Thumbs up? Sentiment Classification using Machine Learning Techniques

By

Bo Pang,

Lillian Lee,

And Shivakumar Vaithyanathan

Presented by

Tian Tang

Introduction

- Goal: automatic text categorization and organization
- Traditional way: topic-based categorization
- Our attempt: sentiment-based categorization using supervised learning

Why sentiment? (what are the benefits?)

Introduction

- Why sentiment?

  providing succinct summaries for readers (*movie reviews*)

  helping business intelligence applications and recommender systems

  potential applications on message filtering (*recognize and discard “flames”*)

Introduction

- The difficulty lying in sentiment classification: *more subtle way to express, requiring more understanding (difficult!)*

  "How could anyone sit through this movie?" contains no single word that is obviously negative. But people know.

- Comparing to topic-based classification*: often identifiable via keywords alone*

Previous Work

- *Source* or *source style*: author, publisher, native-language background, and “brow”
- *Genre* of text: subjective (“editorial”), objective. (It doesn’t tell us what the opinion is.)
- *Knowledge-based*: semantic orientation
- *Unsupervised learning*: mutual information/ scores computed using statistics

Recent Related Works

- Choi and Cardie (2008) proposed a method to classify the sentiment polarity of a sentence basing on compositional semantics. In their method, the polarity of the whole sentence is determined from the prior polarities of the composing words by pre-defined rule (*Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis*)

Recent Related Works

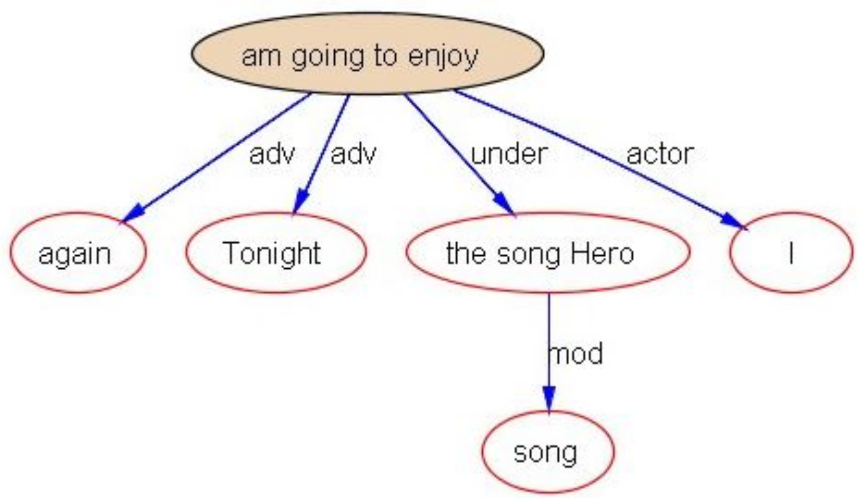
- Syntactic structures were used in the studies of Moilanen and Pulman (2007), but their methods are based on rules and supervised learning was not used to handle polarity reversal. (*Sentiment Composition*)

Recent Related Works

- Wilson et al. (2005) studied a bag-of-features based statistical sentiment classification method incorporating head-modifier relation. (
Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis)

My Experience

- In NetBase Inc., we use pattern matching and dependency tree to capture sentiments and objects.



Tonight, I am going to enjoy the song Hero again.

Experimental Environment

- Data Source: movie reviews from IMDB archive (*rec.arts.movies.reviews*)
- Data Format: with stars or numerical value
- Categories: positive, negative and neutral
- Limitation policy: fewer than 20 reviews form per author per sentiment category allowed
- Data set: 752 negative and 1301 positive reviews from 144 reviewers. (*http://www.cs.cornell.edu/people/pabo/-movie-review-data*)

Experiments

- Random-choice baseline: 50% accuracy
- Human word lists: 700 positive and 700 negative reviews

	Proposed word lists	Accuracy	Ties
Human 1	positive: *dazzling, brilliant, phenomenal, excellent, fantastic* negative: *suck, terrible, awful, unwatchable, hideous*	58%	75%
Human 2	positive: *gripping, mesmerizing, riveting, spectacular, cool, awesome, thrilling, badass, excellent, moving, exciting* negative: *bad, cliched, sucks, boring, stupid, slow*	64%	39%

- Plus introspection and statistics of data:

	Proposed word lists	Accuracy	Ties
Human 3 + stats	positive: *love, wonderful, best, great, superb, still, beautiful* negative: *bad, worst, stupid, waste, boring, ?, !*	69%	16%

Experiments

- Bag-of-features:

  Let *{f1, . . . , fm} be* a predefined set of *m features that can appear in* a document; (examples include the word "still" or the bigram "really stinks".) Let *ni(d) be the number* of times *fi occurs in document d. Then, each* document *d is represented by the* *docume* $\vec{d} := (n_1(d), n_2(d), \ldots, n_m(d))$

Experiments

Machine Learning methods:

- **Naïve Bayes (NB)**: $P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)}$

  where *P(d) plays no role in selecting c*. To estimate* the term *P(d | c), Naive Bayes decomposes it by assuming* the *fi's are conditionally independent given d's class:*

$$P_{\mathrm{NB}}(c \mid d) := \frac{P(c)\left(\prod_{i=1}^{m} P(f_i \mid c)^{n_i(d)}\right)}{P(d)}.$$

Experiments

- **Maximum Entropy** (**MaxEnt**, or **ME**, for

$$P_{\mathrm{ME}}(c \mid d) := \frac{1}{Z(d)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(d,c)\right)$$

where *Z(d) is a normalization function. Fi,c is a feature/class function for feature fi and class c, defined as follows:*

$$F_{i,c}(d,c') := \begin{cases} 1, & n_i(d) > 0 \text{ and } c' = c \\ 0 & \text{otherwise} \end{cases}$$

Experiments

- **Support Vector Machines** (**SVMs** for short)**:**

  $$\vec{w} := \sum_j \alpha_j c_j \vec{d}_j, \quad \alpha_j \geq 0$$

  where the *αj 's are obtained by solving a dual optimization* problem. Those vector *dj such that αj is greater* than zero are called *support vectors, since they are* the only document vectors contributing to *vector w. Classification* of test instances consists simply of determining which side of *vector w's hyperplane they fall on.*

Evaluation

Experimental set-up:

- randomly selected 700 positive-sentiment and 700 negative-sentiment documents
- Divided into three equal-sized folds, maintaining balanced class distributions in each fold
- Attempt to model the potentially important contextual effect of negation
- Focusing on features based on unigrams (with negation tagging) and bigrams

Evaluation

Results:

	Features	# of features	frequency or presence?	NB	ME	SVM
(1)	unigrams	16165	freq.	**78.7**	N/A	72.8
(2)	unigrams	"	pres.	81.0	80.4	**82.9**
(3)	unigrams+bigrams	32330	pres.	80.6	80.8	**82.7**
(4)	bigrams	16165	pres.	77.3	**77.4**	77.1
(5)	unigrams+POS	16695	pres.	81.5	80.4	**81.9**
(6)	adjectives	2633	pres.	77.0	**77.7**	75.1
(7)	top 2633 unigrams	2633	pres.	80.3	81.0	**81.4**
(8)	unigrams+position	22430	pres.	81.0	80.1	**81.6**

Figure 3: Average three-fold cross-validation accuracies, in percent. Boldface: best performance for a given setting (row). Recall that our baseline results ranged from 50% to 69%.

Evaluation

- **Initial unigram results**:

  surpassing the random-choice baseline of 50%

  beating our two human-selected-unigram baselines of 58% and 64%;

  performing well in comparison to the 69% baseline achieved via limited access to the test-data statistics

  but…

Evaluation

- but in **topic-based classification**, all three classifiers have been reported to use bag-of-unigram features to achieve accuracies of 90% and above for particular categories (Joachims, 1998; Nigam et al., 1999) — and such results are for settings with more than two classes (suggesting sentiment categorization is more difficult than topic classification).

Evaluation

- **Feature frequency vs. presence**:

  representing each document *d by a feature-count vector* (*n1(d), . . . , nm(d)).*

  binarizing the document vectors, setting *ni(d) to 1 if and only feature fi appears in d, and* re-running (in order to investigate whether reliance on frequency information could account for the higher accuracies of Naive Bayes and SVMs)

Evaluation

- Results: better performance achieved by accounting only for feature presence, not feature frequency. (in direct opposition to the observations of McCallum and Nigam (1998) with respect to Naïve Bayes topic classification, which may suggests a difference between sentiment and topic categorization)

Evaluation

- **Bigrams**: to capture more context in general, but not conditional independent from unigrams (not imply that Naïve Bayes will necessarily do poorly)
- Results: not improving performance beyond that of unigram presence (at least in this setting)

Evaluation

- **Parts of speech (POS)**: serving as a crude form of word sense disambiguation ("I love this movie" and "a love story")
- Results: using adjectives alone shows a very poor performance (against to intuitive expectation); applying explicit feature-selection algorithms on unigrams could improve performance

Evaluation

- **Position**: tagging each word according to the position of its appearance (first quarter, last quarter, or middle half of the document)
- Results: not differing greatly from using unigrams alone

Discussions

- Best: SVMs Worst: Naïve Bayes
- The differences are not very large
- Not comparable to topic-based classification
- Future work:

  Figuring out what accounts for differences between topic and sentiment to improve performance of sentiment classification;

  Identification of features indicating whether sentences are on-topic

Thanks for watching!

Tian