# Selecting Attributes for Sentiment Classification Using Feature Relation Networks

Ahmed Abbasi, *Member*, *IEEE*, Stephen France, *Member*, *IEEE*,
Zhu Zhang, and Hsinchun Chen, *Fellow*, *IEEE*

**Abstract**—A major concern when incorporating large sets of diverse n-gram features for sentiment classification is the presence of noisy, irrelevant, and redundant attributes. These concerns can often make it difficult to harness the augmented discriminatory potential of extended feature sets. We propose a rule-based multivariate text feature selection method called Feature Relation Network (FRN) that considers semantic information and also leverages the syntactic relationships between n-gram features. FRN is intended to efficiently enable the inclusion of extended sets of heterogeneous n-gram features for enhanced sentiment classification. Experiments were conducted on three online review testbeds in comparison with methods used in prior sentiment classification research. FRN outperformed the comparison univariate, multivariate, and hybrid feature selection methods; it was able to select attributes resulting in significantly better classification accuracy irrespective of the feature subset sizes. Furthermore, by incorporating syntactic information about n-gram relations, FRN is able to select features in a more computationally efficient manner than many multivariate and hybrid techniques.

**Index Terms**—Natural language processing, machine learning, text mining, subspace selection, affective computing.

## 1 Introduction

The Internet is rich in directional text (i.e., text containing opinions and emotions). The web provides volumes of text-based data about consumer preferences, stored in online review websites, web forums, blogs, etc. Sentiment analysis has emerged as a method for mining opinions from such text archives. It uses machine learning methods combined with linguistic attributes/features in order to identify among other things the sentiment polarity (e.g., positive, negative, and neutral) and intensity (e.g., low, medium, and high) for a particular text. Important applications of text sentiment analysis include evaluating consumer perceptions [25], [26], [36], shedding light on investor opinions [8], and assessing the quality of online reviews [42].

Although it serves numerous functions, text sentiment analysis remains a challenging problem. It requires the use of large quantities of linguistic features [2], [4]. Various types of n-gram features have emerged for capturing sentiment cues in text. However, few studies have attempted to integrate these heterogeneous n-gram categories into a single feature set due to the inherent challenges. Noise and redundancy in the feature space increase the likelihood of overfitting. They also prevent many quality features from being incorporated due to computational limitations, resulting in diminished accuracy [17]. Further, large text feature spaces span hundreds of thousands of features, making many powerful feature selection methods infeasible. Consequently existing feature selection methods do not adequately address attribute relevance and redundancy issues, which are critical for text sentiment analysis [40].

In this study, we propose the use of a rich set of n-gram features spanning many fixed and variable n-gram categories. We couple the extended feature set with a feature selection method capable of efficiently identifying an enhanced subset of n-grams for opinion classification. The proposed Feature Relation Network is a rule-based multivariate n-gram feature selection technique that efficiently removes redundant or less useful n-grams, allowing for more effective n-gram feature sets. FRN also incorporates semantic information derived from existing lexical resources, enabling augmented weighting/ranking of n-gram features. Experimental results reveal that the extended feature set and proposed feature selection method can improve opinion classification performance over existing selection methods.

The remainder of the paper is organized as follows: Section 2 provides a review of related work on features and feature selection methods for sentiment analysis. It also identifies research gaps. Section 3 provides our research design. Section 4 includes an experimental evaluation of the proposed features and selection method in comparison with existing feature sets and feature selection techniques. Finally, Section 5 outlines conclusions and future directions.

- *A. Abbasi and S. France are with the Sheldon B. Lubar School of Business, University of Wisconsin—Milwaukee, Lubar Hall, PO Box 742, 3202 N. Maryland Ave., Milwaukee, WI 53201-0742. E-mail: {abbasi, france}@uwm.edu.*
- *Z. Zhang is with the Department of Management Information Systems, Eller College of Management, University of Arizona, 430BB McClelland Hall, 1130 E. Helen St., PO Box 210108, Tucson, AZ 85721-0108. E-mail: zhuzhang@eller.arizona.edu.*
- *H. Chen is with the Artificial Intelligence Lab, Department of Management Information Systems, Eller College of Management, University of Arizona, 430Z McClelland Hall, 1130 E. Helen St., PO Box 210108, Tucson, AZ 85721-0108. E-mail: hchen@eller.arizona.edu.*

*Manuscript received 23 Apr. 2009; revised 11 Oct. 2009; accepted 24 Jan. 2010; published online 6 July 2010.*
*Recommended for acceptance by D. Tao.*
*For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2009-04-0370. Digital Object Identifier no. 10.1109/TKDE.2010.110.*

## 2 Related Work

Opinion mining involves several important tasks, including sentiment polarity and intensity assignment [18], [31]. Polarity assignment is concerned with determining whether a text has a positive, negative, or neutral semantic orientation. Sentiment intensity assignment looks at whether the positive/negative sentiments are mild or strong. Given the two phrases "I don't like you" and "I hate you," both would be assigned a negative semantic orientation but the latter would be considered more intense.

Effectively classifying sentiment polarities and intensities entails the use of classification methods applied to linguistic features. While several classification methods have been employed for opinion mining, Support Vector Machine (SVM) has outperformed various techniques including Naïve Bayes, Decision Trees, Winnow, etc. [1], [2], [7], [29]. The most popular class of features used for opinion mining is n-grams [28], [38]. Various n-gram categories have attained state-of-the-art results [3], [27]. Larger n-gram feature sets require the use of feature selection methods to extract appropriate attribute subsets. Next, we discuss these two areas: n-gram features and feature selection techniques used for sentiment analysis.

### 2.1 N-Gram Features for Sentiment Analysis

N-gram features can be classified into two categories: fixed and variable. Fixed n-grams are exact sequences occurring at either the character or token level. Variable n-grams are extraction patterns capable of representing more sophisticated linguistic phenomena. A plethora of fixed and variable n-grams have been used for opinion mining, including word, part-of-speech (POS), character, legomena, syntactic, and semantic n-grams.

Word n-grams include bag-of-words (BOWs) and higher order word n-grams (e.g., bigrams, trigrams). Word n-grams have been used effectively in several studies [28]. Typically, unigrams to trigrams are used [3], [27], though 4-grams have also been employed [34]. Word n-grams often provide a feature set foundation, with additional feature categories added to them [4], [27], [34], [38].

Given the pervasiveness of adjectives and adverbs in opinion-rich text, POS tag, n-grams are very useful for sentiment classification [10], [12]. Additionally, some studies have employed word plus part-of-speech (POSWord) n-grams. These n-grams consider a word along with its POS tag in order to overcome word-sense disambiguation in situations where a word may otherwise have several senses [38]. For example, the phrase "quality of the" can be represented with the POSWord trigram "quality-noun of-prep the-det."

Character n-grams are letter sequences. For example, the word "like" can be represented with the following two and three letter sequences "li, ik, ke, lik, ike." While character n-grams were previously used mostly for style classification, they have recently been shown to be useful in related affect classification research attempting to identify emotions in text [2].

Legomena n-grams are collocations that replace once (hapax legomena) and twice occurring words (dis legomena) with "HAPAX" and "DIS" tags [2], [38]. Hence, the trigram "I hate Jim" would be replaced with "I hate HAPAX" provided "Jim" only occurs once in the corpus. The intuition behind such collocations is to remove sparsely occurring words with tags that will allow the extracted n-grams to be more generalizable [37], [38].

TABLE 1
N-Gram Features Used for Sentiment Analysis

| N-Gram Category | Examples | Prior Studies |
|---|---|---|
| Character | q, u, qu, ua, al, li, qua, ual, ali | [2] |
| Word | quality, quality of, quality of the | [1, 8, 25, 28, 38, 27] |
| POS Tag | noun, noun prep, noun prep det | [12, 28] |
| Word/POS Tag | quality-*noun* of-*prep* the-*det* | [38] |
| Legomena | the UNIQUE, of the UNIQUE<br>different-*adj* U-*noun* | [2]<br>[37, 38] |
| Syntactic Phrase Patterns | <subj> passive-verb<br>DECL::NP VERB NP<br><subj> ActInfVP | [33]<br>[12]<br>[34] |
| Semantic Phrase Patterns | SYN125 of the<br>strong-tyranny, weak-aberration<br>n+aj, n+dj, av+n<br>POSITIVE of the<br>APP/Appreciation:ORI/Negative | [6]<br>[33]<br>[10]<br>[27]<br>[4] |

Syntactic phrase patterns are learned variable n-grams [34]. Riloff et al. [33] developed a set of syntactic templates and information extraction patterns (i.e., instantiations of those templates) reflective of subjective content. Given a set of predefined templates, patterns with the greatest occurrence difference across sentiment classes are extracted. For example, the template "<subj> passive-verb" may produce the pattern "<subj> was satisfied." Such phrase patterns can represent syntactic phenomena difficult to capture using fixed-word n-grams [12], [38].

Semantic phrase patterns typically use an initial set of terms or phrases, which are manually or automatically filtered and coded sentiment polarity/intensity information. Many studies have used WordNet to automatically generate semantic lexicons [19], [23] or semantic word classes [6]. Riloff et al. [33] used a semiautomated approach to construct sets of strong/weak subjectivity and objective nouns. Others have manually annotated or derived semantic phrases [4], [10].

Table 1 provides a summary of n-gram features used for opinion classification. Based on the table, we can see that many n-gram categories have been used in prior opinion mining research. However, few studies have employed large sets of heterogeneous n-grams. As stated before, most studies utilized word n-grams in combination with one other category, such as POS tag, legomena, semantic, or syntactic n-grams, e.g., [1], [4], [27], [34], [38].

### 2.2 Feature Selection for Sentiment Analysis

Prior sentiment classification studies have placed limited emphasis on feature selection techniques, despite their benefits [20]. Feature selection can potentially improve classification accuracy [17], narrow in on a key feature subset of sentiment discriminators, and provide greater insight into important class attributes. There are two categories of feature selection methods [15], [16], both of which have been used in prior sentiment analysis work: univariate and multivariate.

TABLE 2
Univariate Methods Used for Sentiment Classification

**Chi Squared** [35]

$$\chi^2(a,Y) = \sum_{a_{x_j}\in\{0,1\}} \sum_{i\in Y} \frac{\left(F(a_{x_j},Y=i) - E(a_{x_j},Y=i)\right)^2}{E(a_{x_j},Y=i)}$$

where: $\chi^2(a,Y)$ is the chi-squared value for feature $a$ across classes $Y$

$X = [x_1, x_2, ...x_m]$ are the training examples

$a_{x_j} = 1$ if the training instance $x_j$ contains feature $a$, $a_{x_j} = 0$ otherwise

$F(a_{x_j}, Y=i)$ is the observed frequency of $a_x$, when $Y = i$

$E(a_{x_j}, Y=i) = \frac{p(a)\,p(Y=i)}{m}$ is the expected value of $a_x$, when $Y = i$, across $X$

**Information Gain** [2, 3, 13]

$$IG(Y,a) = H(Y) - H(Y \mid a)$$

where: $IG(Y,a)$ is the information gain for feature $a$

$H(Y) = -\sum_{i\in Y} p(Y=i)\log_2 p(Y=i)$ is the entropy across classes $Y$

$H(Y \mid a) = -\sum_{j\in a} p(a=j) \sum_{i\in Y} p(Y=i \mid a=j)\log_2 p(Y=i \mid a=j)$

is the entropy of $Y \mid a$

**Log Likelihood Ratio** [12, 27, 39]

$$w(a) = \max_i \left( p(a \mid Y=i) \log \frac{p(a \mid Y=i)}{p(a \mid \neg Y=i)} \right)$$

where: $w(a)$ is the log likelihood for feature $a$ across classes $Y$

TABLE 3
Multivariate Methods Used for Sentiment Classification

**Decision Tree Models** [1]

Given training examples $X = [x_1, x_2, ...x_m]$ and class labels $y = [y_1, y_2, ...y_m]$
Initialize subset of surviving features $s = [1, 2, ...p]$ and ranked feature list $r = []$
Repeat until $s = []$ or some stopping criterion has been reached
  Train the classifier $\alpha = DT(X, y)$
  Extract the decision tree features from $\alpha$ into $d = [s(a_1), s(a_2), ...s(a_t)]$
  Update the ranked feature list $r = [r, d]$
  Update the surviving features $s = s(1: a_1 - 1, a_1 + 1: a_2 - 1, a_2 + 1: a_t - 1, a_t + 1: p)$
Output ranked feature list $r$

**Feature Subsumption Hierarchy** [34]

Initialize subset of surviving features $s = [1,2,...p]$
Repeat until all potential feature subsumptions have been evaluated
  If feature s($a$) representationally subsumes feature s($b$) based on the hierarchy
    If $IG(Y, s(a)) \geq IG(Y, s(a)) + \delta$
      Eliminate s($b$) from the feature set $s = s(1: b-1, b+1: p)$
where: $\delta$ is a parameter
$IG(Y, s(a))$ is the information gain for feature $s(a)$ across classes $Y$

**Genetic Algorithm** [3]

Given training examples $X = [x_1, x_2, ...x_m]$ and class labels $y = [y_1, y_2, ...y_m]$
Initialize solution population $s = [s_1, s_2, ...s_r]$, where $s_x = [s_{x1}, s_{x2}, ...s_{xp}]$, $s_{xn} \in \{0,1\}$
and the feature $s_{xn}$ is included in solution $s_x$ if $s_{xn} = 1$
Repeat until some stopping criterion has been reached
  Initialize population of new solutions $t = [t_1, t_2, ...t_r]$, where $t_x = []$
  Evaluate each solution's fitness $F_{s_x} = Fitness(s_x, X, y)$
  Select solutions based on fitness and add to $t$, where $p(s_x \in t) \propto F_{s_x} (\sum_{i=1}^{r} F_{s_i})^{-1}$
  For each of the $r/2$ solution pairs in $t$
    If random number $q \in [0,1] < thresh_c$, crossover $t_x$ and $t_{x+1}$ at point $k$
    $t_x = [t_x(t_{x1} : t_{xk}), t_{x+1}(t_{x+1k+1} : t_{x+1p})]$ and $t_{x+1} = [t_{x+1}(t_{x+11} : t_{x+1k}), t_x(t_{xk+1} : t_{xp})]$
  For each of the $r$ solutions in $t$
    For each $t_{x_n}$ in $t_x$, if $u \in [0,1] < thresh_m$, mutate $t_{x_n}$ where $t_{x_n} = 1 - t_{x_n}$
  Set $s$ equal to $t$ for the next iteration $s = t$

**Recursive Feature Elimination** [3]

Given training examples $X = [x_1, x_2, ...x_m]$ and class labels $y = [y_1, y_2, ...y_m]$
Initialize subset of surviving features $s = [1, 2, ...p]$ and ranked feature list $r = []$
Repeat until $s = []$
  Train the classifier $\alpha = SVM(X, y)$
  Compute the weight vector $w = \sum_k \alpha_k y_k x_k$
  Compute the ranking criteria $c_i = |w_i|$, for all $i$
  Find the feature with the smallest ranking criterion $a = \arg\min(c)$
  Update the ranked feature list $r = [s(a), r]$
  Eliminate the feature with the smallest ranking criterion $s = s(1: a-1, a+1: p)$
Output ranked feature list $r$

Univariate methods consider attributes individually. Examples include information gain, chi-squared, log likelihood, and occurrence frequency [11]. Although univariate methods are computationally efficient, evaluating individual attributes can also be disadvantageous since important attribute interactions are not considered. It is also easier to interpret the contribution of individual attributes using univariate methods. Most opinion mining studies have used univariate feature selection methods such as minimum frequency thresholds and the log-likelihood ratio [12], [27], [39]. Information gain (IG) [44], [45] has also been shown to work well for various text categorization tasks, including sentiment analysis [3]. Tsutsumi et al. [35] used the Chi-Squared test to select features for text sentiment classification. Table 2 shows select univariate feature selection methods used in sentiment classification studies.

Multivariate methods consider attribute groups or subsets. These techniques sometimes use a wrapper model for attribute selection, where the accuracy of a target classifier is used as an evaluation metric for the predictive power of a particular feature subset [16]. Examples include decision tree models, recursive feature elimination, and genetic algorithms. By performing group-level evaluation, multivariate methods consider attribute interactions. Consequently, these techniques are also computationally expensive in relation to univariate methods. Decision tree models (DTMs) use a wrapper, where a DTM is built on the training data and features incorporated by the tree are included in the feature set [1], [21]. Recursive feature elimination uses a wrapper model based on an SVM classifier [15]. During each iteration, the remaining features are ranked based on the absolute values of their SVM weights, and a certain number/percentage of these are retained [2], [3], [24]. Genetic algorithms (GAs) have been used to search for ideal subsets across the feature subspace in text classification problems such as style [20] and sentiment analysis [3]. A major pitfall associated with GA is that they can be computationally very expensive, since hundreds/thousands of solutions have to be evaluated using a classifier [3]. Feature subsumption hierarchies (FSHs) use the idea of performance-based feature subsumption to remove redundant or irrelevant higher order n-grams [34]. Only those word bigrams and trigrams are retained, which provide additional information over the unigrams they encompass. Table 3 shows multivariate methods used for sentiment classification.

### 2.2.1 Other Feature Selection Methods

In addition to prior sentiment feature selection methods, it is important to briefly discuss multivariate and hybrid methods used in related tasks. Principal component analysis (PCA) has been used considerably for dimensionality

reduction in text style classification problems [46]. Recently, many powerful dimensionality reduction techniques have also been applied to nontext feature selection problems. These include conditional mutual information (CMIM), harmonic mean, geometric mean, general averaged divergence analysis, and discriminative locality alignment (DLA) [14], [47], [48], [49], [50]. CMIM outperformed comparison techniques (including DTM) on image classification and biomedical prediction tasks [14]. DLA outperformed methods such as PCA and linear discriminant analysis on image classification tasks [50].

Hybrid methods that combine univariate measures with multivariate selection strategies can potentially improve the accuracy and convergence efficiency of otherwise slower multivariate methods [5], [22]. For instance, a hybrid GA utilizing the IG measure has been shown to converge faster than regular GA, when applied to feature sets spanning up to 26,000 features [3].

### 2.3 Research Gaps

Based on our review, we have identified appropriate gaps. Most studies have used limited sets of n-gram features, typically employing one or two categories [27], [28]. Larger n-gram feature sets introduce computational difficulties and potential performance degradation stemming from noisy feature sets. For instance, the popular 2,000 movie review testbed developed by Pang et al. [28] has over 49,000 bag-of-words [4]. Higher order n-gram feature spaces can be even larger, with hundreds of thousands of potential attributes. Feature selection methods are needed to help manage the large feature spaces created from the use of heterogeneous n-grams. As Riloff et al. [34] noted, using additional text features without appropriate selection mechanisms is analogous to "throwing the kitchen sink." However, large-scale feature selection requires addressing relevance and redundancy, something many existing methods fail to do [40].

Redundancy is a big problem since there are a finite number of attributes that can be incorporated and n-grams tend to be highly redundant by nature. In the case of univariate methods, redundant features occupy valuable spots that may otherwise be utilized by attributes providing additional information and discriminatory potential. Powerful multivariate methods are capable of alleviating redundancy; however, they are often unsuitable for computational reasons. These methods have typically been applied to smaller feature sets, e.g., [15], [20]. It is unclear whether hybrid feature selection methods have the potential to overcome issues stemming from redundancy. Moreover, most of the feature selection methods described are generic techniques that have been applied to a plethora of problems, since they assess attribute relevance solely based on the training data. Whenever possible, domain knowledge should be incorporated into the feature selection process [16]. Existing lexicons and knowledge bases pertaining to the semantic and syntactic properties of n-grams could be exploited for enhanced assessment of relevance and redundancy associated with text attributes.

TABLE 4
N-Gram Feature Set

| Label | Description | Examples | |
|---|---|---|---|
| N-Char | Character-level n-grams | 1-Char | I, L, O, V, E, C, H, O, C, O, L, A |
| | | 2-Char | LO, OV, VE, CH, HO, OC, CO, OL |
| | | 3-Char | LOV, OVE, CHO, HOC, OCO |
| N-Word | Word-level n-grams | 1-Word | I, LOVE, CHOCOLATE |
| | | 2-Word | I LOVE, LOVE CHOCOLATE |
| | | 3-Word | I LOVE CHOCOLATE |
| N-POS | Part-of-speech tag n-grams | 1-POS | I, ADMIRE_VBP, NN |
| | | 2-POS | ADMIRE_VBP NN |
| | | 3-POS | I ADMIRE_VBP NN |
| N-POSWord | Word and POS tag n-grams | 1-POSWord | LOVE ADMIRE_VBP |
| | | 2-POSWord | I I LOVE ADMIRE_VBP |
| | | 3-POSWord | I I LOVE ADMIRE_VBP CHOCOLATE NN |
| N-Legomena | Hapax legomena and Dis legomena n-grams | 2-Legomena | LOVE DIS |
| | | 3-Legomena | I LOVE DIS |
| N-Semantic | Semantic class n-grams | 1-Semantic | SYN-Pronoun, SYN-Affection |
| | | 2-Semantic | SYN-Pronoun SYN-Affection |
| | | 3-Semantic | SYN-Pronoun SYN-Affection SYN-Candy |
| IEP-A/E | Information extraction patterns | IEP-A | <possessive> NP, <subj> AuxVP AdjP, <subj> AuxVP Dobj, ActVP <dobj>, ActVP Prep <np> |
| | | IEP-B | <subj> PassVP, InfVP Prep <np>, InfVP <dobj> |
| | | IEP-C | <subj> ActVP |
| | | IEP-D | <subj> ActVP Dobj |
| | | IEP-E | <subj> ActInfVP, <subj> PassInfVP, ActInfVP <dobj> |

## 3 Research Design

We propose the use of a rich set of n-gram features, coupled with the Feature Relation Network (FRN) for enhanced sentiment intensity and polarity classification performance. The proposed FRN feature selection method will be compared against various univariate, multivariate, and hybrid selection techniques used in prior research, including log-likelihood ratio, information gain, chi-squared, recursive feature elimination, decision tree models, and genetic algorithms. The extended feature set and FRN method are discussed in the remainder of this section.

### 3.1 Extended N-Gram Feature Set

We incorporate a rich set of n-gram features, consisted of all the categories discussed in the literature review. The feature set is shown in Table 4. The syntactic n-grams were derived using the Sundance package [33], [34]. This tool extracts n-gram instantiations of predefined pattern templates. Sundance learns n-grams that have the greatest occurrence difference across user-defined classes. For instance, the n-gram "endorsed <dobj>" is generated from the pattern template "ActVP <dobj>." The semantic n-grams were derived using WordNet, following an approach similar to that used by Kim and Hovy [19] and Mishne [23]. Words are clustered into semantic categories based on the number of common items in their synsets. New words are added to the cluster with the highest percentage of synonyms in common provided the percentage is above a certain threshold. Otherwise, the word is added to a new cluster.

### 3.2 Feature Relation Network

For text n-grams, the relationship between n-gram categories can facilitate enhanced feature selection by considering relevance and redundancy, two factors critical to large-scale

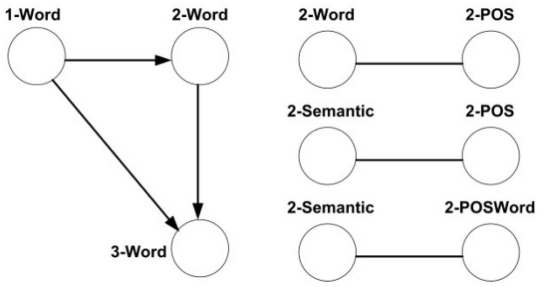



Fig. 1. (left) Subsumption relations between word n-grams and (right) parallel relations between various bigrams.

feature selection [41]. We propose a rule-based multivariate text feature selection method that considers semantic information and also leverages the syntactic relationships between n-gram features in order to efficiently remove redundant and irrelevant ones. Comparing all features within a feature set directly with one another can be an arduous endeavor. However, if the relationship between features can be utilized, thereby comparing only some logical subset of attributes, then the feature selection process can be made more efficient. Given large quantities of heterogeneous n-gram features, the FRN utilizes two important n-gram relations: subsumption and parallel relations. These two relations enable intelligent comparison between features in a manner that facilitates enhanced removal of redundant and/or irrelevant n-grams.

### 3.2.1 Subsumption Relations

In addition to prior sentiment feature selection methods, it is important to briefly discuss multivariate. The notion of subsumption was originally proposed by Riloff et al. [34]. A subsumption relation occurs between two n-gram feature categories, where one category is a more general, lower order form of the other [34]. A subsumes $\mathrm{B}(\mathrm{A} \rightarrow \mathrm{B})$ if B is a higher order n-gram category whose n-grams contain the lower order n-grams found in A. For example, word unigrams subsume word bigrams and trigrams, while word bigrams subsume word trigrams (as shown on the left side of Fig. 1). Given the sentence "I love chocolate," there are six word n-grams: I, LOVE, CHOCOLATE, I LOVE, LOVE CHOCOLATE, and I LOVE CHOCOLATE. The unigram LOVE is obviously important, generally conveying positive sentiment. However, what about the bigrams and trigrams? It depends on their weight, as defined by some heuristic (e.g., log likelihood or information gain). We only wish to keep higher order n-grams if they are adding additional information greater than that conveyed by the unigram LOVE. Hence, given $\mathrm{A} \rightarrow \mathrm{B}$, we keep features from category B if their weight exceeds that of their general lower order counterparts found in A by some threshold t [34]. For instance, the bigrams I LOVE and LOVE CHOCOLATE would only be retained if their weight exceeded that of the unigram LOVE by $t$ (i.e., if they provided additional information over the more general unigram). Similarly, the trigram I LOVE CHOCOLATE would only be retained if its weight exceeded that of the unigram LOVE and any remaining bigrams (e.g., I LOVE and LOVE CHOCOLATE) by $t$.

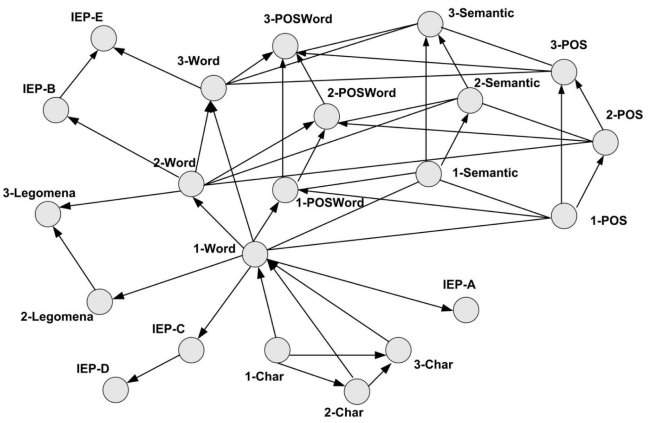



Fig. 2. The feature relation network.

### 3.2.2 Parallel Relations

A parallel relation occurs where two heterogeneous same order n-gram feature groups may have some features with similar occurrences. For example, word unigrams (1-Word) can be associated with many POS tags (1-POS), and vice versa. However, certain word and POS tags' occurrences may be highly correlated. Similarly, some POS tags and semantic class unigrams may be correlated if they are used to represent the same words. For example, the POS tag ADMIRE_VP and the semantic class SYN-Affection both represent words such as "like" and "love." Given two n-gram feature groups with potentially correlated attributes, A is considered to be parallel to B (A—B). If two features from these categories A and B, respectively, have a correlation coefficient greater than some threshold p, one of the attributes is removed to avoid redundancy. The right side of Fig. 1 shows some examples of bigram categories with parallel relations.

Correlation is a commonly used method for feature selection [11], [17]. However, correlation is generally used as a univariate method by comparing the occurrences of an attribute with the class labels, across instances [11]. Comparing attribute intercorrelation could remove redundancy, yet is computationally infeasible, often necessitating the use of search heuristics [17], [40]. FRN allows the incorporation of correlation information by only comparing select n-grams (ones from parallel relation categories within the FRN).

### 3.2.3 The Complete Network

Fig. 2 shows the entire FRN, consisted of the nodes previously described in Table 3. The network encompasses 22 n-gram feature category nodes and numerous subsumption and parallel relations between these nodes. The detailed list of relations is presented in Table 5. The order in which the relations are applied is important to ensure that redundant and irrelevant attributes are removed correctly. Subsumption relations are applied prior to parallel relations. Furthermore, subsumption relations between n-gram groups within a feature category are applied prior to across category relations (i.e., 1-Word → 2-Word is applied prior to 1-Word → 1-POSWord).

TABLE 5
List of Relations between N-Gram Feature Groups

| Feature Group | Relations |
|---|---|
| **Subsumption Relations** | |
| N-Char | 1-Char → 2-Char, 1-Char → 3-Char, 2-Char → 3-Char |
| N-Word | 1-Word → 2-Word, 1-Word → 3-Word, 2-Word → 3-Word |
| N-POS | 1-POS → 2-POS, 1-POS → 3-POS, 2-POS → 3-POS |
| N-POSWord | 1-POSWord → 2-POSWord, 1-POSWord → 3-POSWord, 2-POSWord → 3-POSWord |
| N-Legomena | 2-Legomena → 3-Legomena |
| N-Semantic | 1-Semantic → 2-Semantic, 1-Semantic → 3-Semantic 2-Semantic → 3-Semantic |
| IEP-A/E | 1-Word → IEP-A, 1-Word → IEP-C, IEP-C → IEP-D, 2-Word → IEP-B, 3-Word → IEP-E, IEP-B → IEP-E |
| Char-Word | 1-Char → 1-Word, 2-Char → 1-Word, 3-Char → 1-Word |
| Word-POSWord | 1-Word → 1-POSWord, 2-Word → 2-POSWord, 3-Word → 3-POSWord |
| POS-POSWord | 1-POS → 1-POSWord, 2-POS → 2-POSWord, 3-POS → 3-POSWord |
| Word-Legomena | 1-Word → 2-Legomena, 2-Word → 3-Legomena |
| **Parallel Relations** | |
| Word-POS | 1-Word — 1-POS, 2-Word — 2-POS, 3-Word — 3-POS |
| Word-Semantic | 1-Word — 1-Semantic, 2-Word — 2-Semantic, 3-Word — 3-Semantic |
| POS-Semantic | 1-POS — 1-Semantic, 2-POS — 2-Semantic, 3-POS — 3-Semantic |
| POSWord-Semantic | 1-POSWord — 1-Semantic, 2-POSWord — 2-Semantic, 3-POSWord — 3-Semantic |

### 3.2.4 Feature Weights: Incorporating Semantic Information

Features weights $w(a_x)$ are computed by considering their occurrence distribution across classes in the training data $wt(a_x)$, as well as their semantic weight $ws(a_x)$, which is based on the degree of subjectivity associated with the n-gram. Utilizing the semantic weight in addition to the training weight is intended to enhance relevance measurement and alleviate overfitting attributable to solely relying on training data for the calculation of feature weights.

An n-gram's potential level of subjectivity is derived from SentiWordNet, a lexical resource that contains three sentiment polarity scores (i.e., positivity, negativity, and objectivity) for synsets consisted of word-sense pairs [9]. SentiWordNet contains scores for over 150,000 words, with scores being on a 0-1 scale. For instance, the synset consisting of the verb form of the word "short" and the word "short-change" has a positive score of 0 and a negative score of 0.75. The semantic weight $ws(a_x)$ for an n-gram is computed by

Let $A = \{a_1, a_2, ...a_n\}$ denote a set of word n - grams //e.g., 1 - Word or 2 - Word
For each $a_x$, where $a_x = (a_{x1}, ...a_{xd})$ denotes a tuple in $A$, the weight for $a_x$ is :

$$w(a_x) = wt(a_x) + ws(a_x)$$

Where $wt(a_x)$ is the weight for feature $a_x$ in the training data, given $v$ and $w$ are part of the set of $c$ class labels, $v \neq w$, and $c \geq 2$ :

$$wt(a_x) = \max_{v,w}\left(P(a_x \mid v)\log\left(\frac{P(a_x \mid v)}{P(a_x \mid w)}\right)\right)$$

And $ws(a_x)$ is the semantic weight for feature $a_x$ :

$$ws(a_x) = \frac{1}{d}\sum_{i=1}^{d}\left(\frac{1}{k}\sum_{j=1}^{k} s(a_{xi}, j)\right)$$

Where $s(a_{xi}, j)$ is the sum of the positive and negative scores for the word $a_{xi}$ and $j$ is one of the $k$ senses of $a_{xi}$ in Senti WordNet

Fig. 3. Weighting mechanism for n-grams.

Let $A = \{a_1, a_2, ...a_n\}$ and $B = \{b_1, b_2, ...b_m\}$ denote two sets of n - grams
//e.g., 1 - Word, 3 - Word, etc.

if $A \rightarrow B$ //$A$ subsumes $B$
For each $a_x$, where $a_x = (a_{x1}, ...a_{xd})$ denotes a tuple in $A$ with $w(a_x) > 0$
Let $C \subseteq B$, where $C = \{c_1, c_2, ...c_y\}$
And each $c_x = (c_{x1}, ...c_{xe})$ denotes a tuple in $C$ with $w(c_x) > 0$
Where the tuple $a_x$ is a part of each $c_x$
if $s(a_x) = s(c_x)$ //check the semantic orientation of the two features
if $w(a_x) \geq w(c_x) - t$
$w(c_x) = 0$

if $A - B$ //$A$ is parallel to $B$
For each $a_x$, where $a_x = (a_{x1}, ...a_{xd})$ denotes a tuple in $A$ with $w(a_x) > 0$
Let $C \subseteq B$, where $C = \{c_1, c_2, ...c_y\}$
And $c_x = (c_{x1}, ...c_{xe})$ denotes a tuple in $C$ with $w(c_x) > 0$
Where each $c_x$ is potentially correlated with $a_x$
if $Corr(a_x, c_x) \geq p$
if $w(a_x) \geq w(c_x)$ then $w(c_x) = 0$
if $w(a_x) < w(c_x)$ then $w(a_x) = 0$

Where :
$Corr(a, b)$ is the correlation coefficient for $a$ and $b$ across the $m$ training instances :

$$Corr(a,b) = \frac{\sum_{x=1}^{m}(a_x - \bar{a})(b_x - \bar{b})}{\sqrt{\sum_{x=1}^{m}(a_x - \bar{a})^2 \sum_{x=1}^{m}(b_x - \bar{b})^2}}$$

$w(a_x)$ is the weight for feature $a_x$, computed as described in Fig. 3

$$s(a_x) = \arg\max_{v,w}\left(P(a_x \mid v)\log\left(\frac{P(a_x \mid v)}{P(a_x \mid w)}\right)\right)$$

$t$ and $p$ are predefined thresholds //we used $t = 0.05$ and $p = 0.90$

Fig. 4. The FRN algorithm.

determining the average polarity value across the individual tokens encompassed within the n-gram. For each token $a_{xi}$, the polarity value is the average of the sum of its positive and negative scores for each word-sense pair $s(a_{xi}, j)$ in SentiWordNet, where $j$ is one of the $k$ senses of $a_{xi}$.

Fig. 3 shows the weighting formulation for word n-grams. Other n-gram feature categories use a similar formulation, with minor differences in the computation of $ws(a_x)$. In the case of legomena n-grams, the polarity values are only averaged across words that are not hapax legomena or dis legomena. For POSWord n-grams, the word polarity values are only computed for word-sense pairs, where the sense has the same POS as that of the tag associated with the word. POS tag n-gram's semantic weights are the average of their individual tag's polarity values. A tag's value is computed by first identifying the set of words within the training data that have the tag's POS. Next, the average score across these words is calculated from SentiWordNet for word-sense pairs, where the sense has the same POS as the tag. In the case of semantic class n-grams, the polarity value for a particular token is the average of s(axi, j) scores for all words associated with the token's semantic class. Character n-grams do not receive a semantic weight (i.e., $ws(a_x) = 0$).

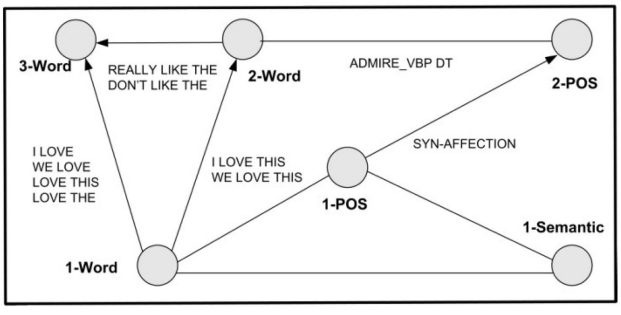
Fig. 4 describes the FRN algorithm details. Given feature a from category A, we first find the feature categories that are subsumed by A (based on the precedence defined in Table 5). Then, all features from these categories containing

| Orientation | Sentence |
|---|---|
| Positive | I LOVE THIS DIGITAL CAMERA. |
| Positive | I LOVE THE POWERFUL LENS A LOT TOO! |
| Positive | I REALLY LIKE THE COMPACT SIZE OF THIS CAMERA. |
| Negative | I DON'T LIKE THE AUTOFOCUS FUNCTION THOUGH. |
| Negative | THE BATTERY LIFE ALSO LEAVES MUCH TO BE DESIRED. |
| Negative | THE FLIMSY CAMERA DESIGN IS JUST NOT VERY FLATTERING. |

(a)

| Feature | Category | Orientation | Weight | FRN Weight |
|---|---|---|---|---|
| LOVE | 1-Word | Positive | 1.0000 | 1.0000 |
| I LOVE | 2-Word | Positive | 1.0000 | 0 |
| WE LOVE | 2-Word | Positive | 1.0000 | 0 |
| LOVE THIS | 2-Word | Positive | 1.0000 | 0 |
| LOVE THE | 2-Word | Positive | 1.0000 | 0 |
| I LOVE THIS | 3-Word | Positive | 1.0000 | 0 |
| WE LOVE THE | 3-Word | Positive | 1.0000 | 0 |
| REALLY LIKE | 2-Word | Positive | 1.0000 | 1.0000 |
| DON'T LIKE | 2-Word | Negative | 1.0000 | 1.0000 |
| REALLY LIKE THE | 3-Word | Positive | 1.0000 | 0 |
| DON'T LIKE THE | 3-Word | Negative | 1.0000 | 0 |
| ADMIRE_VBP | 1-POS | Positive | 0.8239 | 0.8239 |
| SYN-AFFECTION | 1-Semantic | Positive | 0.8239 | 0 |
| ADMIRE_VBP DT | 2-POS | Positive | 0.4621 | 0 |
| LIKE THE | 2-Word | None | 0 | 0 |
| LIKE | 1-Word | None | 0 | 0 |

(b)




(c)

Fig. 5. Example application of FRN to a six-sentence testbed. (a) Example sentences. (b) Feature weights. (c) Feature relation network.

the substring $a$ and having the same semantic orientation are retrieved. The semantic orientation of a feature is defined as the class for which the attribute has the highest probability of occurring. The semantic orientation of features is compared to avoid having features such as DON'T LIKE get subsumed by the unigram LIKE (since the two features have opposing semantic orientations). Feature weights are computed using the procedure described in the prior section and Fig. 3. The weights for the retrieved features are compared against that of $a$, and only those features are retained with a weight greater than $a$ by some threshold $t$.

The parallel relations are enforced as follows: Given feature $a$ from category A, we find the feature categories that are parallel to A. Features from these categories with potential co-occurrence with $a$ are retrieved. The correlation coefficient for these features is computed in comparison with a. If the coefficient is greater than or equal to some threshold $p$, one of the features is removed. We remove the feature with the lower weight (ties are broken arbitrarily). It is important to note that for subsumption and parallel relations, only features still remaining in the feature set are analyzed and/or retrieved (i.e., ones with a weight greater than 0).

TABLE 6
Description of Online Review Testbeds

| Test Bed | Source | # Reviews | # Classes |
|---|---|---|---|
| Digital Cameras | www.epinions.com | 2,000 | 5 (1-5 Stars) |
| Automobiles | www.edmunds.com | 2,000 | 5 (1,3,5,7,9 Stars) |
| Movies | www.rottentomatoes.com | 2,000 | 2 (Positive, Negative) |

Although FRN utilizes subsumption relations as does FSH, it differs from FSH [34] in many ways. First, FRN incorporates seven n-gram feature categories whereas FSH only employs word n-grams and information extraction patterns. Second, FSH utilizes a weighting function that incorporates a unique training data-based weighting heuristic $wt(a_x)$ and a semantic weighting heuristic based on an independent lexicon $ws(a_x)$, while FSH utilizes the feature's IG score. Third, FRN incorporates subsumption and parallel relations, while FSH only uses subsumption. Fourth, FRN represents relations in a network, where features from any category can potentially be removed. In contrast, FSH uses a tree representation, where all features from the highest level node (i.e., word unigrams) are always retained.

Fig. 5 shows an illustration of the FRN applied to a six-sentence testbed (three positive and three negatively oriented sentences). The table in the bottom left corner shows the feature weights for many key categories (e.g., word, POS, and semantic n-grams). The weights depicted include the initial $w(a_x)$, the $wt(a_x)$ based on the six-sentence testbed, $ws(a_x)$, and the adjusted $w(a_x)$ after the FRN has accounted for redundancy. The FRN is able to remove redundant or less useful n-grams, keeping only 6 of the 16 features shown. For example, the bigram I LOVE gets subsumed by the unigram LOVE. Similarly, the semantic class unigram SYN-Affection is parallel to the POS tag ADMIRE_VBP, and therefore, removed. Details for each removed n-gram are provided in the FRN on the right-hand side of the diagram. It is important to note that only the portion of the FRN, which is relevant to these features, is shown. The removed n-grams are placed next to the subsumption or parallel relation responsible for their removal. These features correspond to the features with an adjusted $w(a_x)$ of 0.

## 4 Experiments

We conducted opinion classification experiments on three review testbeds, shown in Table 6. The first contained digital camera reviews collected from Epinions. This testbed featured 1-5 star reviews. We only used whole star reviews (i.e., no half star reviews were included). The second testbed encompassed automobile reviews taken from Edmunds. These reviews were on a continuous 10-point scale. We discretized them into five classes by taking all odd integer reviews. For example, all reviews between 1.0 and 1.99 were assigned 1 star while reviews between 3.0 and 3.99 were considered 2-star reviews. The third testbed was a benchmark movie review data set developed by Pang et al. [28]. This data set contains reviews taken from Rotten Tomatoes

that are either positive or negative. For each testbed, we used a total of 2,000 reviews.

For each testbed, we performed fivefold cross validation on the 2,000 reviews. These reviews were balanced across classes. Hence, there were 400 reviews per class for the digital camera and automobile testbeds and 1,000 reviews per class for the movie review testbed. All experiments were run using WEKA's linear kernel SVM classifier [43]. Feature presence was used as opposed to frequency since it has wielded better results in past research using n-grams for opinion classification [27], [28]. Hence, we used binary feature vectors (1 if the n-gram is present in the document, 0 if it is not present).

The following two metrics were used. The percentage within-one accuracy was incorporated since multiclass opinion classification, involving three or more classes, can be challenging given the relationship and subtle differences between semantically adjacent classes. It is often difficult even for humans to accurately differentiate between, for instance, one- and two-star reviews [8].

$$\%\ \text{Accuracy} = \frac{\#\ \text{correctly assigned}}{\#\ \text{total reviews}},$$

$$\%\ \text{WithinOne} = \frac{\#\ \text{assigned within one class of correct}}{\#\ \text{total reviews}}.$$

Based on our research design, four different experiments were conducted. In Experiment 1, we compared the proposed FRN against various univariate feature selection methods using the extended n-gram feature set as well as word n-grams and a bag-of-words baseline. Experiment 2 compared the FRN method against previously used multivariate feature selection methods. In Experiment 3, FRN was evaluated against various hybrid feature selection methods. Experiment 4 presents ablation and parameter testing results for FRN. In all experiments, FRN was run using $t = 0.05$ and $p = 0.90$.

### 4.1 Experiment 1: Comparison of FRN against Univariate Feature Selection Methods

We ran the FRN in comparison with LL, IG, and CHI. All four of these feature selection methods were run on the extended feature set described in Section 3.1, which encompassed the word, POS, POSWord, character, legomena, syntactic, and semantic n-grams. In order to assess the impact of using the extended feature set, we also compared two additional feature sets: bag-of-words and word n-grams. These feature sets were only run in conjunction with LL, resulting in two additional feature/feature selection combinations, BOW/LL and WNG/LL. BOW/LL constituted a baseline while WNG/LL was employed since it had performed well in prior opinion classification studies [27]. For the three feature sets (i.e., all n-grams, WNG, and BOW), we extracted all feature occurring at least three times [2], [28]. The extracted features were ranked using the aforementioned four feature selection methods on the training data for each of the five cross-validation folds. Hence, for each fold, the weights for all features occurring three times or more in the 1,600 training reviews were computed.

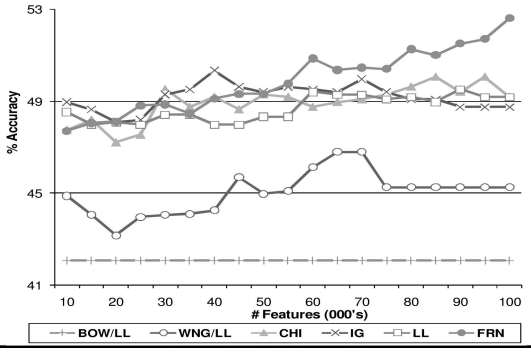
When comparing feature sets and selection methods, it is difficult to decide upon the number of features that should be included. Different feature set sizes can wield varying performance depending on the nature of the features and selection methods employed. In order to allow a fair comparison between feature selection methods, we evaluated the top 10,000 to 100,000 features (i.e., the highest weighted/ranked attributes), in 2,500 feature increments. Hence, 37 feature quantities were used for all three feature sets. The total number of BOW typically did not exceed 20,000, so only that many were evaluated. Such a setup is consistent with experimental designs used in prior research, e.g., [15], [34].

Fig. 6 shows the results for all six methods across the three testbeds. The table on the left of the figure shows the best percentage accuracy, the number of features used to attain these best results, pairwise t-test results using this number of features on random 90-10 training-testing splits ($n = 30$), the area under the curve (AUC), and p-values for pairwise t-tests across the different feature subset sizes ($n = 37$). The first t-test was intended to measure the significance of the best results, while the second measured the overall effectiveness across feature subset sizes. BOW/LL was not compared on the second t-test, since it did not have enough features to generate a sufficient number of feature subsets. The charts on the right show the results for all 37 feature subsets (using the top 10,000 to 100,000 features). Looking at the left side of Fig. 6, FRN outperformed LL, IG, CHI, WNG/LL, and BOW/LL on all three testbeds in terms of best accuracy and AUC. FRN's best accuracy values were 3-4 percent better than any of the comparison techniques across all three testbeds. Based on the pairwise t-test results, FRN significantly outperformed the comparison methods, with all p-values significant at $\text{alpha} = 0.05$.

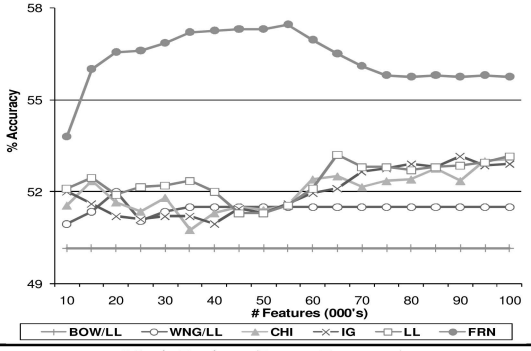
The charts on the right side of Fig. 6 show the accuracies for the feature selection methods, using between 10,000 and 100,000 features. FRN outperformed LL, CHI, WNG/LL, and BOW/LL on all three testbeds by a wide margin, with considerably better accuracy on virtually all feature subset sizes. It also outperformed IG on all but one feature subset size on the movie and automobile review data sets. However, IG had slightly better accuracy on a few of the 37 feature subset sizes on the digital camera testbed. Nevertheless, FRN had a higher AUC and its best accuracy was 2 percent greater than that of IG.

Looking at the results by feature set, techniques that utilized the extended feature set (i.e., LL, IG, and CHI) outperformed WNG/LL and the BOW/LL baseline on the digital camera data sets. They also had slightly better performance on the automobile data set. However, WNG/LL had better performance on the movie testbed. Overall, the extended feature set did not provide a significant performance increase over word n-grams when using univariate feature selection methods. This is not surprising since the extended feature set includes many redundant attributes across the various categories, which univariate feature selection methods are unable to remove. Consequently, the univariate methods require more attributes from the extended feature set to get the necessary depth required for enhanced opinion classification accuracy; only a subset of the highest weighted features is truly providing additional discriminatory potential. This is evidenced by

| Selection Method | Best Acc. | T-test p-values | # Feat. | AUC | T-test p-values |
|---|---|---|---|---|---|
| **Digital Cameras (Epinions)** | | | | | |
| FRN | 52.60 | - | 100K | 897.35 | - |
| IG | 50.32 | < 0.001 | 40K | 885.40 | 0.027 |
| CHI | 50.05 | < 0.001 | 85K | 881.15 | < 0.001 |
| LL | 49.51 | < 0.001 | 90K | 876.20 | < 0.001 |
| WNG/LL | 46.78 | < 0.001 | 65K | 810.26 | < 0.001 |
| BOW/LL | 42.07 | < 0.001 | 10K | 757.26 | n/a |




| | | | | | |
|---|---|---|---|---|---|
| **Automobiles (Edmunds)** | | | | | |
| FRN | 57.45 | - | 55K | 1015.73 | - |
| LL | 53.20 | < 0.001 | 65K | 942.03 | < 0.001 |
| CHI | 53.05 | < 0.001 | 100K | 935.85 | < 0.001 |
| IG | 53.15 | < 0.001 | 90K | 935.20 | < 0.001 |
| WNG/LL | 52.00 | < 0.001 | 20K | 926.48 | < 0.001 |
| BOW/LL | 50.15 | < 0.001 | 10K | 902.70 | n/a |




| | | | | | |
|---|---|---|---|---|---|
| **Movie Reviews (Rotten Tomatoes)** | | | | | |
| FRN | 89.65 | - | 55K | 1583.18 | - |
| WNG/LL | 85.95 | < 0.001 | 65K | 1538.94 | < 0.001 |
| IG | 85.70 | < 0.001 | 50K | 1540.33 | < 0.001 |
| CHI | 85.50 | < 0.001 | 65K | 1530.65 | < 0.001 |
| LL | 85.80 | < 0.001 | 100K | 1527.20 | < 0.001 |
| BOW/LL | 83.95 | < 0.001 | 15K | 1508.25 | n/a |

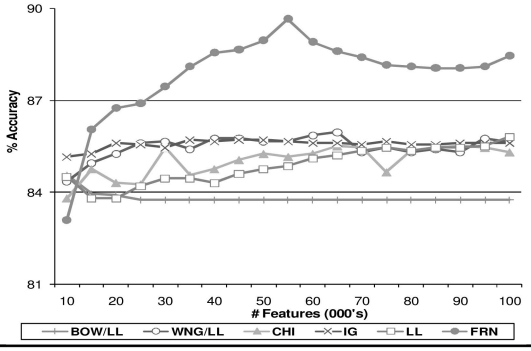
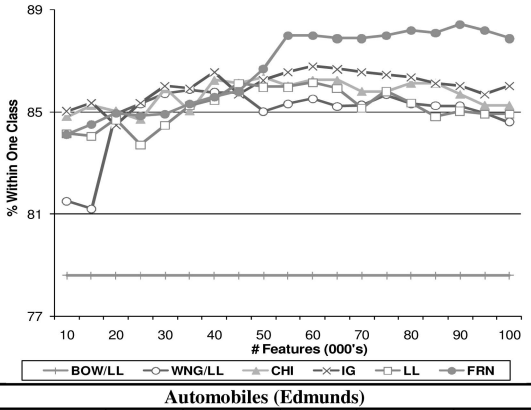



Fig. 6. Accuracy results for FRN and univariate methods.

the general upward slope of LL, CHI, and IG as the feature subset sizes increase.

The results emphasize the need to combine rich, extended feature sets with more powerful feature selection techniques

| Selection Method | Best Acc. | T-test p-values | # Feat. | AUC | T-test p-values |
|---|---|---|---|---|---|
| **Digital Cameras (Epinions)** | | | | | |
| FRN | 88.42 | - | 90K | 1561.03 | - |
| IG | 86.78 | < 0.001 | 60K | 1548.25 | 0.012 |
| CHI | 86.34 | < 0.001 | 50K | 1542.84 | 0.002 |
| LL | 86.14 | < 0.001 | 60K | 1533.31 | < 0.001 |
| WNG/LL | 85.85 | < 0.001 | 35K | 1530.09 | < 0.001 |
| BOW/LL | 78.59 | < 0.001 | 10K | 1414.62 | n/a |




| | | | | | |
|---|---|---|---|---|---|
| **Automobiles (Edmunds)** | | | | | |
| FRN | 90.70 | - | 40K | 1615.80 | - |
| LL | 88.20 | < 0.001 | 20K | 1579.58 | < 0.001 |
| CHI | 88.10 | < 0.001 | 70K | 1576.03 | < 0.001 |
| IG | 88.00 | < 0.001 | 85K | 1575.98 | < 0.001 |
| WNG/LL | 85.95 | < 0.001 | 30K | 1543.03 | < 0.001 |
| BOW/LL | 80.55 | < 0.001 | 10K | 1449.90 | n/a |

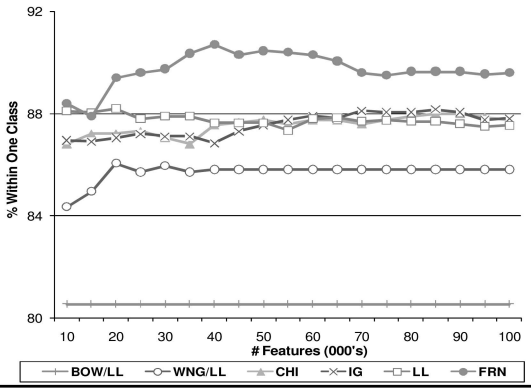



Fig. 7. Within-one results for FRN and univariate methods.

capable of exploiting the additional information these feature sets can provide while overcoming the increased noise and redundancy levels that are inevitable. The experiments demonstrate FRN's ability to garner enhanced performance when using the extended feature set (as compared to existing univariate feature selection methods).

Fig. 7 shows the percentage within-one results on the digital camera and automobile testbeds (the movie review data set only had two classes). The within-one accuracies tended to fall in the 86-90 percent range for FRN, LL, IG, and CHI; suggesting that the majority of errors do indeed fall within one class (e.g., a 1-star review getting misclassified as a 2-star, or vice versa). FRN outperformed all comparison methods on both testbeds in terms of best within-one accuracy and AUC. FRN's best within-one accuracy values were at least 2-3 percent better than the comparison techniques, with all pairwise t-test p-values significant at $\mathrm{alpha} = 0.05$. Based on the charts on the right side of Fig. 7, FRN outperformed LL, CHI, IG, WNG/LL,

and BOW/LL on virtually all feature subset sizes on the automobile testbed and for all feature subset sizes larger than 45,000 on the digital camera data set.

### 4.2 Experiment 2: Comparison of FRN against Multivariate Feature Selection Methods

We compared the FRN against multivariate feature selection methods. The comparison methods incorporated were RFE, FSH, DLA, CMIM, DTM, GA, and PCA. For RFE and DTM, we began with the 200,000 most frequently occurring features in the training data (for that particular fold). Since these multivariate methods' underlying classification models analyze the entire feature set in unison, additional features could not be included for computational reasons. RFE was run using a linear kernel SVM classifier. Each iteration, the 2,500 features with the lowest SVM weights were eliminated (as described in Section 2.3) until only 10,000 features remained. FSH was run using a subsumption threshold of 0.05, since this yielded the best results on the testing data. PCA was run using a sparse implementation designed to reduce computational times and virtual memory constraints. The PCA output was used as input for DLA [50], which was run using a weight of 1, and with 10-20 nearest neighbors (10 on automobiles, 20 on digital cameras and movie reviews), as these settings attained the best results on the testing data. Since CMIM is a binary technique [14], its feature rankings on the multiclass data sets were the average of all binary class comparisons' feature scores (i.e., the average of the 10 comparisons' scores per feature on the automobile and digital camera testbeds). DTM was run iteratively, where for each iteration, all features selected by DTM were added to the feature set with a rank lower than the features added in the previous iteration. The GA was run for 100 generations with a population size of 30, a crossover rate of 0.60, and mutation probability of 0.001 [3]. It used twofold cross validation on the training data for each fold (run using an SVM classifier) to assess the fitness of a particular solution. Since GA, DLA, and PCA do not rank the feature space (instead performing subset selection or transformation-based reduction), the number of features listed is the amount upon which these methods were applied, not the amount actually selected. For example, the GA value for 20,000 features indicates the results attained by the GA when selecting a subset of the 20,000 most frequently occurring features. For the seven comparison methods (as with FRN), we again evaluated all 37 feature subsets ranging from the top 10,000 to 100,000 features (in 2,500 feature increments).

Fig. 8 show the accuracies for FRN in comparison with the multivariate feature selection methods. FRN had the highest best accuracy value and the greatest AUC on all three testbeds. It significantly outperformed all comparison methods in terms of best accuracy and AUC (all p-values less than 0.05) on the automobile and movie review testbeds. Looking at the charts on the right side of Fig. 8, FRN's performance was far better on those two testbeds, with accuracy values generally 3-6 percent higher than the nearest comparison technique. On the digital camera data set, FRN significantly outperformed all comparison methods in terms of best overall accuracy. For the t-tests across feature set sizes, it significantly outperformed DLA, CMIM,

| Selection Method | Best Acc. | T-test p-values | # Feat. | AUC | T-test p-values |
|---|---|---|---|---|---|
| **Digital Cameras (Epinions)** | | | | | |
| FRN | 52.60 | - | 100K | 897.35 | - |
| RFE | 50.27 | < 0.001 | 70K | 895.96 | 0.372 |
| FSH | 50.05 | < 0.001 | 25K | 891.48 | 0.281 |
| DLA | 51.80 | 0.037 | 100K | 889.34 | 0.015 |
| CMIM | 50.60 | < 0.001 | 85K | 877.05 | < 0.001 |
| DTM | 50.93 | 0.003 | 90K | 895.96 | 0.344 |
| GA | 47.46 | < 0.001 | 50K | 840.64 | < 0.001 |
| PCA | 50.35 | < 0.001 | 80K | 851.08 | < 0.001 |
| **Automobiles (Edmunds)** | | | | | |
| FRN | 57.45 | - | 55K | 1015.73 | - |
| RFE | 53.25 | < 0.001 | 85K | 952.05 | < 0.001 |
| FSH | 53.55 | < 0.001 | 65K | 959.30 | < 0.001 |
| DLA | 53.60 | < 0.001 | 60K | 950.95 | < 0.001 |
| CMIM | 53.75 | < 0.001 | 35K | 948.23 | < 0.001 |
| DTM | 53.15 | < 0.001 | 100K | 947.40 | < 0.001 |
| GA | 52.35 | < 0.001 | 80K | 920.90 | < 0.001 |
| PCA | 51.95 | < 0.001 | 60K | 920.50 | < 0.001 |
| **Movie Reviews (Rotten Tomatoes)** | | | | | |
| FRN | 89.65 | - | 55K | 1583.18 | - |
| RFE | 86.25 | < 0.001 | 70K | 1547.10 | < 0.001 |
| FSH | 86.50 | < 0.001 | 70K | 1541.50 | < 0.001 |
| DLA | 86.90 | < 0.001 | 60K | 1556.40 | < 0.001 |
| CMIM | 86.55 | < 0.001 | 50K | 1537.98 | < 0.001 |
| DTM | 85.85 | < 0.001 | 100K | 1535.83 | < 0.001 |
| GA | 85.55 | < 0.001 | 50K | 1524.08 | < 0.001 |
| PCA | 85.50 | < 0.001 | 50K | 1521.43 | < 0.001 |

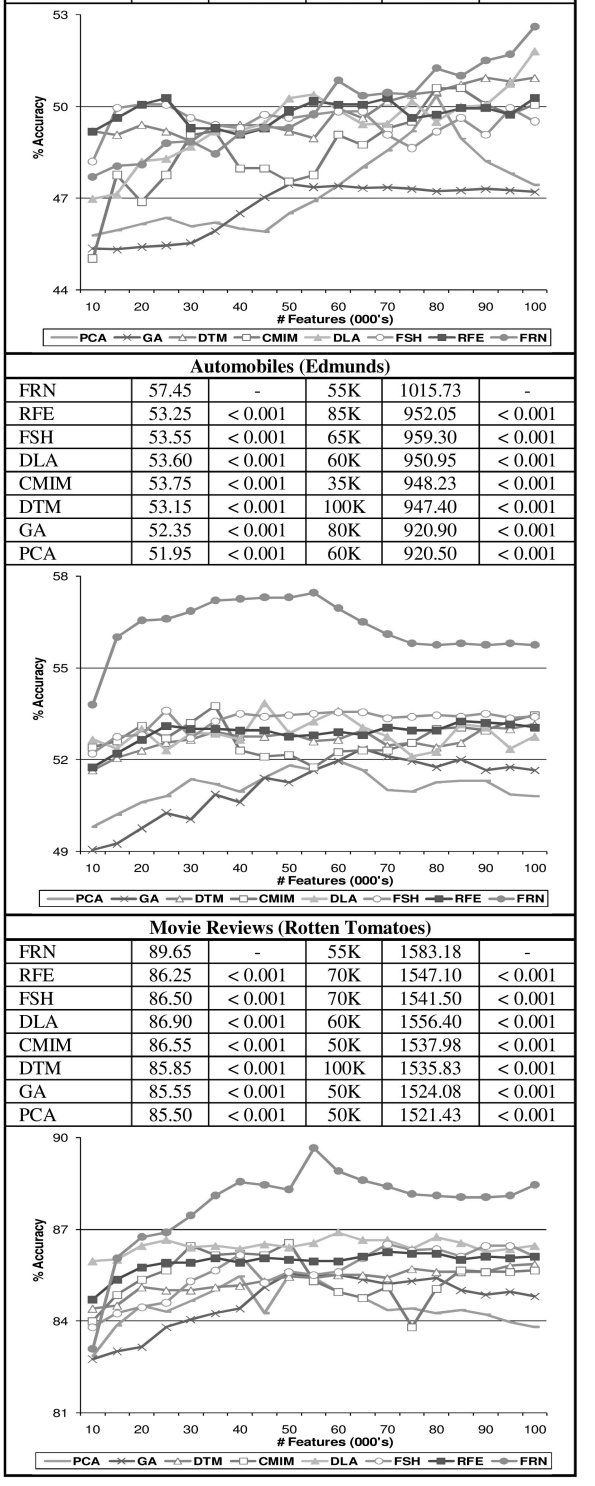
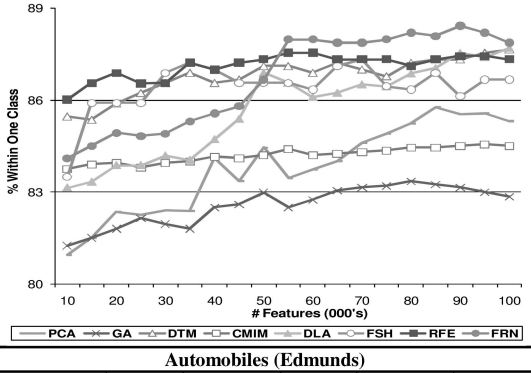



Fig. 8. Accuracy results for FRN and multivariate methods.

| Selection Method | Best Acc. | T-test p-values | # Feat. | AUC | T-test p-values |
|---|---|---|---|---|---|
| **Digital Cameras (Epinions)** | | | | | |
| FRN | 88.42 | - | 90K | 1561.03 | - |
| RFE | 87.04 | < 0.001 | 55K | 1561.14 | 0.464 |
| FSH | 87.32 | 0.011 | 70K | 1556.78 | 0.148 |
| DLA | 87.70 | 0.029 | 100K | 1542.37 | < 0.001 |
| CMIM | 84.55 | < 0.001 | 95K | 1515.63 | < 0.001 |
| DTM | 87.05 | < 0.001 | 100K | 1555.18 | 0.169 |
| GA | 83.35 | < 0.001 | 80K | 1486.73 | < 0.001 |
| PCA | 85.76 | < 0.001 | 85K | 1508.78 | < 0.001 |
| **Automobiles (Edmunds)** | | | | | |
| FRN | 90.70 | - | 40K | 1615.80 | - |
| RFE | 87.50 | < 0.001 | 70K | 1570.73 | < 0.001 |
| FSH | 86.35 | < 0.001 | 25K | 1543.90 | < 0.001 |
| DLA | 87.35 | < 0.001 | 70K | 1560.55 | < 0.001 |
| CMIM | 87.60 | < 0.001 | 100K | 1557.98 | < 0.001 |
| DTM | 87.60 | < 0.001 | 50K | 1571.85 | < 0.001 |
| GA | 86.10 | < 0.001 | 90K | 1542.93 | < 0.001 |
| PCA | 83.90 | < 0.001 | 95K | 1501.25 | < 0.001 |



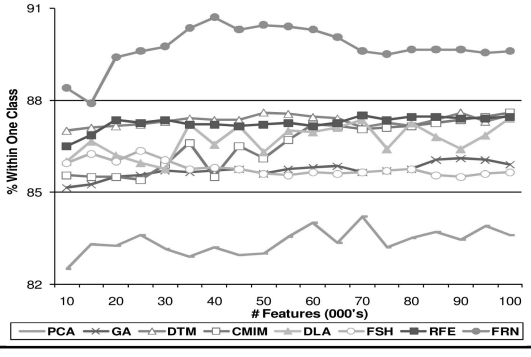




Fig. 9. Within-one results for FRN and multivariate methods.

GA, and PCA. But the performance gain over RFE, DTM, and FSH was not significant (p-values of 0.372 0.344, and 0.281, respectively): these methods had better performance for feature subset sizes less than 50,000-60,000 features while FRN outperformed them on the larger subset sizes.

With respect to the comparison methods, RFE, DLA, and FSH had the best performance. For RFE and FSH, these results are consistent with prior feature selection studies, where these methods also performed well [3], [15], [34]. Although DLA has not previously been applied to text feature selection problems [50], it attained some of the highest best accuracy values (after FRN), suggesting that the method could be highly useful in future sentiment analysis work. The GA had difficulty converging due to the large, noisy input feature spaces. This caused it to perform poorly as compared to prior research where it fared well on feature set sizes under 30,000 features [3].

Fig. 9 shows the percentage within-one results. FRN outperformed all comparison multivariate methods on both testbeds in terms of best within-one accuracy. FRN's best within-one accuracy values were generally 1.5 to 3 percent better than the best comparison techniques. It also outperformed FSH, DLA, CMIM, DTM, GA, and PCA in terms of AUC on both testbeds, and RFE on the automobile testbed. However, its AUC was not significantly better than RFE, FSH, or DTM on the digital camera data set, with RFE attaining a marginally higher AUC value. Once again, this was attributable to FRN being outperformed by these methods for select feature subset sizes below 50,000.

### 4.3 Experiment 3: Comparison of FRN against Hybrid Feature Selection Methods

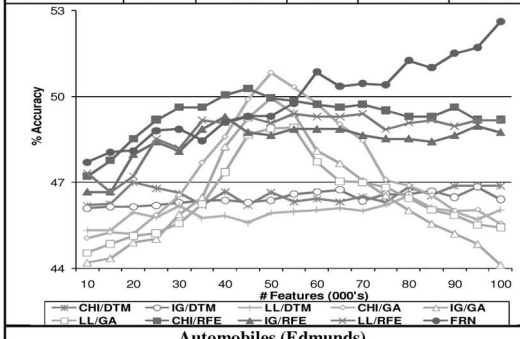
Various hybrid feature selection techniques, which combined univariate and multivariate methods, were compared against FRN. The hybrid methods used the univariate techniques to rank attributes in the feature space. A subset of these ranked attributes (the top 150,000) was then input into the multivariate methods. The initial number of ranked features incorporated into the multivariate component of the hybrid techniques was determined by trial-and-error, utilizing larger quantities (e.g., 200,000 features) resulted in diminished classification performance and increased runtimes. Three different univariate (IG, CHI, and LL) and multivariate methods (RFE, DTM, and GA) were used, resulting in nine possible hybrid combinations. For instance, the IG/RFE involved running the RFE algorithm on the 150,000 features with the highest IG weights, and recursively eliminating 2,500 features per iteration until only 10,000 remained. As with the previous experiments, each hybrid method was evaluated in terms of its effectiveness on subsets of 10,000 to 100,000 features, in 2,500 feature increments. Multivariate methods such as CMIM and FSH were not considered since they already utilize the entire feature space in a computationally efficient manner. Combining these techniques with univariate methods did not yield any significant advantage in terms of accuracy or computation times.

Fig. 10 shows the accuracies for FRN in comparison with the hybrid feature selection methods. FRN had the highest best accuracy value, the greatest AUC, and significantly outperformed all nine comparison methods (all p-values less than 0.05) in terms of best accuracy and best within-one, as well as for the AUC values across the 37 feature subset sizes, for all three testbeds. With respect to the comparison hybrid methods, LL/RFE had the best overall performance.

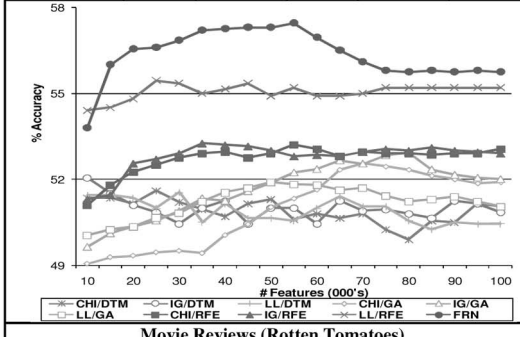
In general, the choice of multivariate method seemed to have a greater impact on performance than the associated univariate method. Hybridizations involving RFE outperformed those containing GA or DTM in terms of their AUC values, while GA outperformed DTM. The hybridizations involving GAs tended to have arc-shaped performance curves across the feature subsets, with diminishing performance for higher feature subset sizes. This was due to their inability to converge once the feature spaces grew too large.

Fig. 11 shows the percentage within-one results. FRN had the best AUC values and significantly outperformed all nine comparison methods for all three testbeds. FRN's best within-one accuracy values were at least 1-2 percent better

| Selection Method | Best Acc. | T-test p-values | # Feat. | AUC | T-test p-values |
|---|---|---|---|---|---|
| **Digital Cameras (Epinions)** | | | | | |
| FRN | 52.60 | - | 100K | 897.35 | - |
| CHI/RFE | 50.27 | < 0.001 | 45K | 888.96 | 0.043 |
| LL/RFE | 49.40 | < 0.001 | 70K | 878.08 | < 0.001 |
| IG/RFE | 49.29 | < 0.001 | 40K | 872.62 | < 0.001 |
| CHI/GA | 50.82 | 0.014 | 50K | 855.21 | < 0.001 |
| IG/GA | 49.95 | < 0.001 | 50K | 838.64 | < 0.001 |
| LL/GA | 48.91 | < 0.001 | 55K | 838.24 | < 0.001 |
| CHI/DTM | 46.89 | < 0.001 | 90K | 837.47 | < 0.001 |
| IG/DTM | 46.84 | < 0.001 | 95K | 835.96 | < 0.001 |
| LL/DTM | 46.54 | < 0.001 | 80K | 826.73 | < 0.001 |




| | | | | | |
|---|---|---|---|---|---|
| **Automobiles (Edmunds)** | | | | | |
| FRN | 57.45 | - | 55K | 1015.73 | - |
| LL/RFE | 55.35 | < 0.001 | 45K | 991.30 | < 0.001 |
| IG/RFE | 53.25 | < 0.001 | 35K | 950.80 | < 0.001 |
| CHI/RFE | 53.05 | < 0.001 | 60K | 949.33 | < 0.001 |
| IG/GA | 52.93 | < 0.001 | 80K | 930.86 | < 0.001 |
| LL/GA | 51.90 | < 0.001 | 50K | 922.55 | < 0.001 |
| CHI/GA | 52.56 | < 0.001 | 70K | 917.62 | < 0.001 |
| IG/DTM | 52.05 | < 0.001 | 10K | 917.45 | < 0.001 |
| LL/DTM | 51.45 | < 0.001 | 15K | 916.05 | < 0.001 |
| CHI/DTM | 51.60 | < 0.001 | 25K | 915.80 | < 0.001 |




| | | | | | |
|---|---|---|---|---|---|
| **Movie Reviews (Rotten Tomatoes)** | | | | | |
| FRN | 89.65 | - | 55K | 1583.18 | - |
| LL/RFE | 87.25 | < 0.001 | 45K | 1567.40 | 0.003 |
| IG/RFE | 85.70 | < 0.001 | 50K | 1540.33 | < 0.001 |
| CHI/RFE | 85.45 | < 0.001 | 55K | 1535.55 | < 0.001 |
| IG/GA | 86.10 | < 0.001 | 70K | 1530.20 | < 0.001 |
| CHI/GA | 84.90 | < 0.001 | 55K | 1512.05 | < 0.001 |
| IG/DTM | 84.65 | < 0.001 | 75K | 1508.68 | < 0.001 |
| LL/GA | 84.35 | < 0.001 | 55K | 1505.98 | < 0.001 |
| LL/DTM | 84.65 | < 0.001 | 75K | 1503.80 | < 0.001 |
| CHI/DTM | 84.00 | < 0.001 | 55K | 1502.28 | < 0.001 |

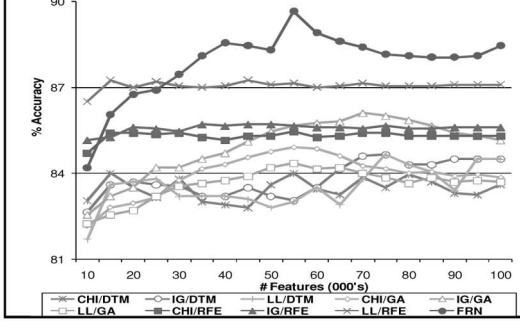
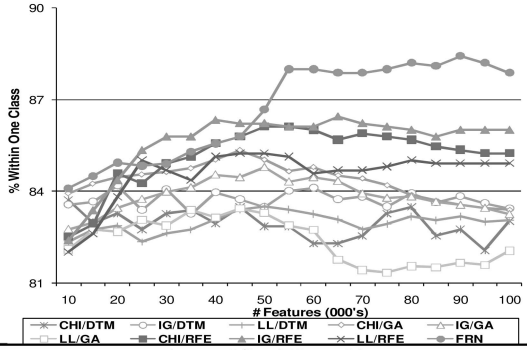



Fig. 10. Accuracy results for FRN and hybrid methods.

| Selection Method | Best Acc. | T-test p-values | # Feat. | AUC | T-test p-values |
|---|---|---|---|---|---|
| **Digital Cameras (Epinions)** | | | | | |
| FRN | 88.42 | - | 90K | 1561.03 | - |
| IG/RFE | 86.34 | < 0.001 | 40K | 1542.57 | < 0.001 |
| CHI/RFE | 86.12 | < 0.001 | 50K | 1534.48 | < 0.001 |
| LL/RFE | 85.25 | < 0.001 | 45K | 1523.37 | < 0.001 |
| CHI/GA | 85.34 | < 0.001 | 45K | 1518.84 | < 0.001 |
| IG/GA | 84.78 | < 0.001 | 50K | 1510.31 | < 0.001 |
| IG/DTM | 84.11 | < 0.001 | 60K | 1507.45 | < 0.001 |
| LL/DTM | 83.51 | < 0.001 | 50K | 1493.88 | < 0.001 |
| CHI/DTM | 83.50 | < 0.001 | 45K | 1492.48 | < 0.001 |
| IG/RFE | 86.34 | < 0.001 | 40K | 1542.57 | < 0.001 |




| | | | | | |
|---|---|---|---|---|---|
| **Automobiles (Edmunds)** | | | | | |
| FRN | 90.70 | - | 40K | 1616.80 | - |
| LL/RFE | 89.70 | 0.046 | 40K | 1610.85 | 0.018 |
| IG/RFE | 88.25 | 0.010 | 40K | 1582.68 | < 0.001 |
| CHI/RFE | 88.10 | 0.003 | 45K | 1580.33 | < 0.001 |
| CHI/DTM | 87.70 | < 0.001 | 60K | 1569.83 | < 0.001 |
| IG/DTM | 87.40 | < 0.001 | 70K | 1568.03 | < 0.001 |
| LL/DTM | 87.00 | < 0.001 | 55K | 1561.30 | < 0.001 |
| CHI/GA | 86.89 | < 0.001 | 65K | 1555.21 | < 0.001 |
| IG/GA | 86.54 | < 0.001 | 50K | 1546.53 | < 0.001 |
| LL/GA | 85.68 | < 0.001 | 50K | 1536.94 | < 0.001 |

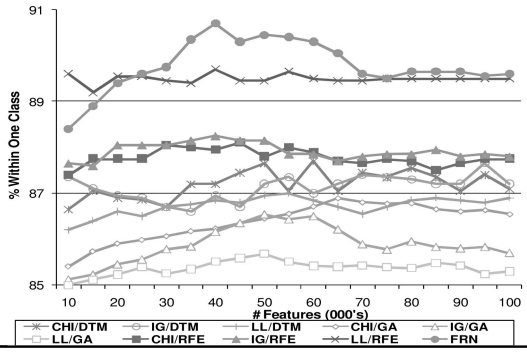



Fig. 11. Within-one results for FRN and hybrid methods.

than the nearest comparison technique, though it generally exceeded comparison methods by 3-4 percent.

### 4.4 Experiment 4: Ablation and Parameter Testing

Ablation testing was performed to evaluate the effectiveness of the key components of FRN: subsumption relations (SRs), parallel relations (PRs), and the semantic weighting (SW). FRN was compared against versions of the algorithm, where some of the three key components were not utilized. For instance, the No-PR version ran the subsumption relations and semantic weighting, but no parallel relations. Similarly, the No-SW/SR version used $wt(a_x)$ as the feature weight and only applied parallel relations (i.e., no subsumption relations and no use of $ws(a_x)$).

The ablation testing results are presented in Fig. 12. The table shows AUC values for each ablation setting across the

| Ablation Setting | Digital Cameras | Automobiles | Movie Reviews |
|---|---|---|---|
| FRN | 897.35 | 1015.73 | 1582.33 |
| No-PR | 882.21 | 1005.78 | 1567.90 |
| No-SR | 876.35 | 1001.63 | 1565.65 |
| No-SR/PR | 871.86 | 999.95 | 1566.10 |
| No-SW | 854.75 | 985.35 | 1560.80 |
| No-SW/PR | 849.18 | 994.68 | 1560.30 |
| No-SW/SR | 839.07 | 991.30 | 1553.43 |

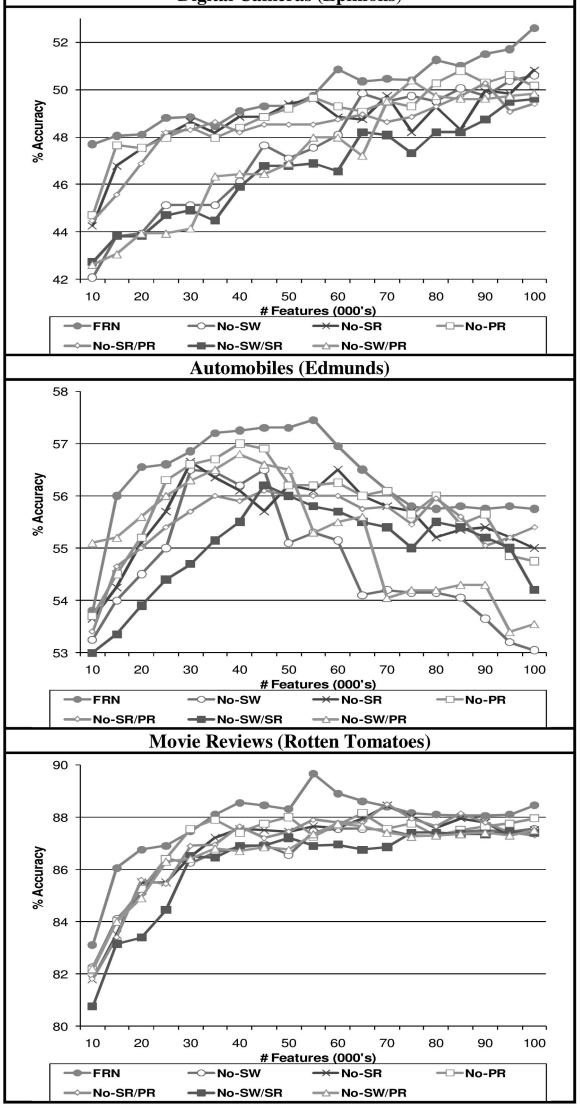



Fig. 12. Accuracy results for FRN ablation testing.

| Parameter Setting SR($t$)-PR($p$) | Digital Cameras | Automobiles | Movie Reviews |
|---|---|---|---|
| 0.05-0.90 | 897.35 | 1015.73 | 1582.88 |
| 0.05-1.00 | 892.74 | 1015.65 | 1576.78 |
| 0.005-0.90 | 900.02 | 1017.78 | 1583.00 |
| 0.005-1.00 | 893.16 | 1016.40 | 1571.42 |
| 0.0005-0.90 | 898.99 | 1017.20 | 1575.81 |
| 0.0005-1.00 | 899.66 | 1016.00 | 1573.99 |
| 0.5-0.90 | 881.36 | 1012.30 | 1556.62 |
| 0.5-1.00 | 883.16 | 1008.85 | 1545.89 |
| 0.05-0.80 | 877.76 | 1006.63 | 1558.92 |
| 0.005-0.80 | 869.68 | 1006.60 | 1549.26 |
| 0.0005-0.80 | 869.90 | 1005.85 | 1548.26 |
| 0.5-0.80 | 866.97 | 1004.03 | 1522.44 |

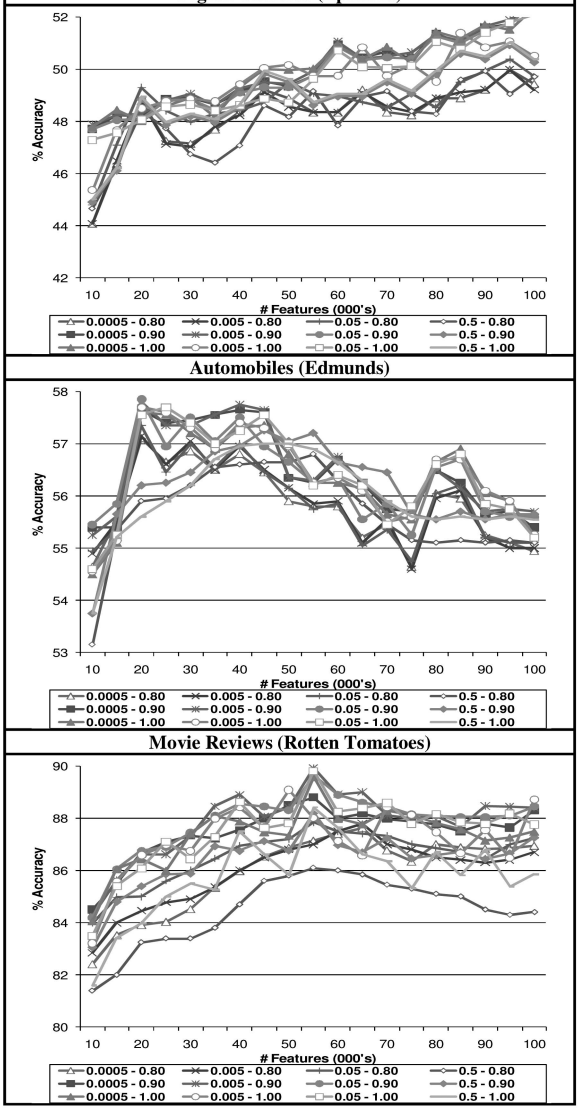



Fig. 13. Accuracy results for FRN parameter testing.

three testbeds, while the figures show the accuracy values across all 37 feature set sizes. All three key components of FRN contributed to the algorithm's overall performance, as evidenced by the fact that removing any component(s) results in a considerable drop in AUC (i.e., at least 10-15 points). With respect to the contribution of individual components, SW seemed to have the biggest impact: variants without SW resulted in the worst performance (e.g., No-SW/PR, No-SW/SR, and No-SW). SR was next in order of impact on AUC, followed by PR.

Two important parameters associated with FRN are the subsumption and parallel relation thresholds ($p$ and $t$ from Fig. 4). Fig. 13 shows the results for different combinations of $t$ and $p$. Four settings were used for $t$ (0.0005, 0.005, 0.05, and 0.5) and three for $p$ (0.80, 0.90, and 1.00), resulting in 12 combinations. FRN performed well for values of $t$ less than or equal to 0.05, and for values of $p$ greater than or equal to 0.90. The results for these ranges of $t$ and $p$ were fairly stable, as signified by the first six rows of AUC values in Fig. 13. For $t = 0.5$ or $p = 0.80$ (bottom six rows of AUC values in Fig. 13), the subsumption/parallel relations were applied too aggressively, resulting in diminished accuracy.

TABLE 7
Average Runtimes (per Fold) for Various Feature Selection Techniques

| Technique | Average Run Time (minutes) | Technique | Average Run Time (minutes) |
|---|---|---|---|
| IG | 54.35 | DLA | 431.18 |
| LL | 60.17 | RFE | 448.80 |
| CHI | 122.42 | LL/DTM | 2104.15 |
| CMIM | 176.13 | IG/DTM | 2210.15 |
| FSH | 203.49 | CHI/DTM | 2293.22 |
| FRN | 326.32 | DTM | 3618.00 |
| IG/RFE | 354.80 | LL/GA | 6830.29 |
| PCA | 361.81 | IG/GA | 6856.50 |
| LL/RFE | 362.34 | CHI/GA | 6907.57 |
| CHI/RFE | 411.67 | GA | 9679.97 |

It is worth noting that the parameter settings employed in experiments 1-3 (i.e., $t = 0.05$, $p = 0.90$) did not yield the best results.

### 4.5 Results Discussion

FRN had significantly higher best accuracy and best percentage within-one values than all 57 comparison conditions, 19 selection methods (univariate, multivariate, and hybrid) across three testbeds (all p-values < 0.001). FRN's AUC for accuracy was significantly better than 54 out of 57 comparison conditions while it significantly outperformed 35 out of 38 comparison conditions in terms of AUC for percentage within-one class. FRN, coupled with the extended feature set, also outperformed WNG and the BOW baseline. Furthermore, the ablation and parameter testing results presented in experiment 4 suggest that all three key components of FRN play an important role and that the algorithm is not overly sensitive to different parameter values for the subsumption and parallel relation thresholds.

In addition to selecting feature subsets capable of providing enhanced sentiment classification performance, FRN also has a faster runtime than many comparison multivariate and hybrid feature selection methods. Table 7 shows the average runtimes for each feature selection technique per fold (i.e., the sum of the fivefold runtimes across the three testbeds, divided by 15). The runtimes are for the amount of time needed to rank the feature space. Since GA is a subset selection method, its number is the average amount of time needed to run the GA on 100,000 features for a single fold. In order to allow a fair comparison, the runtimes listed are all from the same machine. As expected, univariate methods were the fastest since they scored each attribute independently. CMIM, FSH, and FRN had the next shortest runtimes, followed by RFE hybridizations and PCA, with DTM and GA taking the longest. The hybrid methods had shorter runtimes than their multivariate counterparts since the multivariate components of the hybrid techniques had smaller input feature spaces to explore. Additionally, certain hybrid methods outperformed their multivariate counterparts. For instance, CHI/RFE and LL/RFE had better performance than RFE on certain testbeds. Similarly, IG/GA and CHI/GA outperformed GA on some data sets. These results indicate that the use of univariate methods to rank input features can improve accuracy and decrease runtimes for multivariate methods. This is consistent with prior research where hybrid GAs have outperformed standard ones [3].

## 5 Conclusions and Future Directions

In this study, we proposed the use of FRN for improved selection of text attributes for enhanced sentiment classification. FRN's use of syntactic relation and semantic information regarding n-grams enabled it to achieve improved results over various univariate, multivariate, and hybrid feature selection methods. Based on the results attained in this study, we have identified a few future research directions. We believe that FRN may be suitable for other text classification problems, where semantic information is available (e.g., topic, affect, and style classification). We also intend to explore additional potential feature relations. Furthermore, we would like to extend the network by adding additional feature occurrence measurements. In this study, we used feature presence vectors. Other measurements, such as occurrence frequency and various positional/distributional features, could be added, resulting in a multidimensional FRN. Alternate semantic weighting mechanisms could also be explored. Another potential avenue we intend to explore is the development of hybrid feature selection methods that incorporate FRN in conjunction with other multivariate selection techniques.

## Acknowledgments

The authors wish to thank the associate editor and reviewers for their invaluable feedback. They are also grateful to Francois Fleuret and Tianhao Zhang for providing them with implementations of the CMIM and DLA algorithms, respectively.

## References

[1] A. Abbasi and H. Chen, "CyberGate: A System and Design Framework for Text Analysis of Computer Mediated Communication," *MIS Quarterly*, vol. 32, no. 4, pp. 811-837, 2008.

[2] A. Abbasi, H. Chen, S. Thoms, and T. Fu, "Affect Analysis of Web Forums and Blogs Using Correlation Ensembles," *IEEE Trans. Knowledge and Data Eng.*, vol. 20, no. 9, pp. 1168-1180, Sept. 2008.

[3] A. Abbasi, H. Chen, and A. Salem, "Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums," *ACM Trans. Information Systems*, vol. 26, no. 3, article no. 12, 2008.

[4] S. Argamon, C. Whitelaw, P. Chase, S.R. Hota, N. Garg, and S. Levitan, "Stylistic Text Classification Using Functional Lexical Features," *J. Am. Soc. Information Science and Technology*, vol. 58, no. 6, pp. 802-822, 2008.

[5] P.V. Balakrishnan, R. Gupta, and V.S. Jacobs, "Development of Hybrid Genetic Algorithms for Product Line Designs," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 34, no. 1, pp. 468-483, Feb. 2004.

[6] A. Burgun and O. Bodenreider, "Comparing Terms, Concepts, and Semantic Classes in WordNet and the Unified Medical Language System," *Proc. North Am. Assoc. Computational Linguistics Workshop*, pp. 77-82, 2001.

[7] H. Cui, V. Mittal, and M. Datar, "Comparative Experiments on Sentiment Classification for Online Product Reviews," *Proc. 21st AAAI Conf. Artificial Intelligence*, pp. 1265-1270, 2006.

[8] S.R. Das and M.Y. Chen, "Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web," *Management Science*, vol. 53, no. 9, pp. 1375-1388, 2007.

[9] A. Esuli and F. Sebastiani, "SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining," *Proc. Fifth Conf. Language Resources and Evaluation*, pp. 417-422, 2006.

[10] Z. Fei, J. Liu, and G. Wu, "Sentiment Classification Using Phrase Patterns," *Proc. Fourth IEEE Int'l Conf. Computer Information Technology*, pp. 1147-1152, 2004.

[11] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," *J. Machine Learning Research,* vol. 3, pp. 1289-1305, 2004.
[12] M. Gamon, "Sentiment Classification on Customer Feedback Data: Noisy Data, Large Feature Vectors, and the Role of Linguistic Analysis," *Proc. 20th Int'l Conf. Computational Linguistics,* pp. 841-847, 2004.
[13] M. Genereux and M. Santini, "Exploring the Use of Linguistic Features in Sentiment Analysis," *Proc. Corpus Linguistics Conf.,* pp. 27-30, 2007.
[14] F. Fleuret, "Fast Binary Feature Selection with Conditional Mutual Information," *J. Machine Learning Research,* vol. 5, pp. 1531-1555, 2004.
[15] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification Using Support Vector Machines," *Machine Learning,* vol. 46, pp. 389-422, 2002.
[16] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.
[17] M. Hall and L.A. Smith, "Feature Subset Selection: A Correlation Based Filter Approach," *Proc. Fourth Int'l Conf. Neural Information Processing and Intelligent Information Systems,* pp. 855-858, 1997.
[18] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," *Proc. ACM SIGKDD,* pp. 168-177, 2004.
[19] S. Kim and E. Hovy, "Determining the Sentiment of Opinions," *Proc. 20th Int'l Conf. Computational Linguistics,* pp. 1367-1373, 2004.
[20] J. Li, R. Zheng, and H. Chen, "From Fingerprint to Writeprint," *Comm. ACM,* vol. 49, no. 4, pp. 76-82, 2006.
[21] H. Liu and H. Motada, *Feature Extraction, Construction, and Selection—Data Mining Perspective.* Kluwer Academic Publishers, 1998.
[22] H. Liu and L. Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 4, pp. 491-502, Apr. 2005.
[23] G. Mishne, "Experiments with Mood Classification," *Proc. Stylistic Analysis of Text for Information Access Workshop,* 2005.
[24] D. Mladenic, J. Brank, M. Grobelnik, and N. Milic-Frayling, "Feature Selection Using Linear Classifier Weights: Interaction with Classification Models," *Proc. ACM SIGIR,* pp. 234-241, 2004.
[25] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining Product Reputations on the Web," *Proc. ACM SIGKDD,* pp. 341-349, 2002.
[26] T. Nasukawa and T. Nagano, "Text Analysis and Knowledge Mining System," *IBM Systems J.,* vol. 40, no. 4, pp. 967-984, 2001.
[27] V. Ng, S. Dasgupta, and S.M.N. Arifin, "Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews," *Proc. Conf. Computational Linguistics, Assoc. for Computational Linguistics,* pp. 611-618, 2006.
[28] B. Pang, L. Lee, and S. Vaithyanathain, "Thumbs Up? Sentiment Classification Using Machine Learning Techniques," *Proc. Conf. Empirical Methods in Natural Language Processing,* pp. 79-86, 2002.
[29] B. Pang and L. Lee, "A Sentimental Education: Sentimental Analysis Using Subjectivity Summarization Based on Minimum Cuts," *Proc. 42nd Ann. Meeting of the Assoc. Computational Linguistics,* pp. 271-278, 2004.
[30] F. Peng, D. Schuurmans, V. Keselj, and S. Wang, "Automated Authorship Attribution with Character Level Language Models," *Proc. 10th Conf. European Chapter of the Assoc. Computational Linguistics,* 2003.
[31] A. Popescu and O. Etzioni, "Extracting Product Features and Opinions from Reviews," *Proc. Human Language Technology, Empirical Methods in Natural Language Processing,* pp. 339-346, 2005.
[32] E. Riloff and J. Wiebe, "Learning Extraction Patterns for Subjective Expressions," *Proc. Conf. Empirical Methods in Natural Language Processing,* pp. 105-112, 2003.
[33] E. Riloff, J. Wiebe, and T. Wilson, "Learning Subjective Nouns Using Extraction Pattern Bootstrapping," *Proc. Seventh Conf. Natural Language Learning,* pp. 25-32, 2003.
[34] E. Riloff, S. Patwardhan, and J. Wiebe, "Feature Subsumption for Opinion Analysis," *Proc. Conf. Empirical Methods in Natural Language Processing,* pp. 440-448, 2006.
[35] K. Tsutsumi, K. Shimada, and T. Endo, "Movie Review Classification Based on Multiple Classifier," *Proc. 21st Pacific Asia Conf. Language, Information, and Computation,* pp. 481-488, 2007.
[36] P.D. Turney and M.L. Littman, "Measuring Praise and Criticism: Inference of Semantic Orientation from Association," *ACM Trans. Information Systems,* vol. 21, no. 4, pp. 315-346, 2003.
[37] J. Wiebe, T. Wilson, and M. Bell, "Identifying Collocations for Recognizing Opinions," *Proc. Assoc. for Computational Linguistics, European Chapter of the Assoc. for Computational Linguistics Workshop Collocation,* 2001.
[38] J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin, "Learning Subjective Language," *Computational Linguistics,* vol. 30, no. 3, pp. 277-308, 2004.
[39] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment Analyzer: Extracting Sentiments about a Given Topic Using Natural Language Processing Techniques," *Proc. Third IEEE Int'l Conf. Data Mining,* pp. 427-434, 2003.
[40] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," *Proc. 20th Int'l Conf. Machine Learning,* pp. 856-863, 2003.
[41] L. Yu and H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy," *J. Machine Learning Research,* vol. 5, pp. 1205-1224, 2004.
[42] Z. Zhang, "Weighing Stars: Aggregating Online Product Reviews for Intelligent E-Commerce Applications," *IEEE Intelligent Systems,* vol. 23, no. 5, pp. 42-49, Sept. 2008.
[43] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques,* second ed. Morgan Kaufmann, 2005.
[44] C.E. Shannon, "A Mathematical Theory of Communication," *Bell Systems Technical J.,* vol. 27, no. 10, pp. 379-423, 1948.
[45] J.R. Quinlan, "Induction of Decision Trees," *Machine Learning,* vol. 1, no. 1, pp. 81-106, 1986.
[46] A. Abbasi and H. Chen, "Writeprints: A Stylometric Approach to Identity-Level Identification and Similarity Detection in Cyberspace," *ACM Trans. Information Systems,* vol. 26, no. 2, article no. 7, 2008.
[47] W. Bian and D. Tao, "Harmonic Mean for Subspace Selection," *Proc. 19th Int'l Conf. Pattern Recognition,* 2008.
[48] D. Tao, X. Li, X. Wu, and S.J. Maybank, "Geometric Mean for Subspace Selection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 2, pp. 260-274, Feb. 2009.
[49] D. Tao, X. Li, X. Wu, and S.J. Maybank, "General Averaged Divergence Analysis," *Proc. Seventh IEEE Int'l Conf. Data Mining,* pp. 302-311, 2007.
[50] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch Alignment for Dimensionality Reduction," *IEEE Trans. Knowledge and Data Eng.,* vol. 21, no. 9, pp. 1299-1313, Sept. 2009.

**Ahmed Abbasi** received the BS and MBA degrees in information technology from Virginia Tech and the PhD degree in information systems from the University of Arizona. He is currently an assistant professor of information systems at the University of Wisconsin—Milwaukee. He has published several peer-reviewed articles on computer-mediated communication, text mining, online security, and information visualization. His research has appeared in various journals, including the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Intelligent Systems*, the *IEEE Computer*, *ACM Transactions on Information Systems*, *Journal of the American Society for Information Science and Technology*, *MIS Quarterly*, and *Journal of MIS*. He is a member of the IEEE and the Association for Information Systems (AIS).

**Stephen France** received the BS degree in computer science from the University of Durham, the MS degree in statistics and management science from the University of the West of England, and the PhD degree in marketing and supply chain management from Rutgers University. He is currently an assistant professor of marketing at the University of Wisconsin—Milwaukee. He has published several refereed conference papers in the areas of multidimensional scaling, dimensionality reduction, information visualization, and document clustering. He is a member of the INFORMS, the Classification Society, the Psychometric Society, and the IEEE.

**Zhu Zhang** received the BE degree in information systems from Tongji University, the MS degree in information systems from Fudan University, and the PhD degree in information and computer science and engineering from the University of Michigan. He is currently an assistant professor of information systems at the University of Arizona. His work has appeared in the *IEEE Intelligent Systems*, *Journal of the American Society for Information Science and Technology*, *MIS Quarterly*, and *Journal of MIS*, among other outlets. He is a member of the Association for Computational Linguistics (ACL), Association for Information Systems (AIS), and the American Association for Artificial Intelligence (AAAI).

**Hsinchun Chen** received the BS degree from the National Chiao-Tung University, the MBA degree from the State University of New York (SUNY) Buffalo, and the PhD degree in information systems from New York University. He is a professor of information systems and the director of the Artificial Intelligence Lab, University of Arizona. He has authored/edited 20 books, 25 book chapters, and more than 200 SCI journal articles covering digital library, intelligence analysis, biomedical informatics, data/text/web mining, knowledge management, and web computing. He serves on 10 editorial boards, is the editor-in-chief of the *ACM Transactions on Management Information Systems*, and has been an advisor for major US National Science Foundation (NSF), US Department of Justice, US National Library of Medicine, US Department of Defense, US Department of Homeland Security, and other international research programs. He received the IEEE Computer Society 2006 Technical Achievement Award. He is a fellow of the IEEE and the AAAS.