An exploration of sentiment summarization

Philip Beineke and **Trevor Hastie**
Dept. of Statistics
Stanford University
Stanford, CA 94305

Christopher Manning
Dept. of Computer Science
Stanford University
Stanford CA 94305-9040

Shivakumar Vaithyanathan
IBM Almaden Research Center
650 Harry Rd.
San Jose, CA 95120-6099

Abstract

We introduce the idea of a sentiment summary, a single passage from a document that captures an author's opinion about his or her subject. Using supervised data from the Rotten Tomatoes website, we examine features that appear to be helpful in locating a good summary sentence. These features are used to fit Naive Bayes and regularized logistic regression models for summary extraction.

Introduction

The website Rotten Tomatoes, located at `www.rottentomatoes.com`, is primarily an online repository of movie reviews. For each movie review document, the site provides a link to the full review, along with a brief description of its sentiment. The description consists of a rating ("fresh" or "rotten") and a short quotation from the review. Other research (Pang, Lee, & Vaithyanathan 2002) has predicted a movie review's rating from its text. In this paper, we focus on the quotation, which is a main attraction to site users.

A Rotten Tomatoes quotation is typically about one sentence in length and expresses concisely the reviewer's opinion of the movie. To illustrate, Curtis Edmonds's review of the documentary *Spellbound* is encapsulated, "Hitchcock couldn't have asked for a more suspenseful situation." A.O. Scott's review of *Once upon a Time in Mexico* is encapsulated, "A noisy, unholy mess, with moments of wit and surprise that ultimately make its brutal tedium all the more disappointing." A reader can infer from these statements whether or not the overall sentiment is favorable, and get an impression about why. Consequently, we refer to them as *sentiment summaries.*

Apart from movie reviews, it is easy to envision other situations where obtaining such quotations would be useful. A manufacturer may wish to see sentiment summaries for its product reviews; a policy-maker may wish to see them for newspaper editorials; a university lecturer may wish to see them for his or her student feedback. In order to produce sentiment summaries in these situations, we would like to have a process that is automated. As a step towards this goal, we examine the summary quotations and fit statistical models that mimic the process by which they are extracted.

Rotten Tomatoes Data

Over 2,500 critics are listed by Rotten Tomatoes. Some write independently, others for a wide range of publications. As a consequence, reviews vary considerably in format, length, and writing style. There are even non-English language reviews included. In order to obtain a reasonably homogeneous collection of full-length reviews, we restrict our attention to the source publications that Rotten Tomatoes terms "The Cream of the Crop." For various reasons,[1] some of these publications were excluded. Thus our final list of sources is restricted to 14 of them.[2] Combined, there are reviews from over 200 critics.

Several thousand (3897) full-text reviews were downloaded and extracted from the web pages on which they reside. Most HTML formatting was removed, although a few features (e.g. paragraph breaks, italics) were retained for modeling and treated identically to words. The text was then tokenized. All words were shifted to lower-case, and passed through a Porter Stemmer (Porter 1980). In addition, some precautions were taken to ensure that different writing conventions produce the same output. For instance, different methods of writing ". . ." were pooled together.

From there, the Unix command *diff* was used to identify matching substrings between review quotations and their corresponding full text. When the quotation and full text review matched with at most three alterations (inserted strings, deleted strings, or type mis-matches), the capsule

[1] There are many possible reasons that a publication was removed from consideration. These include: few of its reviews have valid links or accompanying quotations; its reviews exist only in audio or video format; its reviews are only a single paragraph in length; its reviews discuss multiple movies within a single document; it requires user log-in; or, its reviews reside on web pages whose format makes it cumbersome to separate the review text from other material.

[2] Included publications are The Arizona Republic, The Atlanta Journal-Constitution, The Boston Globe, The Chicago Reader, The Chicago Sun-Times, CNN, The Detroit Free Press, the Detroit News, Entertainment Weekly, the Hollywood Reporter, the Minneapolis Star-Tribune, the New York Post, the San Francisco Chronicle, the Philadelphia Inquirer, and USA Today.

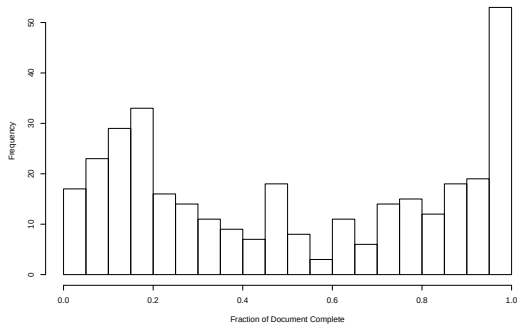

Figure 1: Quotation location within document

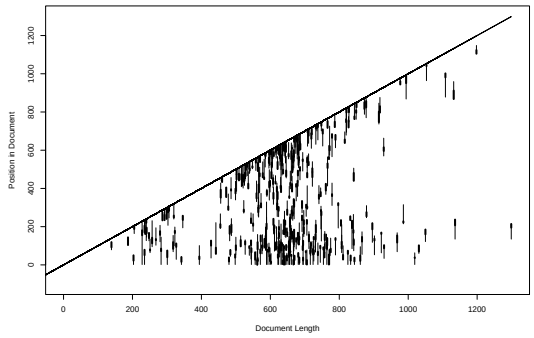

Figure 2: Quotation location versus document length

was deemed to have been found. If a quotation was shorter than five tokens in length, it was required to match the full text precisely. To avoid artifacts of editing, the Rotten Tomatoes quotations are not used for prediction. Rather, we use the original sentences from which they are drawn.

Descriptive Results

Based on exploratory data analysis, several features appear to be predictive of whether a particular span of text will be chosen as a quotation. These include the following.

1. **Location within Paragraph**

   Quotations occur most often at the ends of paragraphs – 47.6% begin at the start of a paragraph, while 26.1% conclude at the end.

2. **Location within Document**

   Figure 1 shows the location of the midpoints of summary quotations within documents. There are two modes in this plot: one is early in the document, while the other is in the final five percent of the text.

   Figure 2 elaborates upon this information, using a sample of 300 documents. In the plot, each vertical line identifies the location of a paragraph that contains a summary quotation. The thicker portion of each line identifies the location of the quotation itself.

Stemmed Word	In Quot.	Elsewhere	Pct. in Quot.
well-made	10	8	55.6%
craft	34	91	27.2%
mildly	17	46	27.0%
to dazzle	23	77	23.0%
to entertain	144	577	20.0%
nevertheless	19	110	16.6%
movie	915	6881	11.7%
film	797	6990	10.2%
to be	719	7963	8.3 %

Figure 3: Words with higher-than-average frequency in quotations

3. **Word Choice**

   In the entire corpus, 6.7% of tokens occur within summary sentences. Given that dictionary element $d_k \in \mathcal{D}$ occurs n_k times in the corpus, we can compare the number of times it occurs in a quotation with a binomial random variable whose parameters are $n = n_k$ and $p = 0.067$. Of the 3,537 types that appear between 50 and 500 times in the corpus, 348 of their counts (9.8%) are above the 99-th percentile of the corresponding binomial distribution. By chance, we would only have expected 35 counts to pass this threshold. This suggests that many types are useful in distinguishing between quotations and other text.

   The words that appear more frequently in quotations often express emotion directly. Words that are interchangeable with "movie" are also more common, as are several other words with varied meanings. Some examples are listed in Figure 3. In addition to words, formatting is a useful predictor. Italicized words and phrases (such as titles) make 8.9 % (893 of 10152) of their appearances in quotations, while parentheses make only 2.9% (569 of 18375).

Statistical Models

We approach sentiment summarization as a classification problem at the sentence level. A review text t_i is viewed as a collection of sentences, $(s_{i1}, \ldots, s_{im(t_i)})$. In training data, each sentence s_{ij} is associated with a label y_{ij}.

$$y_{ij} = \begin{cases} 1 & \text{if } s_{ij} \text{ is the summary} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

For each sentence s_{ij}, we also have two vectors of of features: type features $\mathbf{n}_{ij}$ and location features $\mathbf{l}_{ij}$.

Given a dictionary $\mathcal{D}$, the feature n_{ijk} is the number of times that type $d_k \in \mathcal{D}$ occurs in sentence s_{ij}. Because most types occur only infrequently in quotations, we restrict attention to the 1000 most frequent.

The vector $\mathbf{l}_{ij}$ consists of binary variables that indicate where in a document a sentence occurs. For example,

$$l_{ij1} = \begin{cases} 1 & \text{if } s_{ij} \text{ is in the first paragraph} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Other location variables are used to indicate whether a sentence occurs in the final paragraph, whether it is the first sentence in a paragraph, and whether it is the last sentence in its paragraph.

Using these features, we fit statistical models to estimate

$$\widehat{\Pr}(y_{ij} = 1|\mathbf{l}_{ij}, \mathbf{n}_{ij}) \quad (3)$$

Our chosen summary sentence for document t_i is the one that maximizes the above quantity. We fit these models by two different methods: Naive Bayes and regularized logistic regression.

Naive Bayes

The multinomial Naive Bayes model on a dictionary D is a familiar option for text classification, e.g. (Gale, Church, & Yarowski 1992), (McCallum & Nigam 1998). When there are additional features, the Naive Bayes model has also a natural extension: We simply assume that each additional feature is independent of all the others, conditional upon y. In this case, we invert Bayes' Law by observing:

$$\frac{\Pr(y=1|\mathbf{l},\mathbf{n})}{\Pr(y=0|\mathbf{l},\mathbf{n})} = \frac{\Pr(y=1)}{\Pr(y=0)} \frac{\Pr(\mathbf{n}|y=1)}{\Pr(\mathbf{n}|y=0)} \frac{\Pr(\mathbf{l}|y=1)}{\Pr(\mathbf{l}|y=0)} \quad (4)$$

Regularized Logistic Regression

Given feature vectors $\mathbf{l}_{ij}$ and $\mathbf{n}_{ij}$, a linear logistic regression model takes the form:

$$\log \frac{\widehat{\Pr}(y_{ij}=1|\mathbf{l}_{ij},\mathbf{n}_{ij})}{\widehat{\Pr}(y_{ij}=0|\mathbf{l}_{ij},\mathbf{n}_{ij})} = \alpha_0 + \alpha' \mathbf{l}_{ij} + \beta' \mathbf{n}_{ij} \quad (5)$$

Most often, this model is fit by maximizing the conditional likelihood of the parameters for the training y given the feature values. However, this is not desirable when the number of features is too large. In order to prevent overfitting, a regularization parameter λ is introduced. Then we have a modified maximization problem.

$$(\hat{\alpha}, \hat{\beta}) = \arg\max_{\alpha,\beta} \log lik(\alpha, \beta) - \lambda \|\beta\|^2 \quad (6)$$

Here we penalize the coefficients that are associated with type features but not the ones associated with location features. This is because type features are only rarely active, whereas location features are frequently active, so their coefficients are easier to estimate.

Regularized logistic regression has been used in other text classification problems, as in (Zhang & Yang 2003). For further information on regularized model fitting, see for instance (Hastie, Tibshirani, & Friedman 2001).

Results

Models were fit using 25 randomly chosen sets of 2000 training documents each. Figure 4 shows their success rate at identifying the correct sentence in test documents. When the desired quotation spans multiple sentences, a prediction that chooses any of them is deemed correct.

Method	Features	Pct. Correct	Std. Error
Random	none	6.3%	–
Logist. Reg.	loc.	14.5%	0.3%
Naive Bayes	loc.; type	23.1%	0.5%
Logist. Reg.	loc.; type	25.8%	0.6%

Figure 4: Prediction match rate

A complication in viewing these results is the fact that some review documents contain multiple statements of their overall opinion. For instance, the following sentence is predicted as a sentiment summary: "*Mulholland Drive* is rapt and beautiful and absorbing, but apart from a few scenes ... it lacks the revelatory charge that *Blue Velvet* had 15 years ago." Although this does not match the Rotten Tomatoes quotation, it is otherwise an excellent choice.

The above example suggests that other approaches can be useful in evaluating automatically-produced sentiment summaries. This is one of many topics for further study in sentiment summarization. Although Rotten Tomatoes is an excellent source of supervised data in the movie domain, the summarization problem will differ according to context. In some cases, we will want methods that do not require large amounts of domain-specific supervised data. Here we have treated the problem as one of text classification, but many approaches are possible.

References

Gale, W. A.; Church, K. W.; and Yarowski, D. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities* 26:415–439.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag.

McCallum, A., and Nigam, K. 1998. A comparison of learning models for naive bayes text classification. In *Working Notes of the 1998 AAAI/ICML Workshop on Learning for Text Categorization*.

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)*.

Porter. 1980. An algorithm for suffix stripping. *Program* 14 (3):130–137.

Zhang, J., and Yang, Y. 2003. "robustness of regularized lienar classification methods in text categorization". In *Proceedings of the 26th Annual International ACM SIGIR Conference (SIGIR 2003)*.