Thumbs up? Sentiment Classification using Machine Learning Technique

Bo Pang, Lillian lee and Shivakumar Vaithyanathan
EMNLP 2002

Presented by
Lu Wang

Sentiment Classification

- Classify documents (e.g., reviews) based on the overall sentiments expressed by authors
- Similar but different from topic-based text classification
  - In topic-based text classification, topic words are important.
  - In sentiment classification, sentiment words are more important, e.g., great, excellent, horrible, bad, worst, etc.

Baseline

- Assumption
  - There are certain words people tend to use to express strong sentiments, so that it might suffice to simply produce a list of such words by introspection and rely on them alone to classify the texts
  - They did classification by counting the number of positive and negative words in the documents according to the word lists

Baseline

- Random-choice baseline: 50%

	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%

Figure 1: Baseline results for human word lists. Data: 700 positive and 700 negative reviews.

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

Figure 2: Results for baseline using introspection and simple statistics of the data (including *test* data).

Problem description

- Aim in this work:
  - Whether it suffices to treat sentiment classification simply as a special case of topic-based categorization
  - Whether special sentiment-categorization methods need to be developed

Experimental Set-up

- Domain
  - movie review
- Data Source
  - Internet Movie Database(IMDb)
- Preprocess
  - Ratings were automatically extracted and converted into one of three categories: positive, negative or neutral
- Data set:
  - Randomly selected 700 positive-sentiment and 700 negative-sentiment documents
  - Then, they divided this data into three equal-sized folds, maintaining balanced class distributions in each fold

Machine Learning Methods

- Standard bag-of-features framework
- Let $\{f_1,...,f_m\}$ be a predefined set of m features that can appear in a document
- Let $n_i(d)$ be the number of times f_i occurs in document d.
- Then, each document d is represented by the document vector $\vec{d} := (n_1(d), n_2(d), ..., n_m(d))$

Naive Bayes

- Assign to a given document d the class
  - $c^* = \arg\max_c P(c \mid d)$
  - $P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)}$
  - $P_{NB}(c \mid d) := \frac{P(c)(\prod_{i=1}^{m} P(f_i \mid c)^{n_i(d)})}{P(d)}$
- Conditional independence assumption between the features

Maximum Entropy

- $P_{ME} := \frac{1}{Z(d)} \exp(\sum_i \lambda_{i,c} F_{i,c}(d,c))$
- $Z(d)$ is a normalization function
- $F_{i,c}$ is a feature/class function for feature f_i and class c, defined as follows:

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

- $\lambda_{i,c}$ are feature-weight parameters
- MaxEnt makes no assumption about the relationships between features

Support Vector Machines

- Highly effective at traditional text categorization
- Find a hyperplane $\vec{w}$ not only separates the document vectors in one class from those in the other, but for which the separation, or margin, is as large as possible
- $\vec{w} := \sum_j \alpha_j c_j \vec{d}_j, \alpha_j \geq 0$
  - $c_j \in \{-1, 1\}$
  - α_j is obtained by solving a dual optimization problem

Conclusion

- Naive Bayes tends to do the worst and SVMs tend to do the best
- Unigram presence information is the most effective
- "thwarted expectation", many words indicative of the opposite sentiment to that of the entire review
  - "This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up"
- Some form of discourse analysis is necessary

Evaluation

- The tag "NOT_" was added to every word between a negation word ("not", "isn't", "didn't") and the first punctuation mark following the negation word
  - Don'tlike →NOT_like
- Feature
  - Focused or features based on unigrams and bigrams
  - 16,165 unigrams appearing at least 4 times in the 1400-document corpus
  - 16,165 most often occurring bigrams in the same data(they didn't add negation tags to the bigrams)

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%.