Editor: **Hsinchun Chen,** University of Arizona, hchen@eller.arizona.edu

AI and Opinion Mining, Part 2

As a follow-up to our last Trends & Controversies department, we present two additional articles on opinion mining from distinguished experts in computer science and information systems. These articles present unique innovative research, computational methods, and selected results and examples.

In "Sentiment Quantification," Andrea Esuli and Fabrizio Sebastiani argue that the opinion-mining community has traditionally neglected whether the analysis of large quantities of text should be carried out at the individual or aggregate level. They review several sentiment-quantification methods that can help address some of these issues.

In the third and last paper, "Intelligent Feature Selection for Opinion Classification," Ahmed Abbasi proposes an intelligent feature-selection approach inspired by feature subsumption hierarchies (FSH) that incorporates syntactic and semantic information. The proposed approach illustrates how rich, heterogeneous feature sets, coupled with appropriate feature-selection mechanisms, can improve opinion classification performance.

Sentiment Quantification

Andrea Esuli and Fabrizio Sebastiani, *Italian National Council of Research*

Opinion mining has come to play a key role in text mining applications for customer relationship management, consumer attitude detection, brand and product positioning, and market research. Interest in these applications has spawned a new generation of companies and products devoted to online reputation management, market perception, and online content monitoring.

Historically, one of the most important incarnations of opinion mining has been sentiment classification, the task of classifying a given piece of natural language text (be it a short remark, blog post, or full-blown product review) not according to its topic (as in standard text classification) but according to the opinions expressed in it. One interesting instance of sentiment classification is detecting whether a given product review is positive or negative, which is an example of *binary* classification. More subtly, it might be interesting to detect *how* positive or negative the review is; if we express the possible values on a finite scale of integers, such as between one (very negative) and five (very positive), this is an example of *ordinal* classification (also known as "ordinal regression").

Sentiment classification is pervasive in all contexts where opinions must be mined from large quantities of text. For instance, in a typical customer relationship management application, a company might ask customers to fill out a questionnaire to determine their opinions on a product or service they recently purchased. If the questionnaire contains open questions, the company will need to bin the textual answers into classes that represent different types of opinions. For example, an online bank that polls its customers on how satisfied they are with their online account might use classes such as "satisfied overall," "unhappy with website navigation," "customer ready to churn," and so forth. When the large amount of questionnaires received makes manual processing too expensive or simply infeasible given the time constraints, automatically classifying respondents becomes the only available option. Because the "opinion" dimension is of key importance to this classification endeavor, the technology used must combine sentiment analysis techniques and (more traditional) text classification techniques based on supervised learning.[1]

Published by the IEEE Computer Society

Sentiment classification of textual answers returned within questionnaires could serve other purposes as well. Other applications might include survey coding for the social or political sciences (such as when open questions inquire about the respondents' beliefs, social status, or political leanings)[2] or market research (such as when open questions deal with the respondents' perception of products, brands, or advertising campaigns).

Another important sentiment classification application is managing online product reviews. Such reviews are available across numerous specialized websites (Amazon, Epinions.com, Ratingz.net, and TripAdvisor.com are only a few examples) and increasingly influence consumers' product-purchasing decisions. While structured reviews from such websites consist of a textual product evaluation and a score expressed on an ordered scale of values, many others (such as those to be found in newsgroups, blogs, and other venues for spontaneous discussion) contain only a textual evaluation, with no score attached. These latter reviews are difficult for an automated system to manage, especially when we need to determine, based on the reviews alone, the best perceived product in the lot or whether product x is considered better than product y.

Tools capable of interpreting a text-only product review and classifying it according to how positive it is are thus of the utmost importance. Such a tool would "star-rate" a product review—that is, assign it a certain number of "stars" (from one to five) based on its textual content. Additionally, it could compute the average star-rating obtained by a given product (as resulting from the product reviews written by different consumers) and rank all the products in a given range (for example, all horror movies released between 2006 and 2008 and produced in the US) according to their computed average star-rating.

Individual or Aggregate?

The opinion mining community has traditionally neglected whether the analysis of these large quantities of text should be carried out at the individual or aggregate level. This is an important issue because some of the applications we have discussed so far (namely, open-answer classification for customer satisfaction analysis) demand attention at the individual level, while others (such as open-answer classification for market research or review classification for product or brand positioning) are best analyzed at the aggregate level.

When classifying thousands of questionnaires according to whether the respondent belongs to the class "customer ready to churn," a telecom company is likely interested in accurately classifying each individual customer because it might want to contact them individually to offer improved conditions. Conversely, in a market research application in which the questionnaire asks about the respondent's perception of a given ad campaign, the company is likely not interested in whether a specific individual belongs to the class "liked the campaign," but rather it wants to know the percentage of respondents that belong to the class. Similarly, given a large set of star-rated reviews of a given MP3 player, we are interested in knowing the statistical distribution of the answers across the possible star-ratings, and we are not interested in individual ratings. These examples demonstrate that not all these contexts are alike in terms of the granularity at which the results are to be analyzed. Some applications (ideally) demand that every single item be correctly classified, while others instead (ideally) demand that the true percentage of items that belong to the class be correctly *quantified*. Although in most applications of classification by topic the individual level of analysis seems the more (if not the only) appropriate one, the aggregate level of analysis features prominently in sentiment classification applications. We thus argue for a new focus shift within the opinion-mining community, from sentiment classification to *sentiment quantification*, a shift that recognizes the two as distinct application needs, each requiring specific tools in order to be addressed optimally.

Obviously, classification is a more difficult task than quantification. In fact, the ideal classifier is by definition also an ideal quantifier, but an ideal quantifier is not necessarily an ideal classifier. In fact, to perfectly estimate the percentage of items that belong to the class, a classifier must only deliver an equal number of false positives and false negatives since the two compensate each other when quantifying class frequencies.

Interestingly, George Forman noted only recently (although not in the context of sentiment classification) that the results of classification sometimes need to be analyzed purely at the aggregate level.[3] The history of classification is thus a history of analysis at the individual level.

Evaluating Sentiment Quantification ...

Which mathematical measure should we use to evaluate quantification accuracy? Quite reasonably, for the case of binary classification, Forman proposed the use of normalized cross entropy,[3] better known as

Kullback-Leibler Divergence (KLD) and defined as

$$KLD(p,q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

KLD is a measure of the error made in estimating a true distribution p by means of a predicted distribution q. Thus, KLD is in principle suitable to our needs because quantifying exactly means predicting how the test items are distributed across the classes.

It might seem that optimizing classification *a fortiori* means optimizing quantification. In other words, on the surface it would seem obvious that the more we improve a classifier's accuracy at the individual level, the higher its accuracy at the aggregate level will become, and that the only way to improve a classifier's ability to correctly estimate the distribution of test cases across classes is to improve its ability to classify individual items. Unfortunately, we contend this is not true, or at least that this depends on what we mean by "accuracy at the individual level." To see this, we need to look at the definition of F_1, the standard evaluation function for binary classification, which is defined as

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (1)$$

where TP, FP, and FN indicate the numbers of true positives, false positives, and false negatives, respectively, from a standard contingency table. Equation 1 shows that F_1 deteriorates with $(FP + FN)$ and not with $|FP - FN|$, as would instead be required of a function that truly optimizes quantification. For example, according to F_1, a classifier $\hat{\Phi}_1$ for which $FP = 50$ and $FN = 50$ is worse (all other things being equal) than a classifier $\hat{\Phi}_2$ for which $FP = 0$ and $FN = 10$. However, $\hat{\Phi}_1$ is better than $\hat{\Phi}_2$ according to KLD, and according to any reasonable measure for evaluating quantification accuracy. Indeed, $\hat{\Phi}_1$ is a perfect quantifier since FP and FN are equal and thus compensate each other, so that the distribution of the test items is estimated perfectly.

The situation is the same for ordinal classification, the task we need to solve for star-rating product reviews. The standard evaluation measure for ordinal classification is mean absolute error (MAE), which is the numerical distance between the item's true and predicted classes, averaged across the test items. For instance, assigning two stars to a review that is really worth five stars incurs in an absolute error of three. MAE is obviously not a good measure for ordinal quantification. In fact, an ordinal classifier that has classified all test items correctly aside from swapping equal numbers of items between two classes c_i and c_j, has perfectly estimated the distribution of items across the ordered classes, regardless of the number of swapped items and of the "distance" between c_i and c_j. Examples analogous in spirit to the previous one can show that an ordinal classifier $\hat{\Phi}_1$ might be better than another ordinal classifier $\hat{\Phi}_2$ in terms of MAE but would be worse than $\hat{\Phi}_2$ in terms of any reasonable evaluation function for ordinal quantification.

Therefore, which functions should be used to evaluate ordinal quantification? To the best of our knowledge, we know of no measure that has been proposed for this task. To this purpose, in our ongoing research we are adopting the Earth Mover's Distance (EMD),[4] a function often used in content-based image retrieval for computing the distance between two images' color histograms. EMD computes the minimal cost incurred in turning one distribution into the other, where the cost is computed as the probability mass that must be moved from one class to another, weighted by the distance between the two classes.

… and Optimizing It

The examples of the previous section demonstrate that simply improving classification accuracy is not the optimal way of improving quantification accuracy. This not only indicates that classification and quantification are two different, albeit related tasks, it also indicates that quantification should be tackled according to methods different from the ones that prove optimal for classification.

Concerning this, Forman proposed several learning methods explicitly devised for binary quantification and experimentally showed that they improve quantification accuracy with respect to standard methods originally devised with just (individual) classification in mind.[3] However, none of these methods are based on explicitly optimizing the function eventually used in evaluating quantification. We are currently pursuing this line of research in our ongoing work. In particular, the idea is that of adopting the SVM_{multi} approach,[5] which consists of using a learning device based on support vector machines (SVMs) that lets us optimize any nonlinear evaluation function that can be directly computed from a contingency table, such as KLD. The approach is fundamentally different from conventional learning algorithms: instead of generating a binary classifier that classifies individual test instances one at a time, SVM_{multi} generates a classifier that conceptually classifies an entire set of test instances in one shot. By doing so, SVM_{multi} can optimize properties of entire sets of instances that, as KLD, are not linear functions of individual instances.

We hope to report the results of experimenting with this approach on sentiment quantification data sets in the near future. Concerning the optimization of *ordinal* quantification, instead, further research is still needed to devise ordinal regression methods that can explicitly optimize EMD.

References

1. T. Macer, M. Pearson, and F. Sebastiani, "Cracking the Code: What Customers Say, in their own Words," *Proc. 50th Ann. Conf. Market Research Soc.* (MRS 07), MRS, 2007.
2. D. Giorgetti and F. Sebastiani, "Automating Survey Coding by Multiclass Text Categorization Techniques," *J. Am. Soc. Information Science and Technology*, vol. 54, no. 14, 2003, pp. 1269–1277.
3. G. Forman, "Quantifying Counts and Costs via Classification," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, 2008, pp. 164–206.
4. Y. Rubner, C. Tomasi, and L.J. Guibas, "A Metric for Distributions with Applications to Image Databases," *Proc. 6th Int'l Conf. Vision* (ICCV 98), IEEE CS Press, 1998, pp. 59–66.
5. T. Joachims, "A Support Vector Method for Multivariate Performance Measures," *Proc. 22nd Int'l Conf. Machine Learning* (ICML 05), ACM Press, 2005, pp. 377–384.

Andrea Esuli is a researcher at ISTI-CNR. He has a PhD in information engineering from the University of Pisa, Italy. Contact him at andrea.esuli@isti.cnr.it.

Fabrizio Sebastiani is a senior researcher at ISTI-CNR. He has a "Laurea" degree in computer science from the University of Pisa, Italy. Contact him at fabrizio.sebastiani@isti.cnr.it.

Intelligent Feature Selection for Opinion Classification

Ahmed Abbasi, *University of Wisconsin-Milwaukee*

Although text opinion mining involves many important tasks, accurately assigning sentiment polarities (such as positive, negative, or neutral) and intensities (such as high or low) remains a critical challenge. Given the complexities and nuances associated with opinion classification, it is generally considered more difficult than traditional text mining tasks such as topic-based document categorization. Consequently, prior sentiment-analysis studies have used more sophisticated feature representations, well beyond bag-of-words and word n-grams. The features used include part-of-speech tag n-grams, syntactic phrase patterns, legomena-based collocations, as well as manually and semiautomatically constructed syntactic and semantic phrase patterns and lexicons.[1,2,5] Although these features represent potentially important sentiment discriminators, incorporating them in unison can produce feature spaces spanning tens of thousands of attributes, a situation resulting in the age-old conundrum of disentangling quality from quantity. In addition to the obvious ramifications pertaining to computational feasibility, we must also consider the trade-offs between representational richness and noise, between generalization ability and over-fitting (memorization). Without appropriate feature-selection mechanisms, using large heterogeneous feature spaces is analogous to "throwing the kitchen sink."[3]

This problem is exacerbated by the lack of feature-selection methods specifically crafted for opinion classification. Most existing feature-selection methods are generic techniques that are uniformly applied to input feature value matrices. Examples include information gain, log likelihood, chi squared, and decision-tree models.[2,3,4,8] When applied to text, these methods are often more artificial than they are intelligent. Text features are multidimensional in terms of their informational composition.[4] In addition to various occurrence measures (such as presence and frequency), they encompass lexicology and morphology-based characteristics (including semantics and syntax). There is a need for intelligent feature-selection (IFS) methods that can exploit the syntactic properties of text features while simultaneously leveraging relevant sentiment-related semantic information.

An excellent example of a feature-selection approach tailored to sentiment analysis that utilizes the syntactic relations between text attributes is feature subsumption hierarchies (FSH).[1] Given a set of word n-grams and syntactic n-gram patterns, FSH uses the idea of performance-based feature subsumption to remove redundant or irrelevant higher order n-grams. For instance, only the word bigrams and trigrams that provide additional information (measured using some heuristic) over the unigrams they encompass are retained.[1] For example, the bigram "I like" may be subsumed by the unigram "like," but "basket case" may be retained because it contains important sentiment information not provided by "basket" or "case" alone.

Inspired by FSH, this article presents an IFS approach that incorporates syntactic and semantic information. The proposed approach helps illustrate how rich, heterogeneous feature sets, coupled with appropriate feature-selection mechanisms, can improve opinion-classification performance.

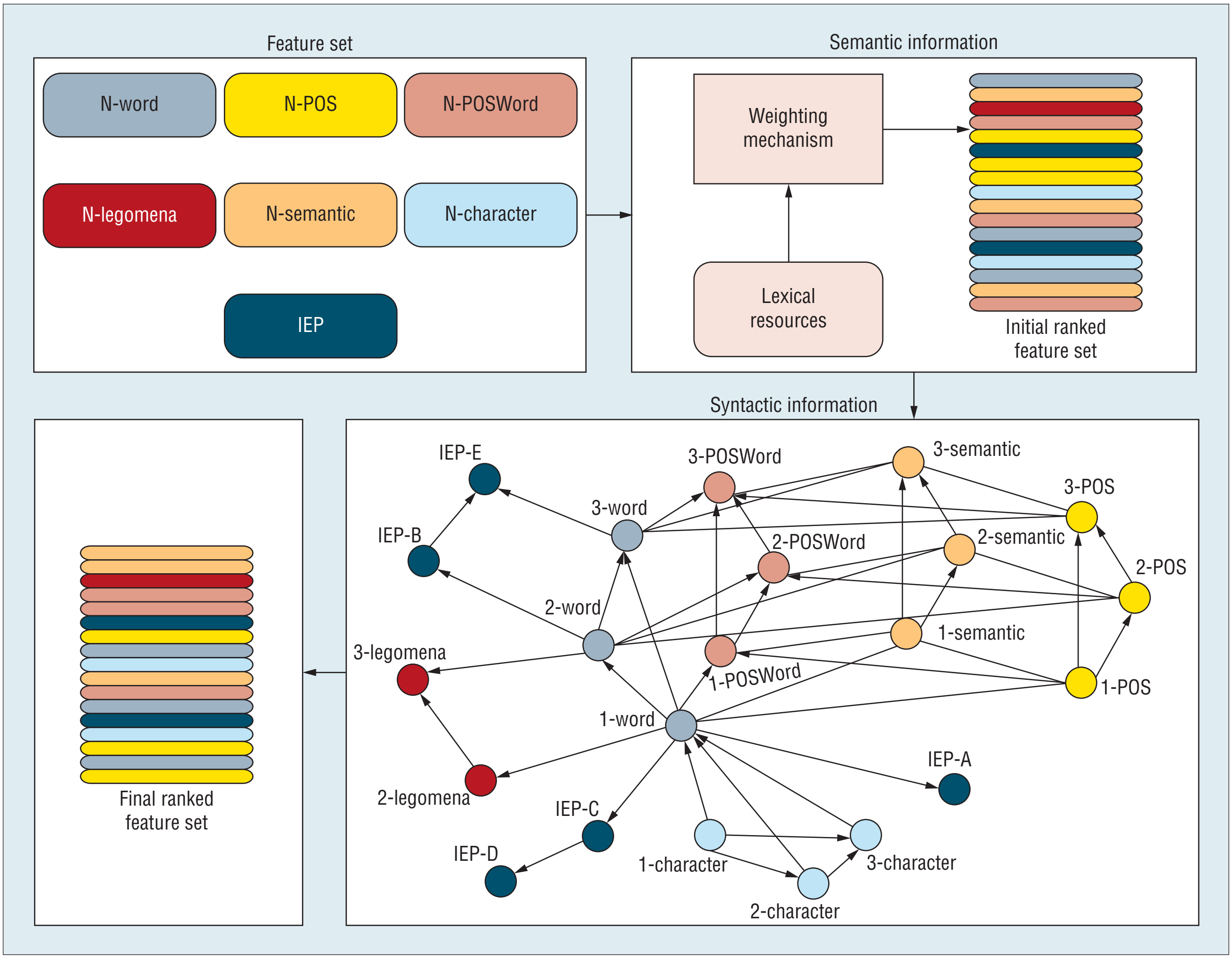

Figure 1. An intelligent feature-selection approach for opinion classification.

An IFS Approach

Figure 1 depicts the design layout for the proposed IFS approach, which uses semantic and syntactic information to refine large input feature spaces. In the example presented here, various categories of n-gram features were used. Although others could also have been incorporated, those utilized include character n-grams, word n-grams, parts-of-speech (POS) tag n-grams, word plus POS tag n-grams, legomena n-grams,[2] information extraction patterns (IEP),[1,3] and semantic patterns. For each category, I use unigrams, bigrams, and trigrams.

Semantic Information

The features' initial weights are an amalgamation of their occurrence distribution across classes in the training data as well as their degree of subjectivity, which is derived from SentiWordNet, a publicly available lexical resource.[6] Figure 2 presents the initial weighting formulation for word n-grams. Given a word n-gram feature a_x that consists of d tokens, the initial weight $w(a_x)$ is the sum of $wt(a_x)$ and $ws(a_x)$, where $ws(a_x)$ is computed by 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}. The computation of $ws(a_x)$ for other n-gram feature categories differs slightly. For instance in the case of parts-of-speech (POS) tag plus word n-grams, the word polarity values are only computed for word-sense pairs in SentiWordNet where the sense has the same POS as that of the tag associated with the word.

Syntactic Information

The IFS approach uses a feature relation network (FRN) that utilizes two important syntactic n-gram relations: subsumption and parallel relations.

In the syntactic information box in Figure 1, subsumption relations are denoted with arrows, while parallel relations are depicted using solid lines. These two relations enable intelligent comparison between features to facilitate enhanced removal of redundant and/or irrelevant attributes. Each remaining feature with a weight greater than 0 is first checked for potential subsumptions, then analyzed for parallel relations.

A subsumption relation occurs between two n-gram feature categories where one category is a more general, lower-order form of the other.[1] A subsumes B (A → B) if B is a higher order n-gram category with n-grams that contain the lower-order n-grams found in A. For example, word unigrams subsume word bigrams and trigrams, while word bigrams subsume word trigrams. Hence, given A → 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.[1] For instance, the bigrams "I love" and "love chocolate" would only be retained if their weight exceeded that of the unigram "love" by t—that is, if they provided additional information over the more general unigram. Otherwise, they would be assigned a final weight of 0.

A parallel relation occurs when two heterogeneous same-order n-gram feature groups may have some features with similar occurrences. For example, word unigrams can be associated with many POS tags, and vice versa. However, certain word and POS tags' occurrences might be highly correlated. Given two n-gram feature groups with potentially correlated attributes, A is considered to be parallel to B (A—B). If two features from categories A and B, respectively, have a correlation coefficient greater than some threshold p, one of the attributes is removed to avoid redundancy—that is, it is assigned a final weight of 0.

The weight for $a_x = (a_{x1}, \ldots a_{xd})$ 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 that 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 SentiWordNet.

Figure 2. Initial weighting mechanism for word n-grams.

Evaluation

The IFS approach was evaluated on three online product review testbeds, each consisting of 2,000 reviews: digital camera reviews from Epinions, automobile reviews from Edmunds, and movie reviews from Rotten Tomatoes. All three test beds had two classes that were balanced in terms of the number of reviews per class (1,000 each). Five-fold cross validation was used,[7,8] where feature selection was performed on the binary feature presence vectors for the 1,600 training instances during each fold. The selected features were input into a linear kernel support vector machine (SVM) classifier. The 10,000 to 100,000 features with the highest final weights were run in 2,500 feature increments. Hence, 37 feature quantities were used for all three feature sets.

IFS, as well as IFS ablations using only syntactic or semantic information, were compared against two commonly used feature selection methods: information gain and log likelihood. All five of these feature-selection methods were applied to the feature set depicted in Figure 1. Additionally, a word n-gram feature set used in conjunction with log likelihood was also included. Table 1 and Figure 3 shows the evaluation results. Table 1 depicts the area under the

Table 1. Best accuracy and area under the curve (AUC) values for different feature-selection methods across test beds.

Feature selection	Digital cameras Best accuracy (%)	Digital cameras AUC	Automobiles Best accuracy (%)	Automobiles AUC	Movies Best accuracy (%)	Movies AUC
IFS	89.2	1581	90.7	1618	89.7	1582
Semantic IFS	87.8	1566	89.7	1603	88.5	1566
Syntactic IFS	87.6	1559	89.2	1595	87.6	1560
Information gain	86.7	1549	87.8	1574	85.7	1540
Log likelihood	86.1	1540	88.2	1582	85.8	1527
Word n-gram	85.2	1519	86.0	1546	86.0	1539

curve (AUC) value as well as the best percentage accuracy across the different sized feature sets, and Figures 3a through 3c show the accuracies using the top 10,000 to 100,000 features on each of the test beds.

Using semantic and syntactic information, IFS resulted in feature sets with the best accuracy and AUC values on all three test beds. IFS outperformed information gain and log likelihood by 2 to 4 percent in terms of best accuracy and 30 to 55 points in terms of AUC, while the word n-gram feature set was surpassed by 4 to 5 percent in terms of best accuracy. These comparison feature-selection methods were outperformed by the word n-gram feature set on the movie review test bed, demonstrating how larger feature sets can be detrimental when appropriate feature-selection methods are not availed.[1] Moreover, both the semantic and syntactic information contributed to the IFS approach's overall effectiveness, as evidenced by the performance degradation that resulted when either form of information was omitted.

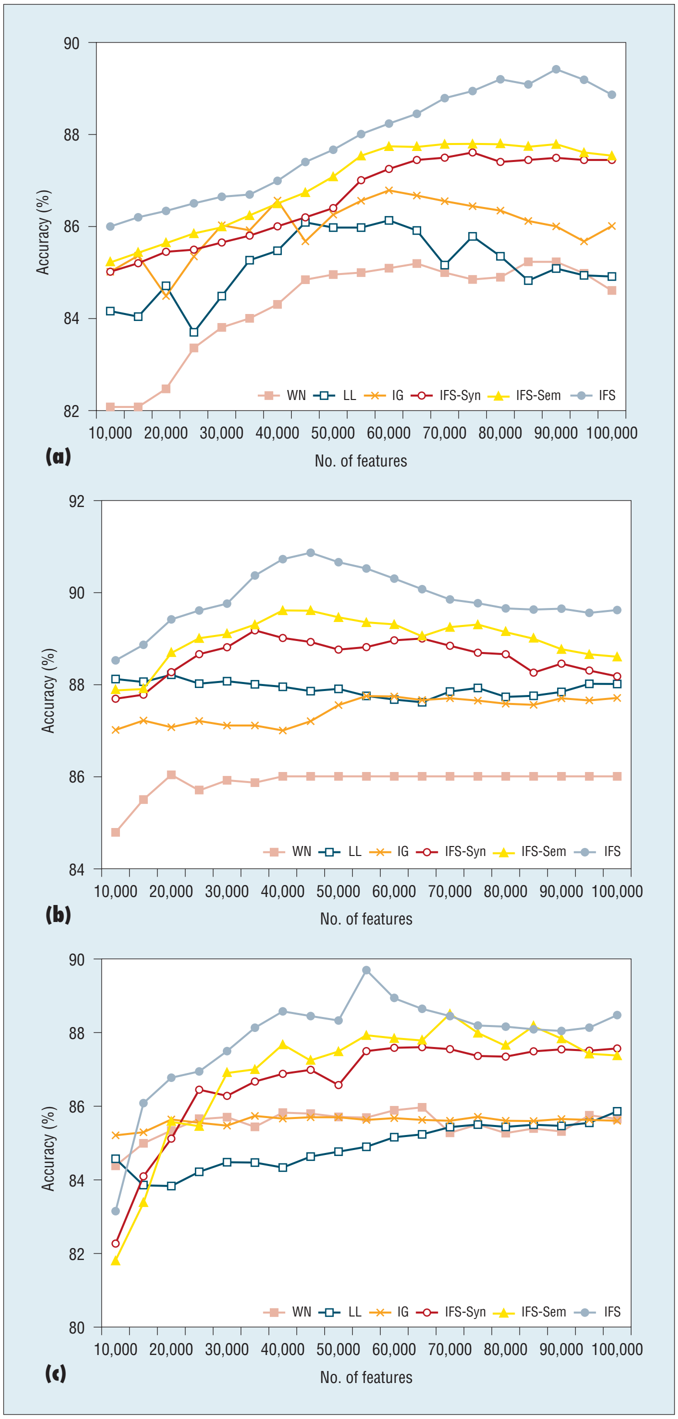

Figure 3. Evaluation results for intelligent feature selection compared to prior feature-selection methods. Online product reviews were tested for (a) digital cameras from Epinions, (b) automobiles from Edmunds, and (c) movie reviews from Rotten Tomatoes.

Future Research

This approach was intended to illustrate how IFS can be combined with larger feature sets for enhanced opinion-classification performance. There are many ways in which IFS for opinion classification can be extended in future research. Numerous additional feature categories could be used, resulting in even more robust feature sets. The syntactic and semantic information modules could be expounded on, for instance, by incorporating additional lexical resources and real-world knowledge bases.

Traditionally, sentiment-analysis research has relied on two types of feature occurrence measures (frequency and presence), while researchers have yet to methodically explore additional distributional and positional measurements. Recently, distributional measures such as compactness and first appearance have been successfully applied to topic-based text categorization.[9] These measures could be used to supplement existing occurrence measures. Hence, we could use IFS mechanisms to reduce opinion-classification feature spaces in a 2D manner: across feature categories (such as specific text features) and various occurrence measures associated with those features.

Future feature-selection efforts could explore the unique challenges associated with performing opinion classification at the document-level versus sentence-, phrase-, and word-level classification. Furthermore, there are other sentiment-analysis tasks that could benefit from improved feature selection, such as opinion holder identification and sentiment target detection. Given the plethora of potential future directions, one thing is for certain: IFS could help alleviate the quagmire associated with learning features for opinion classification, thereby allowing the kitchen sink to remain where it belongs.

References

1. E. Riloff, J. Wiebe, and T. Wilson, "Learning Subjective Nouns using Extraction Pattern Bootstrapping," *Proc. 7th Conf. Natural Language Learning*, ACM Press, 2003, pp. 25–32.
2. A. Abbasi et al., "Affect Analysis of Web Forums and Blogs using Correlation Ensembles," *IEEE Trans. Knowledge and Data Eng.*, vol. 20, no. 9, 2008, pp. 1168–1180.
3. E. Riloff, S. Patwardhan, and J. Wiebe, "Feature Subsumption for Opinion Analysis," *Proc. Conf. Empirical Methods in Natural Language Processing*, ACM Press, 2006, pp. 440–448.
4. A. Abbasi and H. Chen, "CyberGate: A Design Framework and System for Text Analysis of Computer Mediated Communication," *MIS Quarterly*, vol. 32, no. 4, 2008, pp. 811–837.
5. Y. Dang, Y. Zhang, and H. Chen, "A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews," *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 46–53.
6. A. Esuli and F. Sebastiani, "SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining," *Proc. 5th Conf. Language Resources and Evaluation* (LREC), European Assoc. Language Resources, 2006, pp. 417–422.
7. T. Mullen and N. Collier, "Sentiment Analysis Using Support Vector Machines with Diverse Information Sources," *Proc. Conf. Empirical Methods in Natural Language Processing*, ACM Press, 2004, pp. 412–418.
8. A. Abbasi et al., "Selecting Attributes for Sentiment Classification using Feature Relation Networks," to be published in *IEEE Trans. Knowledge and Data Eng.*, 2010; http://www.sba.uwm.edu/abbasi_a/index_files/IEEETKDE_FRN.pdf.
9. X.B. Xue and Z.H. Zhou, "Distributional Features for Text Categorization," *IEEE Trans. Knowledge and Data Eng.*, vol. 21, no. 3, 2009, pp. 428–444.

Ahmed Abbasi is an assistant professor of management information systems in the Sheldon B. Lubar School of Business at the University of Wisconsin-Milwaukee. He has a PhD in management information systems from the University of Arizona. Contact him at abbasi@uwm.edu.