama, M. and T. Tokunaga (1994). A Probabilistic Model for Text Categorization Based on a Single Random Variable with Multiple Values. In Proceedings of the 4th Conference on Applied Natural Language Processing.

Jacobs, P. (1992). Joining Statistics with NLP for Text Categorization. In Proceedings of the 3rd Conference on Applied Natural Language Processing.

Jo, T. C. (1999). Text categorization with the concept of fuzzy set of informative keywords. Proceedings of FUZZ-IEEE'99, IEEE International Conference on Fuzzy Systems. Seoul, KR, IEEE Computer Society Press, Los Alamitos, US: 609-614.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. Proceedings of ECML-98, 10th European Conference on Machine Learning. Chemnitz, DE, Springer Verlag, Heidelberg, DE: 137-142.

Joachims, T. (2000). Estimating the Generalization Performance of a SVM Efficiently. Proceedings of ICML-00, 17th International Conference on Machine Learning. P. Langley. Stanford, US, Morgan Kaufmann Publishers, San Francisco, US: 431-438.

Junker, M., Sintek, M., and Rinck, M. (2000). Learning for text categorization and information extraction with ILP. In J. C. a. S. Dzeroski (Ed.), Proceedings of the 1st Workshop on Learning Language in Logic: 247-258. Bled, SL: Springer Verlag, Heidelberg, DE.

Kammeyer, T. and Belew, R. K. (1996). Stochastic Context-Free Grammar Induction with a Genetic Algorithm Using Local Search. Foundations of Genetic Algorithms, Morgan Kaufmann.

Keller, B. (1992). A Logic for Representing Grammatical Knowledge. European Conference on Artificial Intelligence: 538-542, European Conference on Artificial Intelligence.

Keller B. and Lutz R. (1997a). Evolving stochastic context-free grammars from examples using a minimum description length principle. Workshop on Automata Induction Grammatical Inference and Language Acquisition, ICML-97, Nashville, Tennessee.

Keller, B. and R. Lutz (1997b). Learning stochastic context-free grammars from corpora using a genetic algorithm. International conference on Artificial Neural Networks and Genetic Algorithms.

Kleinberg, J. M. (1999). "Authoritative sources in a hyperlinked environment." Journal of the ACM, 46(5): 604-632.

Ko, Y., Park, J., and Seo, J. (2002). Automatic Text Categorization using the Importance of Sentences, Proceedings of COLING-02, the 19th International Conference on Computational Linguistics. Taipei, TW.

Ko, Y. and J. Seo (2000). Automatic Text Categorization by Unsupervised Learning. Proceedings of COLING-00, the 18th International Conference on Computational Linguistics. Saarbrucken, DE.

Ko, Y. and J. Seo (2002). Text Categorization using Feature Projections. Proceedings of COLING-02, the 19th International Conference on Computational Linguistics. Taipei, TW.

Krier, M. and F. Zacc'a (2002). ”Automatic categorization applications at the European Patent Office.” World Patent Information, 24: 187-196.

Kupiec, J. (1992). ”Robust Part-of-speech tagging using a hidden Markov model.” Computer Speech and Language, 6.

Kwok, J. T. (1998). Automated text categorization using support vector machine. Proceedings of ICONIP'98, 5th International Conference on Neural Information Processing. Kitakyushu, JP: 347-351.

Lam, W. and C. Y. Ho (1998). Using a generalized instance set for automatic text categorization. Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval. Melbourne, AU, ACM Press, New York, US: 81-89.

Lam, W. and K.-Y. Lai (2001). A Meta-Learning Approach for Text Categorization. Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval. New Orleans, US, ACM Press, New York, US: 303-309.

Lam, W., Low, K. F., and Ho, C. Y. (1997). Using a Bayesian Network Induction Approach for Text Categorization. In M. E. Pollack (Ed.), Proceedings of IJCAI-97, 15th International Joint Conference on Artificial Intelligence: 745-750. Nagoya, JP: Morgan Kaufmann Publishers, San Francisco, US.

Lari, K. and Young, S. J. (1990). ”The estimation of stochastic context-free grammars using the Inside-Outside algorithm.” Computer Speech and Language, 4: 35–56.

Larkey, L. S. (1998). Automatic essay grading using text categorization techniques. Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval. Melbourne, AU, ACM Press, New York, US: 90-95.

Larkey, L. S. and W. B. Croft (1996). Combining classifiers in text categorization. Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval. Zurich, CH, ACM Press, New York, US: 289-297.

Leek, T. R. (1997). ”Information extraction using hidden Markov models.”

Leopold, E. and J. Kindermann (2002). ”Text Categorization with Support Vector Machines: How to Represent Texts in Input Space?” Machine Learning, 46(1/3): 423-444.

Lewis, D. D. (2000). Machine learning for text categorization: background and characteristics. Proceedings of the 21st Annual National Online Meeting. M. E. Williams. New York, US, Information Today, Medford, USA: 221-226.

Lewis, D. D. and M. Ringuette (1994). A comparison of two learning algorithms for text categorization. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval. Las Vegas, US: 81-93.

Liere, R. and P. Tadepalli (1997). Active learning with committees for text categorization. Proceedings of AAAI-97, 14th Conference of the American Association for Artificial Intelligence. Providence, US, AAAI Press, Menlo Park, US: 591-596.

Liere, R. and P. Tadepalli (1998). Active Learning with Committees: Preliminary Results in Comparing Winnow and Perceptron in Text Categorization. Proceedings of CONALD-

98, 1st Conference on Automated Learning and Discovery. Pittsburgh, US, AAAI Press, Menlo Park, US.

Lima, L. R. D., Laender, A. H., and Ribeiro-Neto, B. A. (1998). A hierarchical approach to the automatic categorization of medical documents. In L. Bouganim (Ed.), Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management: 132-139. Bethesda, US: ACM Press, New York, US.

Mack R., Mukherjea S., A. Soffer, N. Uramoto, E. Brown, A. Coden, J. Cooper, A. Inokuchi, B. Iyer, Y. Mass, H. Matsuzawa, and Subramaniam, L. V (2004). "Text analytics for life science using the Unstructured Information Management Architecture." IBN systems journal, 43.

McCallum, A., Freitag, D., and Pereira, F. (2000a). Maximum Entropy Markov Models for Information Extraction and Segmentation, Proc. 17th International Conf. on Machine Learning: 591-598: Morgan Kaufmann, San Francisco, CA.

McCallum, A., Freitag, D., and Pereira, F. (2000b). Maximum Entropy Markov Models for Information Extraction and Segmentation. Paper presented at the Proceedings of the 17th International Conference on Machine Learning.

Moens, M.-F. and J. Dumortier (2000). "Text categorization: the assignment of subject descriptors to magazine articles." Information Processing and Management, 36(6): 841-861.

Nardiello, P., Sebastiani, F., and Sperduti, A. (2003). Discretizing continuous attributes in AdaBoost for text categorization. In F. Sebastiani (Ed.), Proceedings of ECIR-03, 25th European Conference on Information Retrieval: 320-334. Pisa, IT: Springer Verlag.

Neuhaus, P. and N. Broker (1997). The Complexity of Recognition of Linguistically Adequate Dependency Grammars. Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics., New Jersey.

Osborne M. and Briscoe T. (1998). Learning Stochastic Categorial Grammars. Computational Natural Language Learning, Association for Computational Linguistics: 80-87.

Pollard, C. and I. A. Sag (1994). "Head-Driven Phrase Structure Grammar." Chicago, Illinois, University of Chicago Press and CSLI Publications.

Rambow, O. and A. K. Joshi (1994). " A Formal Look at Dependency Grammars and Phrase-Structure Grammars, with Special Consideration of Word-Order Phenomena." Current Issues in Meaning-Text Theory. L. Wanner. London, UK, Pinter.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. Proc. EMNLP: Association for Computational Linguistics, New Brunswick, New Jersey.

Riloff, E. (1993a). Automatically Constructing a Dictionary for Information Extraction Tasks. In Proceedings of the Eleventh National Congress on Artificial Intelligence, AAAI Press / MIT Press.

Riloff, E. (1993b). Automatically Constructing a Dictionary for Information Extraction Tasks. National Conference on Artificial Intelligence: 811-816.

Riloff, E. (1994). Information Extraction as a Basis for Portable Text Classification Systems. Amherst, US, Department of Computer Science, University of Massachusetts.

Riloff, E. and W. Lehnert (1994). "Information extraction as a basis for high-precision text classification." ACM Transactions on Information Systems, 12(3): 296-333.

Rodriguez, M. D. B., Gomez-Hidalgo J. M., and Diaz-Agudo, B. (1997). Using WordNet to Complement Training Information in Text Categorization. Proceedings of RANLP-97, 2nd International Conference on Recent Advances in Natural Language Processing. Tzigov Chark, BL.

Rosenfeld, R. (1997). A whole sentence maximum entropy language model. Proceedings of the IEEE Workshop on Speech Recognition and Understanding., Santa Barbara, California.

Rosenfeld B., Feldman R., *et al.* (2004). TEG: a hybrid approach to information extraction. Conference on Information and Knowledge Management, Washington, D.C., USA.

Ruiz, M. E. and P. Srinivasan (1997). Automatic Text Categorization Using Neural Networks. Proceedings of the 8th ASIS/SIGCR Workshop on Classification Research. E. Efthimiadis. Washington, US, American Society for Information Science, Washington, US: 59-72.

Ruiz, M. E. and P. Srinivasan (1999). Combining Machine Learning and Hierarchical Indexing Structures for Text Categorization. Proceedings of the 10th ASIS/SIGCR Workshop on Classification Research. Washington, US, American Society for Information Science, Washington, US.

Ruiz, M. E. and P. Srinivasan (1999). Hierarchical neural networks for text categorization. Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval. Berkeley, US, ACM Press, New York, US: 281-282.

Sable, C. and K. Church (2001). Using Bins to Empirically Estimate Term Weights for Text Categorization. Proceedings of EMNLP-01, 6th Conference on Empirical Methods in Natural Language Processing. Pittsburgh, US, Association for Computational Linguistics, Morristown, US: 58-66.

Schapire, R. E. and Y. Singer (2000). ”BoosTexter: a boosting-based system for text categorization.” Machine Learning, 39(2/3): 135-168.

Sebastiani, F. (2002). ”Machine learning in automated text categorization.” ACM Computing Surveys, 34(1): 1-47.

Sebastiani, F., Sperduti A., and Valdambrini, N. (2000). An improved boosting algorithm and its application to automated text categorization. Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management, US, ACM Press, New York, US: 78-85.

Seymore, K., McCallum A., and Rosenfeld, R. (1999). Learning Hidden Markov Model Structure for Information Extraction. AAAI 99 Workshop on Machine Learning for Information Extraction.

Siolas, G. and F. d’Alche-Buc (2000). Support Vector Machines based on a semantic kernel for text categorization. Proceedings of IJCNN-00, 11th International Joint Conference on Neural Networks. Como, IT, IEEE Computer Society Press, Los Alamitos, US. 5: 205-209.

Soucy, P. and G. W. Mineau (2001). A Simple KNN Algorithm for Text Categorization. Proceedings of ICDM-01, IEEE International Conference on Data Mining. San Jose, CA, IEEE Computer Society Press, Los Alamitos, US: 647-648.

Taira, H. and M. Haruno (1999). Feature selection in SVM text categorization. Proceedings of AAAI-99, 16th Conference of the American Association for Artificial Intelligence. Orlando, US, AAAI Press, Menlo Park, US: 480-486.

Taira, H. and M. Haruno (2001). Text Categorization Using Transductive Boosting. Proceedings of ECML-01, 12th European Conference on Machine Learning. Freiburg, DE, Springer Verlag, Heidelberg, DE: 454-465.

Takamura, H. and Y. Matsumoto (2001). Feature Space Restructuring for SVMs with Application to Text Categorization. Proceedings of EMNLP-01, 6th Conference on Empirical Methods in Natural Language Processing. Pittsburgh, US, Association for Computational Linguistics, Morristown, US: 51-57.

Tan, A.-H. (2001). Predictive Self-Organizing Networks for Text Categorization. Proceedings of PAKDD-01, 5th Pacific-Asia Conferenece on Knowledge Discovery and Data Mining. Hong Kong, CN, Springer Verlag, Heidelberg, DE: 66-77.

Tan, C.-M., Wang, Y.-F., and Lee, C. D. (2002). ”The use of bigrams to enhance text categorization.” Information Processing and Management, 38(4): 529-546.

Tkach, D. (1998). ”Turning information into knowledge.” a white paper from IBM.

Vapnik, V. (1995). The Nature of Statistical Learning Theory, Springer-Verlag.

Vert, J.-P. (2001). Text Categorization Using Adaptive Context Trees. Proceedings of CICLING-01, 2nd International Conference on Computational Linguistics and Intelligent Text Processing. A. Gelbukh. Mexico City, ME, Springer Verlag, Heidelberg, DE: 423-436.

Wai-chiu, W. and A. W.-c. Fu (2000). Incremental Document Clustering for Web Page Classification. In Proceedings of 2000 International Conference on Information Society in the 21st Century: Emerging Technologies and New Challenges (IS2000), Aizu-Wakameatsu City, Fukushima, Japan.

Weigend, A. S., Wiener, E. D., and Pedersen, J. O. (1999). ”Exploiting hierarchy in text categorization.” Information Retrieval, 1(3): 193-216.

Wilks, Y. (1997). Information Extraction as a Core Language Technology. SCIE: 1-9.

Wise, J. A. (1999). ”The ecological approach to text visualization.” Journal of the American Society for Information Science, 50(13): 1224-1233.

Wong P., Cowley W., Foote H., Jurrus E., Thomas J. (2000), ”Visualizing sequential patterns for text mining,” Proc. IEEE Information Visualization.

Wong P., Whitney P., Thomas J. (1999), Visualizing Association Rules for Text Mining. Proceedings of the 1999 IEEE Symposium on Information Visualization.

Xu, J. and W. B. Croft. (1996). Query expansion using local and global document analysis. In SIGIR ’96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 4-11, Zurich.

Yang, Y. (2001). A Study on Thresholding Strategies for Text Categorization. Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval. New Orleans, US, ACM Press, New York, US: 137-145.

Yang, Y., Ault, T., Pierce, T., and Lattimer, C. W. (2000). Improving text categorization methods for event tracking. Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval. Athens, GR, ACM Press, New York, US: 65-72.

Yang, Y. and C. G. Chute (1994). ”An example-based mapping method for text categorization and retrieval.” ACM Transactions on Information Systems, 2(3): 252-277.

Yang, Y. and X. Liu (1999). A re-examination of text categorization methods. Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval. Berkeley, US, ACM Press, New York, US: 42-49.

Yavuz, T. and H. A. Guvenir (1998). Application of k-nearest neighbor on feature projections classifier to text categorization. Proceedings of ISCIS-98, 13th International Symposium on Computer and Information Sciences, Ankara, TR, IOS Press, Amsterdam, NL: 135-142.

Yeh, A., Hirschman, L., and Morgan, A. (2002). ”Background and Overview for KDD Cup 2002 Task 1: Information Extraction from Biomedical Articles.” KDD Explorarions, 4(2): 87-89.

Zhang, J., R. Jin, Yang Y., and Hauptmann, A. (2003). Modified Logistic Regression: An Approximation to SVM and Its Applications in Large-Scale Text Categorization. Proceedings of ICML-03, 20th International Conference on Machine Learning. Washington, DC, Morgan Kaufmann Publishers, San Francisco, US.

Zhang, J. and Y. Yang (2003). Robustness of regularized linear classification methods in text categorization. Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval, Smeaton. Toronto, CA, ACM Press, New York, US: 190-197.

Zhang, T. and F. J. Oles (2001). ”Text Categorization Based on Regularized Linear Classification Methods.” Information Retrieval, 4(1): 5-31.