Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence(IJCAI-01), pages 979-984, Seattle, WA, August, 2001

Mining Soft-Matching Rules from Textual Data

Un Yong Nahm and **Raymond J. Mooney**
Department of Computer Sciences,
University of Texas, Austin, TX 78712-1188
{pebronia,mooney}@cs.utexas.edu

Abstract

Text mining concerns the discovery of knowledge from unstructured textual data. One important task is the discovery of rules that relate specific words and phrases. Although existing methods for this task learn traditional logical rules, soft-matching methods that utilize word-frequency information generally work better for textual data. This paper presents a rule induction system, TEXTRISE, that allows for partial matching of text-valued features by combining rule-based and instance-based learning. We present initial experiments applying TEXTRISE to corpora of book descriptions and patent documents retrieved from the web and compare its results to those of traditional rule and instance based methods.

1 Introduction

Text mining, discovering knowledge from unstructured natural-language text, is an important data mining problem attracting increasing attention [Hearst, 1999; Feldman, 1999; Mladenić, 2000]. Existing methods for mining rules from text use a hard, logical criteria for matching rules [Feldman and Hirsh, 1996; Ahonen-Myka *et al.*, 1999]. However, for most text processing problems, a form of soft matching that utilizes word-frequency information typically gives superior results [Salton, 1989; Cohen, 1998; Yang, 1999]. Therefore, the induction of soft-matching rules from text is an important, under-studied problem.

We present a method, TEXTRISE, for learning soft-matching rules from text using a modification of the RISE algorithm [Domingos, 1996], a hybrid of rule-based and instance-based (nearest-neighbor) learning methods. Such a hybrid is good match for text mining since rule-induction provides simple, interpretable rules, while nearest-neighbor provides soft matching based on a specified similarity metric. Currently in TEXTRISE, we use the vector-space model from information retrieval (IR) to provide an appropriate similarity metric [Salton, 1989].

We present results on applying TEXTRISE to two text databases, one of book information extracted from an online bookstore, and another of patent applications available on the web. We evaluate the quality of the discovered rules on independent data by measuring the similarity of predicted text and actual text. By comparing results to the predictions made by nearest-neighbor and mined association rules, we demonstrate the advantage of mining soft-matching rules.

2 Background

2.1 Mining Rules from Text

Several researchers have applied traditional rule induction methods to discover relationships from textual data. FACT [Feldman and Hirsh, 1996] discovers rules from text using a well-known technique for *association rule mining*. For example, it discovered rules such as “Iraq ⇒ Iran”, and “Kuwait and Bahrain ⇒ Saudi Arabia” from a corpus of Reuters news articles. Ahonen *et al.*(1998) also applied existing data mining techniques to discover *episode rules* from text. For example: “If “chemicals” and “processing” occurs within 2 consequent words, the word “storage” co-occurs within 3 words.” is an episode rule discovered from a collection of Finnish legal documents.

In addition, decision tree methods such as C4.5 and C5.0, and rule learners such as FOIL, and RIPPER have been used to discover patterns from textual data [Nahm and Mooney, 2000b; Ghani *et al.*, 2000]. All of these existing methods discover rules requiring an exact match.

2.2 Mining Information Extracted from Text

Nahm and Mooney(2000a; 2000b) introduced an alternative framework for text mining based on the integration of *information extraction* (IE) and traditional data mining. IE is a form of shallow text understanding that locates specific pieces of data in natural-language text. Traditional data mining assumes that information is in the form of a relational database; unfortunately, in many applications, information is only available in the form of unstructured documents. IE addresses this problem by transforming a corpus of textual documents into a structured database. An IE module can extract data from raw text, and the resulting database can be processed by a traditional data mining component.

In this work, extracted textual data was mined using traditional rule induction systems such as C4.5rules [Quinlan, 1993] and RIPPER [Cohen, 1995]. Rules were induced for predicting the text in each slot using the extracted information in all other slots. However, the heterogeneity of textual databases causes a problem: the same or similar objects are often referred to using different (but similar) textual strings. This issue becomes clear when we consider the Web, a vast and dynamic warehouse of text documents, as a potential target for text mining. Since the Web has no centralized moderator, it is highly heterogeneous, making it difficult to apply strict matching to text extracted from web documents [Cohen, 1998].

2.3 Information Retrieval Vector-Space Model

The vector-space model is typically used in IR to determine the similarity of two documents. In this model, a text is represented as a vector of real numbers, where each component corresponds to a word that appears in the set of all documents and the value is its frequency in the document. This is also known as a *bag of-words* (BOW) representation. The similarity of two documents x and y is the *cosine of the angle* between two vectors $\vec{x}$ and $\vec{y}$ representing x and y respectively, and calculated by the following formula:

$$Similarity(x, y) = \frac{\vec{x} \cdot \vec{y}}{\mid \vec{x} \mid \times \mid \vec{y} \mid} \quad (1)$$

where $\mid \vec{x} \mid$ and $\mid \vec{y} \mid$ are the norms of each document vector.

The TFIDF (Term Frequency, Inverse Document Frequency) weighting scheme [Salton, 1989] is used to assign higher weights to distinguished terms in a document. TFIDF makes two assumptions about the importance of a term. First, the more a term appears in the document, the more important it is (*term frequency*). Second, the more it appears through the entire collection of documents, the less important it is since it does not characterize the particular document well (*inverse document frequency*). In the TFIDF framework, the weight for term t_j in a document d_i, w_{ij} is defined as follows:

$$w_{ij} = tf_{ij} \times \log_2 \frac{N}{n} \quad (2)$$

where tf_{ij} is the frequency of term t_j in document d_i, N the total number of documents in collection, and n the number of documents where term t_j occurs at least once.

2.4 RISE: Learning Soft-Matching Rules

The RISE induction algorithm unifies rule-based and instance-based learning [Domingos, 1996]. Instead of requiring rules to match exactly, RISE makes predictions by selecting the closest matching rule according to a standard distance metric used by nearest-neighbor methods (a modified Euclidian distance). By generating generalized rules instead of remembering specific instances, and by using a similarity metric rather than exact matching to make predictions, it elegantly combines the properties of rule induction and instance-based learning.

Soft-matching rules are acquired using a specific-to-general (bottom-up) induction algorithm that starts with maximally specific rules for every example, and then repeatedly minimally generalizes each rule to cover the nearest example it does not already cover, unless this results in a decrease in the accuracy of the overall rule base on the training data. This process repeats until any additional generalization decreases accuracy. When classifying examples, the nearest rule is used to predict the class. A leave-one-out method is used to determine the performance of the rule base on the training data, since an example is always correctly classified by its corresponding initial maximally-specific rule. In extensive experiments, RISE was fairly consistently more accurate than alternative methods, including standard rule-based and instance-based algorithms. Training is also reasonably efficient computationally, requiring time $O(e^2a^2)$ where e is the number of examples, and a the number of attributes.

Book Description
Title : Harry Potter and the Goblet of Fire (Book 4)
Author : Joanna K. Rowling
Comments: This book was the best book I have ever read. If you are in for excitement this book is the one you want to read.
Subject: Fiction, Mystery, Magic, Children, School, Juvenile Fiction, Fantasy, Wizards

Representation
Author = {“joanna”, “rowling”}
Title = {“harry”, “potter”, “goblet”, “fire”, “book”}
Comments = {“book”, “book”, “read”, “excitement”, “read”}
Subject = {“fiction”, “mystery”, “magic”, “children”, “school”, “juvenile”, “fiction”, “fantasy”, “wizards”}

Figure 1: An example of representation for a book document

3 The TEXTRISE Algorithm

RISE is not directly applicable to mining rules from extracted text because: 1) its similarity metric is not text-based and 2) it learns rules for classification rather than text prediction. TEXTRISE addresses both of these issues. We represent an IE-processed document as a list of bags of words (BOWs), one bag for each slot filler. We currently eliminate 524 commonly-occurring stop-words but do not perform stemming. Figure 1 shows an example for a book description and its BOW representation. Standard set-operations are extended to bags in the obvious way [Peterson, 1976]. A learned rule is represented as an antecedent that is a conjunction of BOWs for some subset of slots and a conclusion that is a predicted BOW for another slot (see Figure 4 for examples).

The standard TFIDF-weighted cosine metric is used to compute the similarity of two BOWs. The similarity of two examples (i.e. extracted documents) or rules is the average similarity of the BOWs in their corresponding slots. Bag intersection is used to compute the minimal generalization of two BOWs. The minimal generalization of two examples or rules is the minimal generalization of the BOWs in each of their corresponding slots. A rule is said to *cover* an example document if all of its antecedent BOWs are sub-bags of the example's corresponding BOWs. To extend the algorithm from classification to text prediction, we define a new measure for the accuracy of a rule set on an example set: $TextAccuracy(RS, ES)$ is the average cosine similarity of the predicted fillers for the examples in ES to the corresponding fillers predicted by a rule set RS. The algorithms for gen-

Inputs: $R = (A_1, A_2, ..., A_n, C_R)$ is a rule
$E = (E_1, E_2, ..., E_n, C_E)$ is an example.
A_i, E_i, C_R, and C_E are bags-of-words, possibly empty.
Output: R' is the generalized rule.
Function Most_Specific_Generalization (R, E)
For i := 1 to n do
$A_i{}' := A_i \cap E_i$
$R' := (A_1{}', A_2{}', ..., A_n{}', C_R \cap C_E)$
Return R'.

Figure 2: Generalization of a rule to cover an example

Input: ES is the training set.
Output: RS is the rule set.
Function TextRISE (ES)
$RS := ES$.
Compute $TextAccuracy(RS, ES)$.
Repeat
For each rule $R \in RS$,
$\hat{E} := \underset{E \in ES'}{arg\,max}\ Similarity(E, R)$
where $ES' = \{E' : E' \in ES$ and
E' is not covered by $R\}$
R' := Most_Specific_Generalization($R, \hat{E}$)
RS' := RS with R replaced by R'
If $TextAccuracy(RS', ES) \geq TextAccuracy(RS, ES)$
Then $RS := RS'$
If R' is identical to another rule in RS,
Then delete R' from RS.
Until no increase in $TextAccuracy(RS, ES)$ is obtained.
Return RS.

Figure 3: The TEXTRISE rule-learning algorithm

eralizing a rule to cover an example and for learning rules are described in Figure 2 and Figure 3 respectively. The algorithm is a straightforward modification of RISE using the new similarity and predictive-accuracy metrics, and is used to induce soft-matching rules for predicting the filler of each slot given the values of all other slots. Our implementation makes use of the the BOW library [McCallum, 1996] for the bag-of-words text processing.

3.1 Interestingness Measures

The output of TEXTRISE is an unordered set of soft matching rules. Ranking rules based on an interestingness metric can help a human user focus attention on the most promising relationships. Several metrics for evaluating the "interestingness" or "goodness" of mined rules, such as *confidence* and *support*, have been proposed [Bayardo Jr. and Agrawal, 1999]. However, the traditional definitions for these metrics assume exact matches for conditions. Consequently, we modify these two common metrics for judging the goodness of soft-matching rules.

A *rule* consists of an antecedent and a consequent, and is denoted as $A \rightarrow C$ where A is equal to $A_1 \wedge A_2 \wedge ... \wedge A_i$. The *similarity-support* of an antecedent A, denoted as $simsup(A)$, is the number of examples in the data set that are soft-matched by A. In other words, $simsup(A)$ is the number of examples for which A is the closest rule in the rule base. The similarity-support of rule $A \rightarrow C$, denoted as $simsup(A \rightarrow C)$, is defined as the sum of similarities between C and the consequents of the examples soft-matched by A in the data set. In these definitions, we replace the traditional hard-matching constraints for a rule with weaker constraints determined relative to all the other rules in the rule base. Similarity-confidence of a rule $A \rightarrow C$, denoted by $simconf(A \rightarrow C)$, is computed as below.

$$simconf(A \rightarrow C) = \frac{simsup(A \rightarrow C)}{simsup(A)}$$

4 Evaluation

4.1 Data Sets

Two domains are employed in our evaluation of TEXTRISE: book data from `Amazon.com` and patent data downloaded from `Getthepatent.com`. We manually developed simple pattern-based IE systems or "wrappers" to automatically extract various labeled text-fields from the original HTML documents. The text extracted for each slot is then processed into a bag-of-words after removal of stop-words and remaining HTML commands.

The book data set is composed of 6 subsets, science fiction, literary fiction, mystery, romance, science, and children's books. 1,500 titles were randomly selected for each genre to make the total size of the book data set to be 9,000. A wrapper extracts 6 slots: `titles`, `authors`, `subject` terms, `synopses`, published `reviews`, and customer `comments`. Sample rules from this domain are given in Figure 4. Numbers associated to each word denotes the number of occurrences in the bag. The similarity-confidence and similarity-support values for each rule are also given.

3,000 patent documents were collected from dynamically generated web pages returned by a keyword search for "artificial intelligence". Four slots of `titles`, `abstracts`, `claims`, and `descriptions` are extracted for each patent. Sample rules are given in Figure 5.

4.2 Results

Unlike a standard rule learner that predicts the presence or absence of a specific slot value, TEXTRISE predicts a bag-of-words for each slot. Therefore, we evaluate the performance of TEXTRISE by measuring the average cosine similarity of the predicted slot values to the actual fillers for each slot. We compare the system to a standard nearest-neighbor method to show that TEXTRISE's compressed rule base is superior at predicting slot-values. In both methods, prediction is made by selecting the closest rule/example using only the text in the antecedent slots. We also tested nearest-neighbor without using information extraction to show the benefit of IE-based text mining. To clearly show IE's role, the only change made to nearest-neighbor was to treat the set of BOW's for the antecedent slots as a single, larger BOW.

The experiments were performed on the 9,000 book descriptions using ten-fold cross validation. Learning curves for predicting the `title` slot are shown in Figure 6. The graph shows 95% confidence intervals for each point. All the results on average similarities, precisions, and F-measures were statistically evaluated by a one-tailed, paired *t*-test. For

Rules from 2,500 Book Descriptions

title nancy(1), drew(1)
synopses nancy(1)
subject children(2), fiction(2), mystery(3), detective(3), juvenile(1), espionage(1)
$\rightarrow$
author keene(1), carolyn(1)
$[49.71\%, 1.99]$

synopses role(1), protein(1), absorption(1), metabolism(4), vitamins(1), minerals(1)
reviews health(1)
subject science(1), human(1), physiology(1)
$\rightarrow$
title nutrition(1)
$[27.87\%, 0.56]$

author beatrice(1), gormley(1)
synopses witness(1), ufo(1), landing(1)
subject science(1), fiction(2)
$\rightarrow$
reviews aliens(1), ufo(1), book(2)
$[13.79\%, 0.34]$

title charlotte(1), perkins(1), gilman(1)
synopses work(1), utopias(1), herland(1), ourland(1)
reviews gilman(1), author(1)
subject literature(2), criticism(2), classics(1), women(1), literary(1)
$\rightarrow$
comments utopia(1), feminist(1)
$[36.46\%, 0.73]$

title dance(1)
$\rightarrow$
subject romance(2), fiction(2)
$[31.68\%, 0.98]$

Figure 4: Sample Rules from Book Descriptions

each training set size, two pairs of systems(TextRISE versus nearest-neighbor and nearest-neighbor versus nearest-neighbor without information extraction) were compared to determine if their differences were statistically significant ($p < 0.05$).

The results indicate that TextRISE does best, while nearest-neighbor without IE does worst. This shows TextRISE successfully summarizes the input data in the form of prediction rules. The rule-compression rate of TextRISE is about 80%, which means the number of rules TextRISE produces is 80% of the number of examples originally stored in the initial rule base. We conducted the same experiments for other slots, and found similar results except for predicting the `author` slot. In predicting the `author` slot, neither information extraction nor TextRISE improves performance over simple nearest-neighbor.

In addition to the textual similarity, we developed analogs for precision and recall. Precision and recall were defined as follows, where C is the correct BOW and P is the predicted one.

$$Precision = Similarity(C \cap P, P) \tag{3}$$

$$Recall = Similarity(C \cap P, C) \tag{4}$$

Rules from 3,000 AI-Related Patent Documents

abstract device(2), images(1)
claims invention(4), system(5)
description information(5), data(2), control(2), stored(2), point(1), user(3), application(2), flow(1), object(1), operation(1), software(1), storage(1)
$\rightarrow$
title video(1)
$[9.44\%, 0.54]$

title automated(1)
claims device(4), based(1), determining(3), comprising(4), input(3), plurality(2), comprises(5), claim(7)
$\rightarrow$
abstract apparatus(1)
$[7.25\%, 0.42]$

Figure 5: Sample Rules from Patent Documents

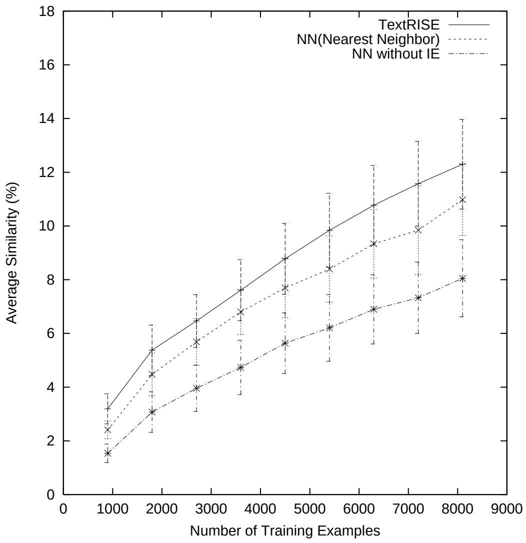

Figure 6: Average similarities for book data (`title`)

F-measure is defined as the harmonic mean for precision and recall as follows:

$$F\!-\!measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

Learning curves for precision and F-measure are presented in Figure 7 and Figure 8. TextRISE provides higher precision, since the conclusions of many of its rules are smaller generalized BOWs, and overall F-measure is moderately increased.

To compare TextRISE with traditional rule mining methods, we generated association rules using the Apriori algorithm [Agrawal and Srikant, 1994] and a publicly available implementation [Borgelt, 2000]. We treated each word in each slot as a separate item and generated associations between them. Among all the generated rules, those with words for the slot to be predicted are selected. For each test example, a prediction is made by building a BOW using the conclusions of all matching association rules. With the default

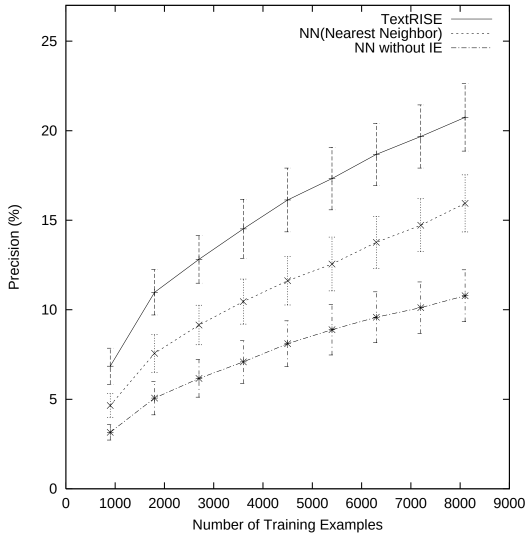

Figure 7: Precisions for book data (`title`)

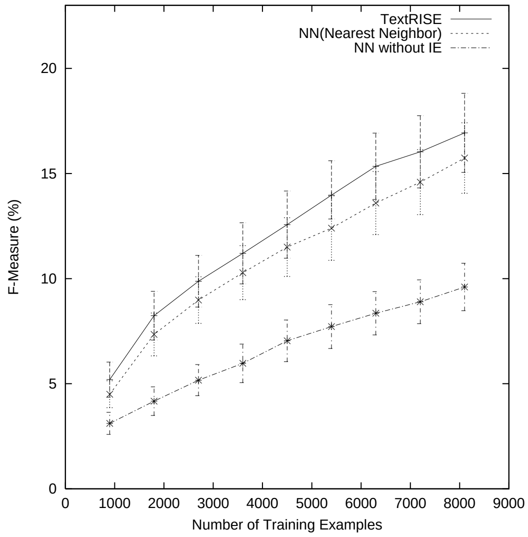

Figure 8: F-measures for book data (`title`)

parameter setting (minimum support of 10% and minimum confidence of 80%), the average similarity of predictions is almost 0%. We lowered the minimum support and the confidence until memory problems occurred on a SUN Ultra Sparc 1 (120MB). With the lowest minimum setting for support (3%) and confidence (30%), the average similarity remains very low: 0.0005% for 900 training examples and 0.0014% for 8,100 training examples. These results strongly suggest the usefulness of soft-matching rules in prediction tasks for textual data.

5 Related Research

Several previous systems mine rules from text [Feldman and Hirsh, 1996; Ahonen *et al.*, 1998]; however, they discover hard-matching rules and do not use automated information extraction. Ghani *et al.*(2000) applied several rule induction methods to a database of corporations automatically extracted from the Web. Interesting rules such as "Advertising agencies tend to be located in New York" were discovered; however, such learned rules must exactly match extracted text. WHIRL is a query processing system that combines traditional database and IR methods by introducing a "soft join" operation [Cohen, 1998]. WHIRL and TEXTRISE share a focus on soft-matching rules for text processing; however, rules in WHIRL must be written by the user while TEXTRISE tries to discover such rules automatically.

6 Future Work

A potential extension of the system is to generalize to a k-nearest-neighbor method that uses the k closest rules rather than just the single nearest rule. The predictions of these k rules could be combined by taking the average of the BOW vectors in their consequents. Likewise during learning, rules could be generalized to the k nearest uncovered examples using a similar averaging technique, possibly rounding values to maintain integer counts and simplify the resulting rules. Another potentially useful change to the generalization algorithm would be to use a semantic hierarchy such as WordNet [Fellbaum, 1998]. For example, the terms "thermodynamics" and "optics" could be generalized to "physics." Finally, for short extracted strings, string edit distance [Wagner and Fisher, 1974] might be a more useful measure of textual similarity than the cosine measure.

Better metrics for evaluating the interestingness of text-mined rules is clearly needed. One idea is to use a semantic network like WordNet to measure the semantic distance between the words in the antecedent and the consequent of a rule, preferring more "surprising" rules where this distance is larger. For example, this would allow ranking the rule "beer $\rightarrow$ diapers" above "beer $\rightarrow$ pretzels" since beer and pretzels are both food products and therefore closer in WordNet.

Although our preliminary results are encouraging, we are planning to evaluate the approach on other corpora such as a larger database of patents, grant abstracts from the National Science Foundation, or research papers gathered by CORA (`www.cora.whizbang.com`) or ResearchIndex (`citeseer.nj.nec.com`).

7 Conclusions

The problem of discovering knowledge in textual data is an exiting new area in data mining. Existing text-mining systems discover rules that require exactly matching substrings; however, due to variability and diversity in natural-language data, some form of soft matching based on textual similarity is needed. We have presented a system TEXTRISE that uses a hybrid of rule-based and instance-based learning methods to discover soft-matching rules from textual databases automatically constructed from document corpora via information extraction. With encouraging results of preliminary experiments, we showed how this approach can induce accurate predictive rules despite the heterogeneity of automatically extracted textual databases.

Acknowledgements

This research was supported by the National Science Foundation under grant IRI-9704943.

References

[Agrawal and Srikant, 1994] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB-94)*, pages 487–499, Santiago, Chile, September 1994.

[Ahonen *et al.*, 1998] Helena Ahonen, Oskari Heinonen, Mika Klemettinen, and A. Inkeri Verkamo. Applying data mining techniques for descriptive phrase extraction in digital document collections. In *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries*, pages 2–11, Santa Barbara, CA, April 1998.

[Ahonen-Myka *et al.*, 1999] Helena Ahonen-Myka, Oskari Heinonen, Mika Klemettinen, and A. Inkeri Verkamo. Finding co-occurring text phrases by combining sequence and frequent set discovery. In Ronen Feldman, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99) Workshop on Text Mining: Foundations, Techniques and Applications*, pages 1–9, Stockholm, Sweden, August 1999.

[Bayardo Jr. and Agrawal, 1999] Roberto J. Bayardo Jr. and Rakesh Agrawal. Mining the most interesting rules. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99)*, pages 145–154, San Diego, CA, August 1999.

[Borgelt, 2000] Christian Borgelt. Apriori version 2.6. http://fuzzy.cs.Uni-Magdeburg.de/~borgelt/, 2000.

[Cohen, 1995] William W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML-95)*, pages 115–123, San Francisco, CA, 1995.

[Cohen, 1998] William W. Cohen. Providing database-like access to the web using queries based on textual similarity. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data (SIGMOD-98)*, pages 558–560, Seattle, WA, June 1998.

[Domingos, 1996] Pedro Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24:141–168, 1996.

[Feldman and Hirsh, 1996] Ronen Feldman and Haym Hirsh. Mining associations in text in the presence of background knowledge. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 343–346, Portland, OR, August 1996.

[Feldman, 1999] Ronen Feldman, editor. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99) Workshop on Text Mining: Foundations, Techniques and Applications*, Stockholm, Sweden, August 1999.

[Fellbaum, 1998] Christiane D. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[Ghani *et al.*, 2000] Rayid Ghani, Rosie Jones, Dunja Mladenić, Kamal Nigam, and Sean Slattery. Data mining on symbolic knowledge extracted from the web. In Dunja Mladenić, editor, *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining*, pages 29–36, Boston, MA, August 2000.

[Hearst, 1999] Marti Hearst. Untangling text data mining. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pages 3–10, College Park, MD, June 1999.

[McCallum, 1996] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow, 1996.

[Mladenić, 2000] Dunja Mladenić, editor. *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining*, Boston, MA, August 2000.

[Nahm and Mooney, 2000a] Un Yong Nahm and Raymond J. Mooney. A mutually beneficial integration of data mining and information extraction. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 627–632, Austin, TX, July 2000.

[Nahm and Mooney, 2000b] Un Yong Nahm and Raymond J. Mooney. Using information extraction to aid the discovery of prediction rules from texts. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining*, pages 51–58, Boston, MA, August 2000.

[Peterson, 1976] James L. Peterson. Computation sequence sets. *Journal of Computer and System Sciences*, 13(1):1–24, August 1976.

[Quinlan, 1993] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo,CA, 1993.

[Salton, 1989] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[Wagner and Fisher, 1974] Robert A. Wagner and Michael J. Fisher. The string to string correction problem. *Journal of the Association for Computing Machinery*, 21:168–173, 1974.

[Yang, 1999] Yiming Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, May 1999.