Computational Intelligence, Volume 22, Number 2, 2006

THE IMPORTANCE OF NEUTRAL EXAMPLES FOR LEARNING SENTIMENT

MOSHE KOPPEL AND JONATHAN SCHLER

Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel

Most research on learning to identify sentiment ignores "neutral" examples, learning only from examples of significant (positive or negative) polarity. We show that it is crucial to use neutral examples in learning polarity for a variety of reasons. Learning from negative and positive examples alone will not permit accurate classification of neutral examples. Moreover, the use of neutral training examples in learning facilitates better distinction between positive and negative examples.

Key words: sentiment analysis, text categorization, machine learning.

1. INTRODUCTION

The problem of how to exploit a labeled corpus to learn classifiers for sentiment analysis has attracted a good deal of interest in recent years (Pang, Lee, and Vaithyanathan 2002; Dave, Lawrence, and Pennock 2003; Shanahan et al. 2005). One common characteristic of almost all these work has been the tendency to define the task as a two-category problem: positive versus negative. In almost all actual polarity problems, including sentiment analysis, there are, however, at least three categories that must be distinguished: positive, negative, and neutral. Not every comment on a product or experience expresses purely positive or negative sentiment. Some—in many cases, most—comments might report objective facts without expressing any sentiment, while others might express mixed or conflicting sentiment. With the single important exception of the recent work of Lee and Pang (2005), research in automated sentiment categorization has ignored such neutral documents.

Researchers are aware, of course, of the existence of neutral documents. The rationale for ignoring them has been a reliance on two tacit assumptions:

- Solving the binary positive versus negative problem automatically solves the three-category problem because neutral documents will simply lie near the boundary of the binary classifier, that is, they will be less negative than the negatives and less positive than the positives.
- There is less to learn from neutral documents than from documents with clearly defined sentiment.

The purpose of this article is to show that there is no basis for either of these myths and that neutrals can be exploited in interesting ways to great effect. The outline of the article is as follows: In Section 2, we will introduce two test corpora corresponding to different types of neutral documents. In Section 3, we will show that neutral documents do not necessarily lie close to the learned positive–negative boundary. In Section 4, we will show that using neutral training documents and standard multiclass learning methods leads to some improvement in classification accuracy but is still suboptimal. In Section 5, we will show that properly combining the respective classifiers obtained by learning from pairwise coupling of classes (i.e., positive vs. negative, positive vs. neutral, and negative vs. neutral) (Dietterich and Bakiri 1995; Hastie and Tibshirani 1998; Fuernkranz 2002) can potentially yield extremely significant improvement in overall classification accuracy.

2. VARIETIES OF NEUTRALITY

We consider two different types of labeled corpora. The first, which we will call the TV corpus, is a collection of posts to chat groups devoted to popular U.S. television shows. These posts were obtained from the Trendum Corporation, each post having been labeled by them as positive, negative, or neutral. We work with 1,974 posts equally distributed among positive, negative, and neutral documents.

The second corpus consists of 4,017 posts in shopping.com's product evaluation pages (http://www.shopping.com)[1] in the areas of digital cameras, strollers, and printers. Contributors to these pages have the option of assigning a rating of 1–5 to a product under review. We labeled reviews that assigned ratings below 3, exactly 3, and above 3 as negative, neutral, and positive, respectively. The corpus was chosen so that it consists of an equal number of positive, negative, and neutral documents.

The neutral documents that appear in the two corpora are of two fundamentally different types. The neutral television chat group posts are generally reports of upcoming or just-seen plots, scheduling announcements, or other objective information. The neutral product reviews are generally mixed reviews highlighting both positive and negative features of a given product; the difference between the two different types of neutrality must be borne in mind in exploiting this material.

3. NEUTRALITY AND BOUNDARY DISTANCE

As a baseline for later experiments, let us begin by determining the extent to which positive and negative examples in our respective corpora can be distinguished from each other using standard machine learning tools. We use as our feature set, for each corpus, the set of all words that appear at least three times in that corpus. Each document is represented as a binary vector in which each entry reflects whether the corresponding word appears in the document. We use as our learning algorithm support vector machines (SVM) with linear kernel, using Weka's implementation of SMO (Witten and Frank 2000). At this stage neutral documents are ignored entirely. For the TV corpus, the training data are classified by the learned classifier with an accuracy of 79.1% and out-of-sample documents (using fivefold cross-validation) are classified with an accuracy of 67.3%. For the shopping.com corpus, the training data are classified by the learned classifier with an accuracy of 90.1% and out-of-sample documents (using fivefold cross-validation) are classified with an accuracy of 82.7%.

Let us now entertain the hypothesis that a learned classifier that distinguishes positive from negative examples could be rigged to identify neutral examples in the following way: use the classifier to score how positive or how negative each example is and classify as neutral those test examples that are neither particularly positive nor particularly negative. Linear SVMs seem particularly appropriate for this task because they are designed to maximize the margin between positive and negative examples. Thus we might hope that neutrals would lie somewhere in the margin.

Unfortunately, this hypothesis proves to be entirely incorrect. To see quite how incorrect it is, we take the SVM classifier learned using all our positive and negative examples as training data. We measure the signed distance from the SVM boundary of each training example, as well as of each neutral example. The idea is that the further an example lies from

[1] The TV corpus is property of Trendum Corporation and has not been made publicly available. The shopping.com corpus has been made available to researchers by request. Our thanks to Amir Ashkenazi for his generosity.

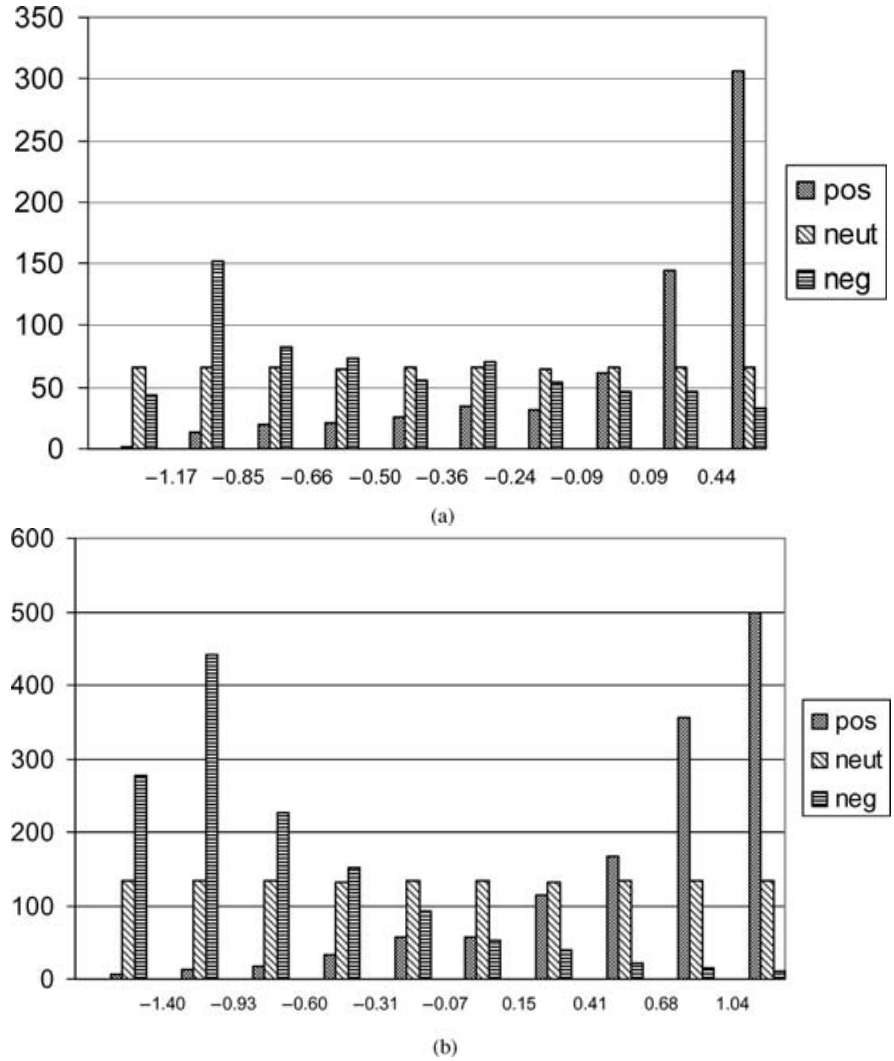

FIGURE 1. (a) Histogram indicating number of examples at various signed distances from SVM boundary in the TV corpus. (b) Histogram indicating number of examples at various signed distances from SVM boundary in the shopping.com corpus.

the boundary on the positive side the more positive it is and similarly for the negative side. In Figure 1(a), we show a histogram indicating the number of negative, neutral, and positive examples that lie at a variety of distance ranges from the SVM boundary for the TV corpus. The bins have been chosen so that each includes an equal number of neutral examples.

According to our hypothesis, one would expect to find the outer ranges dominated by positive examples (on the positive side) and negative examples (on the negative side) and the lower ranges (i.e., close to the boundary) dominated by neutral examples. As is clearly evident, however, this is not the case. Even as one gets close to the boundary, there are not significantly more neutral examples than positive examples on the positive side and not significantly more neutral examples than negative examples on the negative side. Any band around the boundary that we might choose as a "neutral band," in which examples would

be classified as neutral, would result in the misclassification of almost as many non-neutral examples as it would correctly classify as neutral. To be precise, the optimal neutral band would run from −0.55 to 0.1. In this band, there are 272 neutral examples (out of a total of 658 in the corpus), but there are also 48 positive examples and 189 negative examples that are correctly classified by the SVM. Thus an omniscient learner who was given a classifier learned from all positive and negative examples in the corpus and who divined the optimal neutral band would still only correctly classify (as positive, negative, or neutral) 54.8% of all the examples in the corpus. This is only marginally better than the 52.7% accuracy that could have been obtained by simply using the given classifier as is without classifying any examples as neutral.

In Figure 1(b) we show the results of the same experiment on the shopping.com corpus. Here too we find that the lower ranges are not dominated by neutral examples. For this corpus, the optimal neutral band would run from −0.5 to 0.45 and would include 461 neutral examples (out of a total of 1,339 in the corpus), but also 172 correctly classified positive examples and 184 correctly classified negative examples. Thus an omniscient learner who was given a classifier learned from all positive and negative examples in the corpus and who divined the optimal neutral band would correctly classify 63.0% of all examples, as opposed the 60.0% accuracy that could have been obtained by simply using the given classifier as is without classifying any examples as neutral.

All in all, there is clear evidence from both corpora that neutral documents cannot be isolated from positive and negative documents simply by using signed distance from the learned positive–negative SVM boundary.

4. LEARNING FROM NEUTRALS—PRELIMINARY ATTEMPTS

It is evident from the above that if we wish to classify documents as positive, negative, or neutral, we will need to use neutral training documents. In this section we apply six different learning algorithms (in each case, using Weka's (Witten and Frank 2000) implementation) to this problem.

- One-versus-all multiclass SVM—a generalization of SVM from two classes to many classes by learning a separate classifier for each class versus all other classes, collectively, and combining the respective outputs (Hastie and Tibshirani 1998).
- One-versus-one multiclass SVM—a generalization of SVM from two classes to many classes by learning a separate classifier for each class versus each other class, respectively, and combining the respective outputs (Hastie and Tibshirani 1998).
- J4.8, a decision tree learner.
- Naïve Bayes.
- Linear regression.
- Ordinal classification—a meta-learner that takes into account a natural order on the classes (Frank and Hall 2001); we used SVM as the base classifier.

These six algorithms were chosen for the different ways they handle the relationships between the three classes: negative, neutral, positive. J4.8 and Naïve Bayes are natural multiclass learners that treat all three classes identically. The two multiclass versions of SVM each combine three binary classifiers. Linear regression and ordinal classification both explicitly account for the fact that the classes are naturally ordered: negative < neutral < positive.

We ran fivefold cross-validation experiments using each of these methods on each of our two corpora. The results are shown in Figure 2(a) and (b). For the TV corpus,

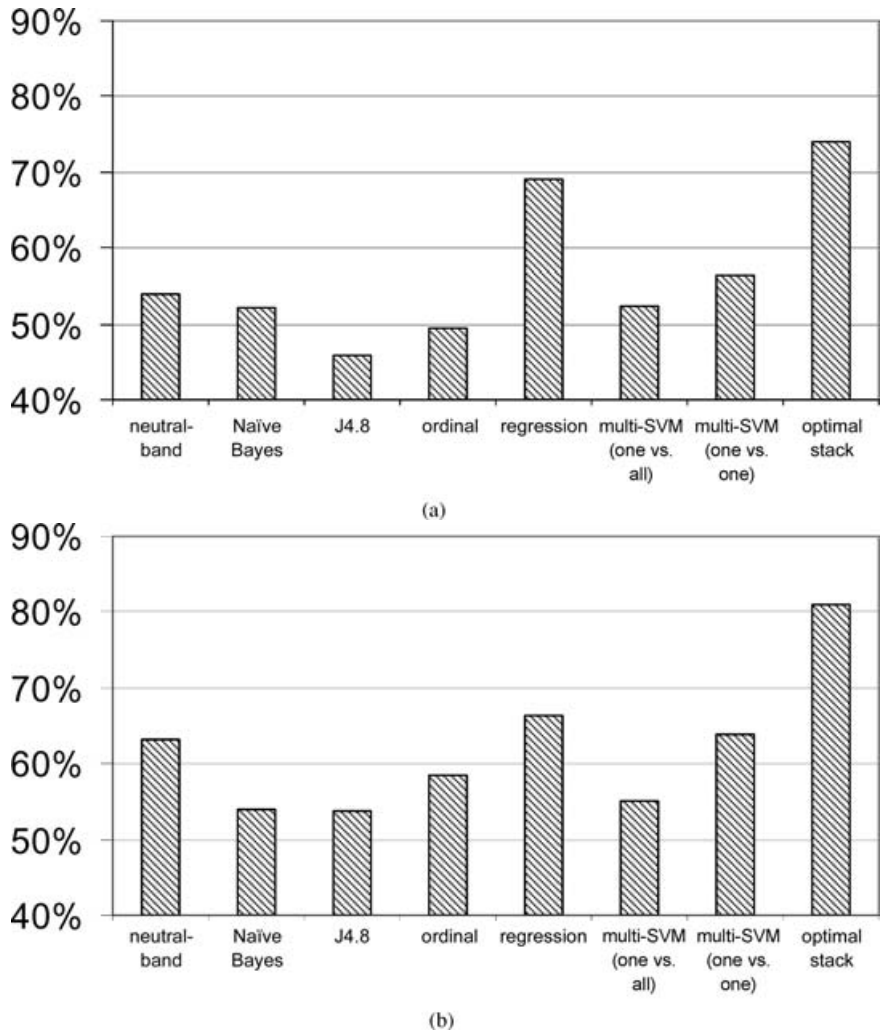

FIGURE 2. (a) Fivefold cross-validation results on the TV corpus using a variety of learning methods. Optimal neutral band is also shown for comparison. (b) Fivefold cross-validation results on the shopping.com corpus using a variety of learning methods. Optimal neutral band is also shown for comparison.

one-versus-all multiclass SVMs yields an accuracy of 52.5%, one-versus-one multiclass SVMs yields an accuracy of 56.4%, J4.8 yields an accuracy of 45.9%, Naïve Bayes yields an accuracy of 52.0%, linear regression yields an accuracy of 69.0%, and ordinal classification yields an accuracy of 49.5%. Note that results of one-versus-one multiclass SVMs and linear regression are better than even the optimal results attainable by an omniscient learner using the boundary method of the previous section. Similarly, for the shopping.com corpus, one-versus-all multiclass SVMs yields an accuracy of 55.0%, one-versus-one multiclass SVMs yields an accuracy of 63.8%, J4.8 yields an accuracy of 53.7%, Naïve Bayes yields an accuracy of 54.0%, linear regression yields an accuracy of 66.3%, and ordinal classification yields an accuracy of 58.4%. Again, results of one-versus-one multiclass SVMs and linear regression are better than that obtained using the optimal band for isolating neutrals.

It should be noted that this improvement is not attributable simply to the fact of having available more training examples. Even if we use only two-thirds of our training examples (so that the total number of training examples is the same as in the previous experiment), we obtain essentially the same results, which are better than the optimum of the boundary method. This is simply the result of the use of neutral examples.

While these results show some improvement over ignoring neutral examples, they still do not make optimal use of the neutrals.

5. OPTIMAL STACKS OF BINARY CLASSIFIERS

Let us reflect for a moment on why each of the above methods might not properly leverage the neutral examples.

In the case of regression and ordinal classification, we assume that neutrals are merely intermediate between positives and negatives. However in a series of papers, Wiebe et al. (2004) have shown that, at least at the sentence level, subjective writing, of whatever sentiment, can be distinguished from objective writing. In some sense, then, we are not leveraging those aspects of neutral examples that are distinct from positives and negatives but not intermediate.

On the other hand, the other methods all ignore the fact that negatives, neutrals, and positives do stand in some particular relationship to each other and instead treat them symmetrically. In the case of J4.8 and Naïve Bayes, this limitation is inherent. This is also true of multiclass versions of SVM, and other linear classifiers, that do not build on combinations of binary classifiers (Crammer and Singer 2001). In the case of extensions of binary learners to multiclass problems that do use combinations of binary classifiers (Dietterich and Bakiri 1995; Hastie and Tibshirani 1998; Fuernkranz 2002), we have the option of making adjustments that take into account the different information that can be gleaned from the different pairwise classifiers.

For example, we might first check if a document is neutral or not (Wiebe et al. 2004) and only if it is deemed non-neutral proceed to determine if it is positive or negative. Clearly, this approach is one of many possible asymmetric approaches.

In this section, we will see that it is crucial to take such asymmetries into account. We begin by running the following experiment. For each of the pairs, negative–positive, negative–neutral, and positive–neutral, we ran fivefold cross-validation experiments. For each example, we recorded how it was classed in the holdout set in each of the three experiments.

5.1. The TV Corpus

Table 1 shows the actual class distribution of examples in the TV corpus assigned to each of the eight possible outcomes.

As can easily be computed from the table, the accuracies of the pairwise classifiers in fivefold cross-validation trials on their respective category pairs are: positive–negative, 67.3%; positive–neutral, 73.7%; negative–neutral, 68.5%. Let us consider how we could, in principle, parlay these pairwise classifiers into the best possible three-class classifier. To do this, let us define a *stack* (Wolpert 1992) as a mapping from each of the eight possible outcomes to some class. Let an *optimal stack* be the mapping from each of the eight possible outcomes to the majority class of the examples with that outcome.

Savicky and Furnkranz (2003) have considered when such optimal stacks (determined using holdout data) might permit optimal use of pairwise coupling. They concluded that this kind of stacking is only occasionally effective. We will see that for the polarity problems we consider here, these methods can potentially be quite effective.

TABLE 1. Class Distribution of Examples per Pairwise Outcomes in TV Corpus

Positive vs. Negative	Positive vs. Neutral	Neutral vs. Negative	Original Category		
			Negative	Neutral	Positive
Negative	Neutral	Negative	**354**	52	
Negative	Neutral	Neutral	117	**154**	148
Negative	Positive	Negative		**47**	
Negative	Positive	Neutral		9	**108**
Positive	Neutral	Negative	**145**	69	
Positive	Neutral	Neutral	42	**225**	46
Positive	Positive	Negative		**90**	
Positive	Positive	Neutral		12	**356**

For a given example, let us use the shorthand Class 1 > Class 2 to mean that the learned classifier of Class 1 versus Class 2 classed the example as Class 1. The optimal stack for these data can be neatly summarized as follows:

If positive > neutral > negative then class = positive
If negative > neutral > positive then class = negative
Else class = neutral.

This simple stack yields an accuracy of 74.9% on the three-class problem, which is, somewhat surprisingly, actually better than that obtained for any of the constituent two-class problems. This illustrates that the best way to distinguish positive examples from negative ones is by leveraging the neutrals.

In fact, this stack not only leverages neutral data, it completely ignores the positive–negative classifier. Any stack that uses the positive–negative will do worse than this stack. One interesting aspect of this stack is that it deviates considerably from majority vote. For example, if both positive and negative defeat neutral, the example is classed as neutral. In this context, that makes some perverse sense: the example likely expresses some mixed sentiment. It is not classed as neutral by either learned classifier because in this corpus most neutral examples are not mixed but simply express no sentiment.

What is most astonishing about this table is the following: When, according to our classifier for positive versus neutral, a test example is classified as positive, it is not necessarily positive, but we can assert with certainty that it is not negative (despite not a single negative example being used in training). Similarly, when, according to our classifier for negative versus neutral, a test example is classified as negative, it is not necessarily negative, but we can assert with certainty that it is not positive (despite not a single positive example being used in training).

Of course, we have chosen the optimal stack post hoc. We still need to show that we can use training data to determine a stack that will work well for an out-of-sample test set. To do so, we run the following experiment. We run fivefold cross-validation in which for each fold the training data are used twice:

1. Classifiers are learned for positive–negative, positive–neutral, and negative–neutral, respectively.
2. Fivefold cross-validation is run within the training set and used to find optimal stacks as described above.

Test examples are then classified by combining the classifiers learned in step 1 according to the stack learned in step 2.

This method yields an accuracy of 74.1%, which is significantly better than the methods considered above, as illustrated in Figure 2(a).

Moreover, when used in this way, neutral examples also improve results for the problem considered by previous researchers in which all test examples are known to be either positive or negative, but not neutral. We simply adapt our method for choosing the optimal stack so that for each of the eight outcome rows, we choose the class with most examples from among positive and negative only. This method classifies positive and negative test examples (in five-fold cross-validation experiments) with an accuracy of 75.1%, which is considerably better than the accuracy of 67.3% obtained by learning SVMs directly from positive and negative training examples (as seen in Section 3 above). Moreover, this increase is not attributable to the fact that the neutral examples provide us with 50% more training examples. Even when we randomly eliminate one-third of the training examples, accuracy on the test set of positives and negatives is 74.3%. We can only conclude that we are better off with a mix of positive, negative, and neutral training examples than with only positive and negative training examples, even when our test set is known to contain only positive or negative examples.

It is interesting to speculate that it may be a general property of polarity problems that pairwise coupling ought to be done in a nonstandard way: symmetric methods such as simple majority vote may be suboptimal. We will see that an analogous principle holds in the shopping.com corpus.

5.2. The Shopping.Com Corpus

Now let us consider the same experiment for the shopping.com corpus (Table 2).

As can be computed from the table, the accuracies of the pairwise classifiers in five-fold cross-validation trials on their respective category pairs are: positive–negative, 82.7%; positive–neutral, 71.8%; negative–neutral, 71.0%. The optimal stack for this corpus yields an accuracy of 82.3% for the three-class problem.

It is evident, though, that the optimal stack in this case is entirely counterintuitive. For example, in the case where neutral > positive > negative and neutral > negative, the majority class is the highly unexpected negative. What is astonishing in this table is that we obtain a result oddly analogous to the result obtained on the TV corpus: The best indication that an

TABLE 2. Class Distribution of Examples per Pairwise Outcomes in Shopping.Com Corpus

Positive vs. Negative	Positive vs. Neutral	Neutral vs. Negative	Original Category		
			Negative	Neutral	Positive
Negative	Neutral	Negative	**1,043**	243	114
Negative	Neutral	Neutral	201	**825**	
Negative	Positive	Negative		**60**	59
Negative	Positive	Neutral		**211**	
Positive	Neutral	Negative	30		**132**
Positive	Neutral	Neutral	**65**		
Positive	Positive	Negative			
Positive	Positive	Neutral			**1,034**

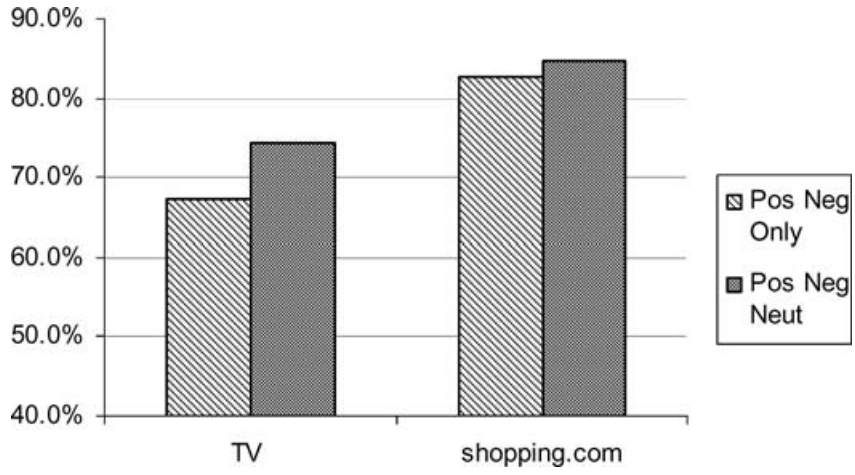

FIGURE 3. Accuracy on positive and negative test examples, using a training set consisting of positive and negative examples only versus using a training set (of equal size) consisting of positive, negative, and neutral examples.

example is not negative is that positive > neutral (this is identical to the rule above). The best indication that an example is not neutral is the fact that positive > negative. As above, in both cases, there are no exceptions.

The optimal stacks in our two corpora are indeed different from each other, a fact that no doubt reflects the differing nature of the neutral examples. However, what these optimal stacks have in common is more telling: each is asymmetric in the way it handles the different pairwise classifiers, each displays certain firm, if counterintuitive, rules, and each classifies with significantly higher accuracy than methods that treat the pairwise classifiers symmetrically or in some fixed relation.

We now run the experiment described in Section 5.1 to determine if we can learn optimal stacks on training data and then apply them successfully to test data. We find that this method yields an accuracy of 80.1%, which is far better than all the methods we considered earlier, as shown in Figure 2(b).

When this method is applied, with adjustments as described above in the discussion of the TV corpus, to test examples known to be either positive or negative, we obtain an accuracy of 85.5%. This is better than the accuracy of 82.7% obtained when training on positives and negatives only. Using only two-third of the training data, we achieve 84.6% accuracy. Thus, once again we find that a mix of training examples including neutrals is superior to a training set of the same size that consists solely of positives and negatives. The results on this experiment for both corpora are summarized in Figure 3.

6. DISCUSSION

We have seen that in learning polarity, neutral examples cannot be ignored. Learning from negative and positive examples alone will not permit accurate classification of neutral examples. Moreover, the use of neutral training examples in learning facilitates better distinction between positive and negative examples.

For the case of sentiment analysis, we find that properly combining pairwise learned classifiers leads to extremely significant improvement in overall classification accuracy. The particular method of combination that is most appropriate depends on the nature of the neutral documents in the corpus as well as other considerations. We have found that in one corpus, in which most neutral documents express no sentiment, such neutral documents can be conveniently used as a foil for testing both for negativeness or positiveness and direct positive versus negative testing can be ignored. When most neutral documents are of mixed sentiment, other stacks might be superior.

More broadly, these results suggest that polarity problems might be best handled as three-class problems using pairwise coupling but combining results in interesting ways. Although Savicky and Fuernkranz (2003) found stacking of pairwise couples to provide uneven results, it appears to be just the right approach for polarity problems. Specifically, there may often be optimal counterintuitive stacks that yield results considerably better than those achievable through voting or related multiclass methods.

REFERENCES

CRAMMER, K., and Y. SINGER. 2001. On the algorithmic implementation of multiclass SVMs. *Journal of Machine Learning Research*, **2**:265–292.

DAVE, K., S. LAWRENCE, and D. M. PENNOCK. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *In* Proceedings of the 12th International World Wide Web Conference WWW-2003, pp. 519–528, Budapest, Hungary.

DIETTERICH, T. G., and G. BAKIRI. 1995. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, **2**:263–286.

FRANK, E., and M. HALL. 2001. A simple approach to ordinal classification. *In* Proceedings of the European Conference on Machine Learning, pp. 145–165, Freiburg, Germany.

FUERNKRANZ, J. 2002. Round robin classification. *Journal of Machine Learning Research*, **2**:721–747.

HASTIE, T., and R. TIBSHIRANI. 1998. Classification by pairwise coupling. *In* Advances in Neural Information Processing Systems 10 (NIPS-97). *Edited by* M. I. Jordan, M. J. Kearns, and S. A. Solla. MIT Press, Cambridge, Massachussetts, pp. 507–513.

PANG, B., L. LEE, and S. VAITHYANATHAN. 2002. Thumbs up? Sentiment classification using machine learning techniques. *In* Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP).

PANG, B., and L. LEE. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *In* Proceedings of the ACL, pp. 115–124, Ann Arbor, MI.

SAVICKY, P., and J. FUERNKRANZ. 2003. Combining pairwise classifiers with stacking. *In* Advances in Intelligent Data Analysis, Vol. V. *Edited by* M. R. Berthold, H. J. Lenz, E. Bradley, R. Kruse, and C. H. Borgelt. Springer, Berlin, pp. 219–229.

SHANAHAN, J. G., Y. QU, and J. WIEBE. (Eds.). 2005. Computing Attitude and Affect in Text. Springer, Dordrecht, The Netherlands.

WITTEN, I. H., and E. FRANK. 2000. Data Mining: Practical Machine Learning Tools with Java Implementations. Morgan Kaufmann, San Francisco.

WIEBE, J., T. WILSON, R. BRUCE, M. BELL, and M. MARTIN. 2004. Learning subjective language. Computational Linguistics, **30**(3):277–308.

WOLPERT, D. H. 1992. Stacked generalization. *Neural Networks*, Vol. 5, pp. 241–259, Pergamon Press.