UCSC on TREC 2006 Blog Opinion Mining

Ethan Zhang and Yi Zhang
University of California Santa Cruz
{ethanz,yiz}@soe.ucsc.edu

1 Introduction

The University of California Santa Cruz team submitted three runs for the TREC Blog Track opinion mining task. We developed a two stage retrieval system. We started with retrieving relevant documents from the corpus for each topic, and then ran each retrieved document through a classifier to estimate the probability that the document contains opinion expressions. The documents were ranked according to the product of the retrieval score and the estimated probability. The Lemur search engine, which is based on the language modeling approach, was used for retrieval. A Bayesian Logistic Regression classifier was trained using a noisy training data set from other domains, which include news articles, product reviews and movie reviews. All runs are automatic.

2 Document Retrieval

The Lemur Toolkit [1] was used to index and retrieve blog documents. The search engine was implemented based on language modeling. The retrieval model used by the software is best documented in http://ciir.cs.umass.edu/ metzler/indriretmodel.html.

HTML tags were stripped while the documents were indexed. Porter stemmer was applied to both index documents and search queries. A stop word list containing 587 common terms was used.

3 Logistic Regression Classifier

Bayesian logistic regression model was used to learn a classifier. In this model, we have:

$$P(y = 1|\beta, \mathbf{x}) = \frac{1}{1 + e^{-\beta^T \mathbf{x}}}$$

where $\mathbf{x}$ represents the input vector including the bias $x_0 = 1$, and β is the parameter vector we are to learn. The output y is computed as follows,

$$y = +1 \text{ if } P(y = 1|\beta, \mathbf{x}) > \ 0.5 \text{ and -1 otherwise.}$$

In our blog opinion mining runs, the input $\mathbf{x}$ is the vector representation of a blog post, in which each component is a binary number indicating whether a term appears in the document. In other words, if we let $T = \{t_1, t_2, ..., t_p\}$ denote the feature set, then $x_i = 1$ if t_i is present in the document and $x_i = 0$ if t_i is absent from the document. The output $y = +1$ means the input expresses opinions, and $y = -1$ means the input does not express opinions.

Let the prior distribution of the model parameter β follow a Gaussian distribution $N(\beta; \mathbf{m}_\beta, V_\beta))$. The posterior probability distribution of model parameter β conditional on the training data D can be calculated based on Bayesian theorem.

$$\begin{aligned} P(\beta|D) &= \frac{P(D|\beta)P(\beta)}{\int_\beta P(D|\beta)P(\beta)} \\ &= \frac{\prod_{i=1}^t P(y_i|\beta, \mathbf{x_i})P(\beta)}{\int_\beta \prod_i P(y_i|\beta, \mathbf{x_i})P(\beta)} \end{aligned}$$

Let V_β be a diagonal matrix with each entry on the diagonal equals v_β. Thus the max a posterior estimation (MAP) of β is:

$$\begin{aligned} \beta_{MAP} &= argmax_\beta P(\beta|D_t) \\ &= argmax_\beta \prod_{i=1}^t P(y_i|\beta, \mathbf{x_i})P(\beta) \\ &= argmax_\beta \sum_{i=1}^t log(1 + \exp(-y_i \beta^T \mathbf{x_i})) \\ & \quad -v_\beta(\beta - \mathbf{m}_\beta)^2 \end{aligned}$$

In the experiments, we arbitrarily set $m_\beta = (0, 0, ..., 0)$ and $v_\beta = 1$. Conjugate gradient descent method was used to find β_{MAP}.

We used the product of retrieval score and classification probability as the combined score to order the results. The same classifier was used for all our runs.

4 Training Data For Opinion Classification

Since there are no publicly available labeled blog documents for opinion mining, we looked for training data in other domains. We needed a mixture of documents expressing opinions and texts that are objective. Our training set contained 82,437 documents, which were obtained from three sources:

- **Yahoo Movies reviews**: We downloaded 15,983 user reviews from http://movies.yahoo.com. The reviews were posted for 15 different movies.
- **Epinions Digital Camera reviews**: We used 6,454 reviews from Epinions.com on digital cameras.
- **Reuters newswire**: 60,000 Reuters finance news stories were used in the training set.

Documents in the Yahoo Movies reviews and Epinions Digital Camera reviews sets were automatically labeled as "opinion" and Reuters news documents were labeled "non-opinion". It is worth mentioning that this is a very noisy training data set, since people do express opinions in Reuters news and the characteristics of this data set are very different from the TREC blog data.

Our goal is to learn a domain and topic independent opinion classifier. However, a classifier trained in a particular domain will overfit the particular domain. For example, if only Digital Camera reviews and news stories were used as training data, terms such as "focus" and "lens" would get very heavy weights because in this set they appear frequently in opinion text but rarely in non-opinion documents. We used two sets of review documents from different domains so that the classifier learned won't overfit a particular product category too much. Ideally, we would like to gather review documents from as many domains as possible.

5 Feature Engineering

We used 6,511 features that contain 5,604 adjectives chosen from the SentiWordNet lexicon [2] and 907 bigrams and trigrams that were automatically selected from the training data.

5.1 SentiWordNet

SentiWordNet is a lexical resource in which each WordNet [3] synset s is assigned three scores: $Obj(s)$, $Pos(s)$ and $Neg(s)$, indicating how objective, positive and negative the terms contained in the synset are. The scores add up to 1:

$$Obj(s) + Pos(s) + Neg(s) = 1.$$

$Pos(s) + Neg(s)$ can be regarded as the subjectivity score of the terms.

We used in our feature set the subjective adjectives extracted from SentiWordNet 1.0. An adjective is considered subjective if any synset that contains it has a subjectivity score above 0.5. Table 1 shows some examples of such adjectives.

	Pos	Neg	Obj
happy	0.875	0	0.125
disastrous	0.0	0.75	0.25
poor	0.0	0.75	0.25
wrong	0.0	0.875	0.125
specific	0.375	0.25	0.375
counterfeit	0.375	0.5	0.125
sturdy	0.5	0.25	0.25
comparable	0.625	0.0	0.375

Table 1: Examples subjective adjectives. The scores belong to the most subjective WordNet synset that contains the adjective.

5.2 Bigrams and Trigrams

We extracted from the Yahoo Movies and Epinions Digital Camera reviews bigrams and trigrams that appear in more than four documents. This left with 43,195 terms to form the feature space. Vectoral representations of all documents in the training set were then used to select the most relevant features. We used the Pearson's correlation coefficient method to select 907 features. Examples of high-scoring bigrams and trigrams are shown in Table 2.

reason i	the settings
in my opinion	settings and
the only thing	this one is
a must	features are
recommend this	great and
i purchased	special effects
i recently	it if you
my favorite	does have
well worth	you haven’t
over and over	funny and

Table 2: Most relevant bigram and trigram features

6 Preliminary Experimental Results

We submitted three automatic runs. In each run, we retrieved documents using the Lemur search engine and ran each retrieved document through a Logistic Regression classifer. The classifier, which was trained once using the document set described in Section 4, was reused for all three runs. The difference between the runs is in the retrieval of documents and is described as follows.

- UCSCAUTO: The title field of the topics was used as the search query. A numeric score is associated with each retrieved document. The classifier then predicts the probability of containing opinion expressions for each document. The product of the retrieval score and the predicted probability was used to order the results.
- UCSCDESC: The topic description was used as the search query to retrieve documents. Words in the description that are not specific to any topic were treated as stop words and removed from the expanded query. Some examples of such words are “provide”, “find”, “regarding”, and “opinion”. The results were then classified the same way as in UCSCAUTO.
- UCSCRELFB: The ordered results of UCSCAUTO were used as pseudo relevance feedback for the search engine. We ran search using the topic title as the query with the pseudo relevance feedback. The documents were ordered according to the retrieval score. No additional classification was performed on the results.

Table 3 shows the average R-precisions over all topics for opinion retrieval and topic relevance. UCSCAUTO performs the best among the three runs. This observation is very different from those reported by some past TREC ad hoc retrieval participants, who observed that descriptions can be helpful. This suggest that blog opinion mining queries are very different from traditional TREC queries, and retrieval techniques that used to work well on old TREC ad hoc retrieval data sets may not work well on blog retrieval task and vice versa.

	Opinion Retrieval	Topic Relevance
UCSCAUTO	0.2354	0.3047
UCSCDESC	0.2198	0.2806
UCSCRELFB	0.2076	0.2992

Table 3: Average R-precisions over all topics for opinion retrieval and topic relevance.

In our two stage ranking system, the opinion classifier works separately from the topic retrieval. To tell how well the classifier learned from other domains works in the blog domain, we did some further experiments to run classifier alone on relevant blog documents about at least one target. The UCSCAUTO run produced 11,677 relevant documents that received a qrel score of 1 and above. We ran the classifier on these documents and the confusion matrix is presented in Table 4. A document is considered true opinion if it received a qrel score of 2 or above. The overall prediction accuracy on the test set is 61%.

References

[1] The Lemur Toolkit for Language Modeling and Information Retrieval, http://www.lemurproject.org/

[2] A. Esuli and F. Sebastiani, “SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining”, In Proceedings of LREC 2006 - 5th Conference on Language Resources and Evaluation

[3] WordNet, http://wordnet.princeton.edu/

	Predicted Yes	Predicted No
Opinion	40%	20%
Non-opinion	19%	21%

Table 4: Test data prediction confusion matrix. The overall accuracy is 61%.