Using Neutral Examples for Learning Polarity

Moshe Koppel, Jonathan Schler
Bar-Ilan University
Computer Science Department
Ramat-Gan, 52900, ISRAEL
{ koppel ,schlerj }@cs.biu.ac.il

Abstract

Sentiment analysis is an example of polarity learning. Most research on learning to identify sentiment ignores "neutral" examples and instead performs training and testing using only examples of significant polarity. We show that it is crucial to use neutral examples in learning polarity for a variety of reasons and show how neutral examples help us obtain superior classification results in two sentiment analysis test-beds.

Many machine-learning problems involve predicting an example's *polarity*: is it (significantly) greater than or less than some standard. One canonical example of learning polarity is *sentiment analysis*, the determination of whether a particular text expresses positive or negative sentiment regarding some issue.

The problem of how to exploit a labeled corpus to learn models for sentiment analysis has attracted a good deal of interest in recent years [Dave et al 2003, Pang et al 2002, Shanahan et al 2005]. One common characteristic of almost all this 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, *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.

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 vs. negative problem automatically solves the three-category problem since neutral documents will simply lie near the boundary of the binary model
- There is less to learn from neutral documents than from documents with clearly defined sentiment

The purpose of this paper is to show that there is no basis for either of those myths and that neutrals can be exploited in interesting ways to great effect.

We consider two labeled corpora. The first consists of 1974 posts to chat groups devoted to popular U.S. television shows. The second consists of about 14,000 posts to shopping.com's product evaluation pages. Both are equally distributed among positive, negative and neutral documents.

Is it in fact the case that neutral documents lie near the boundary of a learned model that distinguishes positive and negative examples? To test this, we trained a linear SVM on all positive and negative documents in the TV corpus. In Figure 1, we show the signed distance from the boundary of the positive and negative training examples, in ascending order from left to right. (This SVM correctly classifies 79.1% of the training examples.) In addition, we show the signed distance from the boundary of all neutral examples. There is no band near the boundary in which the preponderance of examples is neutral. We indicate the band around the boundary that is optimal in terms of overall classification accuracy (positive, negative, or neutral) when all examples in the band are classed as neutral. Even using this optimal band, we attain accuracy of only 54.8%. (Note that simply using the SVM boundary to distinguish positive from negative and not classifying any examples as neutral would yield accuracy of 52.7%.)

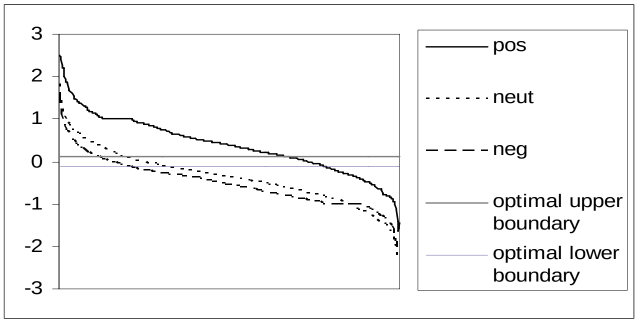

Figure 1 Distance from boundary in the TV shows corpus

Similar results are obtained for the second dataset.

What happens when we try to solve the three-class problem using positive, negative and neutral training documents? Five-fold cross-validation experiments using Weka's implementation of multi-class SVM yields accuracy of 56.5% for the TV corpus and 63.8% for the shopping.com corpus. Interestingly, we will see that these results are far inferior to results obtainable by making even stronger use of neutral documents.

Consider the algorithm used in this experiment for extending a binary algorithm to handle multiple classes, namely, pairwise coupling. In this approach, a model is learned for each pair of classes (positive-negative, positive-neutral, negative-neutral) and these models are then combined. Weka's implementation [Witten and Frank 2000] of the [Hastie and Tibshirani 1998] algorithm treats the three constituent pairwise problems identically. That is, no allowance is made for the particular relationships that positive, negative and neutral examples stand in to each other.

The main point of this paper is that it is crucial to take these special relationships into account. We begin by running the following experiment. For each of the pairs, negative-positive, negative-neutral, and positive-neutral, we ran five-fold cross-validation experiments. For each example, we recorded how it was classed in the holdout set in each of the three experiments.

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

Pos Vs Neg	**Pos Vs Neut**	**Neut Vs neg**	**original category**		
			neg	**neut**	**pos**
Neg	Neut	Neg	**354**	52	
Neg	Neut	Neut	117	**154**	148
Neg	Pos	Neg		**47**	
Neg	Pos	Neut		9	**108**
Pos	Neut	Neg	**145**	69	
Pos	Neut	Neut	42	**225**	46
Pos	Pos	Neg		**90**	
Pos	Pos	Neut		12	**356**

Table 1: Class distribution of examples per pairwise outcomes in TV corpus

As can easily be computed from the table, the accuracies of the pairwise models in five-fold cross-validation trials on their respective category pairs are: positive-negative, 67.3%; positive-neutral, 73.7%; negative-neutral, 68.5%. We want to parlay these pairwise models into the best possible three-class model. 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 2002] have considered when such optimal stacks (determined using holdout data) might permit optimal use of pairwise coupling.

For a given example, let's use the shorthand Class1 > Class2 to mean that the learned model of Class1 vs. Class2 classed the example as Class1. The optimal stack for this 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 accuracy of 74.9% on the three-class problem, which is significantly better than multi-class SVM (and better than any of the constituent two-class problems).

What is most astonishing about this table is the following: When, according to our model for positive vs. 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.) Likewise, when, according to our model for negative vs. 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.)

An analogous (though not identical) principle holds in the shopping.com corpus.

These results strongly suggest that polarity problems be attacked by stacking results of pairwise coupling in non-standard ways, taking full advantage of neutral examples.

References

[Dave, K., Lawrence, S., and Pennock, D. M., 2003] Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the Twelfth International World Wide Web Conference (WWW-2003)*

[Hastie, T. and R. Tibshirani, 1998] Classification by pairwise coupling in M. I. Jordan, M. J. Kearns, and S. A. Solla (eds.) *Advances in Neural Information Processing Systems* 10 (NIPS-97), pp. 507-513. MIT Press, 1998.

[Pang, B., Lee, L. and Vaithyanathan, S., 2002] Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*

[Savicky, P. And Fuernkranz, J., 2003] Combining Pairwise Classifiers with Stacking. in Advances in Intelligent Data Analysis V. (Ed.: Berthold M.R., Lenz H.J., Bradley E., Kruse R., Borgelt Ch.) - Berlin, Springer 2003, pp. 219-229.

[Shanahan, J. G., Yan Q., Janyce W. (Eds.), 2005] *Computing Attitude and Affect in Text*, Springer, Dordrecht, The Netherlands, 2005

[Witten I. H. and Frank E. 2000] "*Data Mining: Practical machine learning tools with Java implementations*" Morgan Kaufmann, San Francisco, 2000

[Wolpert, D.H., 1992], Stacked Generalization, *Neural Networks*, Vol. 5, pp. 241-259, Pergamon Press