CS224N/Ling237 Final Project
Picking the Fresh from the Rotten: Quote and Sentiment Extraction from Rotten Tomatoes Movie Reviews.

Eric Yeh

1. Abstract

Being able to quickly determine whether a movie is worth watching is one of the major appeals of the movie review website Rotten Tomatoes (www.rottentomatoes.com). The site gathers reviews from known movie critics, with each critic's review assigned a binary rating expressing the review's sentiment, *fresh* (positive) or *rotten* (negative), and a *quote*, the single sentence/phrase from the review that best exemplifies that rating. This form capsule summarization allows users the ability to discern the consensus of a group of critics at a glance. A system was constructed to perform the "Rotten Tomatoes Task" of identifying the quote and rating from an unlabeled review. A maximum entropy classifier was trained on a corpus of reviews from Rotten Tomatoes that had their ratings and accompanying quotes identified. Lexical and positional features were used. In addition, the notion of regional coherence, from text-segmentation work, was applied to create features that would help identify potential quotes from a review. A discussion of the nature of this "Rotten Tomatoes Task" is given, as well as future avenues of investigation, including alternative evaluation methods.

2. Overview

The ability to be able to quickly determine the overall sentiment of a document is a topic that has been garnering much interest in recent years, especially with the increase in the number of reviews online. The average user does not have the time to read through each and every product review on the web, to arrive at an informed decision about whether or not to make a purchase. It is this ability to quickly survey the consensus of a group of trusted critics[1] that makes the movie review website Rotten Tomatoes (www.rottentomatoes.com) popular.

For a given movie listed on Rotten Tomatoes, its review page consists of what essentially are very small capsule summaries of a critic's review of that movie. Each of these summaries consists of a rating and a *quote*. The rating is binary: it is either *fresh* (positive), or *rotten* (negative). The reason for this rating strict scheme is simple: ultimately the user needs to make a "go" or "no-go" decision on whether or not to spend $10+ to see the movie (i.e. one does not pay $6 out of a $10 admission for a movie rated 3/5 stars). The quote is the sentence or phrase, chosen by the Rotten Tomatoes editorial staff, that is considered to be the most representative of rating assigned by the review (Figure 1).

"*Fails on so many levels that it is difficult to even know where to start.*"

"*A flawed masterpiece.*"

"*You'll be rooting for these people to get slaughtered out of sheer boredom.*"

Figure 1: Example quotes from Rotten Tomatoes.

Currently, this "Rotten Tomatoes Task" of reading, classifying, and extracting quotes is performed entirely by the human editorial staff. Obviously this is a labor-intensive task, and any form of usable automation can be a boon for consumers looking to scan product reviews, as well as the editors that have to do this sort of thing on a weekly basis.

Movie reviews have certain characteristics that make them unique from other types of reviews. Word types that would normally be associated with sentiment, i.e. "good," "bad," "sucks," are oftentimes used in

[1] For the criteria, please see http://www.rottentomatoes.com/pages/critics.

objective sentences that describe plot elements, which are not indicative of the reviewer's rating. Worse yet: certain reviewers tend to throw in other seemingly opinionated elements that have no immediate and obvious bearing on how they felt about a movie; examples include descriptions of their moody journey to the film festival on a bus, or how an eating disorder is destroying his/her personal life. Thus, a review can be broken down into *subjective* sentences that are expressive of the reviewer's sentiment about the movie, and *objective* sentences that do not have any direct or obvious bearing on or support of that sentiment. The Rotten Tomatoes assigned quote can be thought of as the most subjective sentence of the review, as it is intended to capture the reviewer's sentiment.

Note that the notion of subjectivity in this case does not have any implication of the polarity of that sentiment: a sentence is considered subjective as long as it is indicative, and could be representative of the reviewer's opinion. Following, an objective sentence can contain adjectives and other word descriptors that at first glance can be considered expressive of an opinion, but ultimately has no bearing on the review's sentiment. For example, the following statement from a review of *Magadascar* would seem to be positive in nature, "*The animation is smooth and stylish, with blocky, exaggerated shaping that brings to mind a 3-D version of "Batman: The Animated Series."* However, it served to describe the movie's animation style and production, and ultimately was not reflective of the negative review assigned to it. As (Pang, et al., 2002) have shown, subjective and objective language does not easily fit sets of adjectives and word types that one would normally associate with them. Combined with the fact many reviews are written in a style that makes it just plain hard to figure out if the reviewer liked the film or not (i.e. movie reviews from "The Village Voice"), this makes movie reviews a challenging domain to work in.

Previous work that approached the "Rotten Tomatoes Task" split it into two problems: determining the "fresh"/"rotten" polarity of that review, and choosing a quote that best expresses that review. Review rating classification has generally been good, with accuracies around 80-90%, despite the difficulties with the inherent nature of movie reviews (Pang and Lee, 2004).

Determining which sentence from a review should be used as the quote (or the basis of the quote) has turned out to be a trickier problem. Studies that addressed quote extraction have treated it as a sentence-level classification problem (Beineke et al., 2004, Fingal et al., 2004). However, as those studies have noted, and for anyone who actually has had to do this task manually, it quickly becomes apparent that most reviews contain multiple candidate sentences that each can serve as equally well as the Rotten Tomatoes assigned quote. Selecting a potential quote from that set of good candidates to act as the "correct" quote can be an extremely subjective choice, which would explain why accuracies for previous studies were around 20-30%, when using the Rotten Tomatoes' quotes as the gold standard. In fact, 20-30% should be construed as good values, given the inherent difficulty of anticipating the editor's selection, and that (Fingal et al., 2004) showed that random selection achieved an accuracy of 3%, while human tagging achieved 40% accuracy.

Given these challenges, the system implemented for this project first focused on the task of extracting the quote from a given review, with rating classification second. The system approached the task of quote identification as a sentence level classification problem, using a discriminative, maximum-entropy classifier. Lexical and positional features were used to help determine which of the sentences in the review should be the quote. In addition, features inspired by text-segmentation techniques were used in an attempt to capture the flow of subjectivity and objectivity during the course of a review.

3. Data Preparation

A data dump of unique review IDs, ratings, quotes, critics, and review URLs was obtained from Rotten Tomatoes. Rotten Tomatoes does not cache the original review, as this would be copyright infringement. Instead, a hyperlink is provided that takes the user to the original review itself. This presented a data collection problem, as the original review text had to be downloaded, and the sentence the quote was extracted from, identified. Due to time constraints, reviews were culled from seven sources[2] using the

[2] The sources were (in no particular order): rogerebert.com, Film Threat, The New York Times, Deseret Evening News, sover.net, colossus.net, and the Arkansas Democrat-Gazette.

Python script *cull.py*. The textual content of the reviews were identified and extracted using regular expressions, crafted specifically for each source. Only reviews that were explicitly labeled as "fresh" or "rotten" were used, as the database also contained reviews that had no rating attached (i.e. movie/DVD release previews, etc…).

Reviews were "cleaned up" in stages, using the Python script *critics.py*. HTML tags were stripped, with the exception of <p> and <br> tags, which were converted into paragraph-break delimiter tokens, based on the how each given site encoded paragraph-breaks. Paragraph-breaks were retained as they have proven to be a useful positional feature for identifying quotes. Sentences were identified and separated using MXTerminator (Ratnaparkhi et al., 1997), lower-cased, and stemmed using the Porter-stemmer (Porter, 1980). Punctuation was retained as individual tokens, as they have been shown to be a significant indicator of sentiment (Pang, et al. 2002).

Each sentence was assigned an index value, based on its position in the review. Since paragraph-breaks were significant, they were treated as empty sentences for determining position values, and when extracting sentences from the review based on position.

Locating the sentence the quote came from proved a trickier task, as the Rotten Tomatoes editors often do not use the original sentence in its entirety, and sometimes modify the sentence by paraphrasing it, or changing it's tense. A bigram overlap scheme was implemented following (Fingal et al., 2004), where the sentence identified as the quote was the one that had the greatest bigram overlap score with the Rotten Tomatoes assigned quote (this quote is also lower-cased and stemmed). Reviews whose top matching bigram overlap score did not exceed a certain threshold were thrown out, to ensure the quality of the training, validation, and test sets. This threshold (around 0.3) was arrived upon after viewing a large number of quotes and best-overlap sentences.

Note that in some instances, the chosen quote was composed of two or more sentences. Since the classifier works only on a single sentences basis, the sentence that gave the best overlap with the Rotten Tomatoes quote was considered to be the quote for the review.

For each training review, the review’s sentences were labeled either as a paragraph-break, non-quote, or a “fresh” or “rotten” quote. Quotes have rating information attached to their labels, as previous studies have shown that the language used to describe a movie in a positive manner tends to be different from language used in a negative review (Fingal et al., 2004). This is congruent with experience from reading movie reviews, as critics have a tendency to use significantly different language, such as sarcasm, to lambaste a film. Thus training both on a positive and negative sentiment model was done to increase classifier accuracy. In addition, the “fresh”/”rotten” probabilities assigned to a sentence were used to help determine the sentiment polarity of the review.

The quote identification process resulted in about 9212 usable reviews (labeled with a rating, and quote sentence identified). From this set, a Python function randomly assigned 8000 reviews for the training set, 200 for the test set, and the remainder into development/validation sets of varying sizes.

4. Classifier and Feature Selection

The maximum-entropy classification approach was chosen for the system, as the underlying discriminative model allows for easy inclusion of a variety of features and avoids issues of independence or double counting that can occur with generative models (Manning and Klein, 2003). For an input datum *d* (composed of a set of features), the probability of that datum being labeled as class *c* is given as,

$$P(c \mid d, \vec{\lambda}) = \frac{\exp\left(\sum_i \lambda_i f_i(c,d)\right)}{Z(D)}$$

with λ_i representing the weight associated with feature *i*, and *Z(D)* the normalization factor, computed by summing across all possible class labels. The function $f_i(c,d)$ acts as a binary trigger, activating only if the given feature has been seen with the given class at least once in the training data.

$$f_i(c,d) = \begin{Bmatrix} 1, & \text{Count}_i(\text{d,c}) > 0 \\ 0 & \text{otherwise} \end{Bmatrix}$$

The weights are tuned using the gradient minimizing LBFGS optimization algorithm (Manning and Klein, 2005). For this project, feature weight training was capped at 25 iterations. The value of 25 was strictly arbitrary, chosen as an eyeballed medium between accuracy and computation time.

In this project's system, each sentence in the review is treated as an input datum, with a set of basic features based off of the lexical content of that sentence, as well as the sentence's position. The following basic features types were used for sentences:

- Lexical: unigrams and bigrams.
- Positional: position of the sentence in the review, and in the paragraph.

In addition, subjectivity/objectivity coherence features were used (described below).

Positional features have been shown to be good indicators how likely a sentence is the quote (Beineke et al., 2004). As the editorial staff at Rotten Tomatoes can attest, most assigned quotes occur at either the beginning or end of a review, and tend to appear at the beginning of a paragraph. These positional features were based on which portion of the review the current sentence occurs in (using 20% increments), as well as which part of the paragraph the sentence occurs in.

Statistics of positional features, for sentences that were labeled as quotes, adds weight to the notions that quotes tend to bookend a review (Figure 1), and are likelier to start a paragraph (Figure 2). Note that a significant number of reviews had paragraphs consisting of a single sentence, and were treated as a separate paragraph location feature.

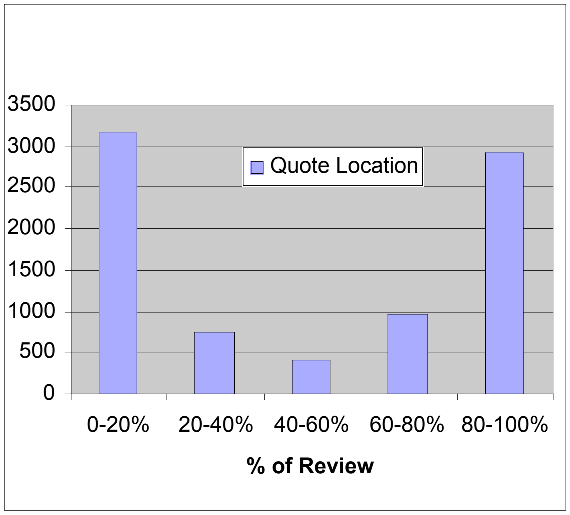

Figure 1: Position of quote within a review.

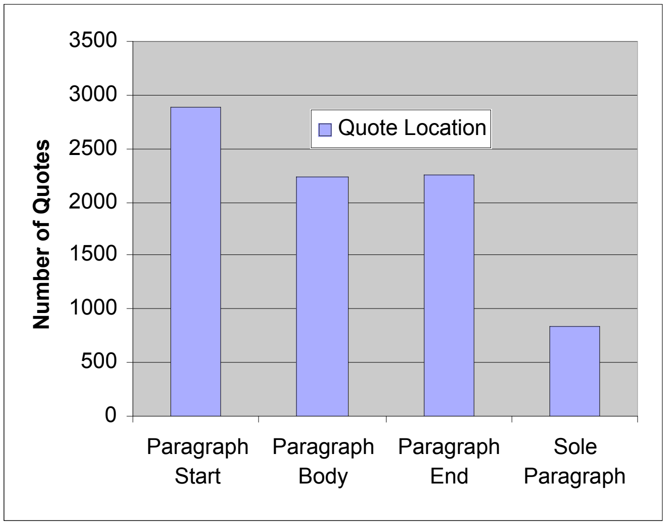

Figure 2: Position of quote within a paragraph.

5. Subjectivity/Objectivity Coherence

Discourse segmentation techniques such as TextTiling (Hearst, 1997) determine the similarity between regions of text via various topicality measures, such as a cosine similarity over vectors of word types. This similarity information is used to segment a given document into regions of text that are "self cohesive" units of discourse. For instance, TextTiling can be used to separate a single technical article into smaller sub-documents, each constituting a sub-topic from the original article.

This idea of cohesiveness was used to indicate segments of a review that are more subjective in nature versus those that are more objective. Movie reviews usually have a smooth "flow" of subjective and objective sentences, instead a haphazard and disjoint mixture. For example, the following review of *Labyrinth* by Roger Ebert (Figure 3), obtained from http://rogerebert.suntimes.com/apps/pbcs.dll/article?AID=/19860627/REVIEWS/606270302/1023, shows that a review tends to be composed of sequences of subjective and objective chunks, usually on a paragraph level.

> *"Labyrinth" is a movie that obviously was made with infinite care and pains, and it began with a real inspiration: Why not create a fantasy out of some of the drawings of M. C. Escher, who is famous for visual paradoxes such as a room with staircases that go "up" in every possible direction? The movie is an impressive production that is often good to look at. Some real thought went into it and the David Bowie soundtrack is fine, yet there's something missing. It never really comes alive.*
>
> *The film takes the form of a nightmare that visits the heroine, an adolescent girl named Sarah (Jennifer Connelly) who lives in a dreamworld of magic and legend and fairy princesses and enchantments.*
>
> *She's left to baby-sit for her baby brother, and when she teasingly wishes the goblins would take him away, she gets her wish. She is visited by Jareth (David Bowie), the ruler of the mystical world that is just out of sight of ordinary eyes. He sets her a task: She can get the child back, but only by finding her way through an endless labyrinth to the castle in the center.*
>
> *Our first view of the labyrinth is impressive. Indeed, all the special effects in the movie are impressive, showing the director, Muppets creator Jim Henson, working at the top of his form. Inside the labyrinth, Sarah faces a series of horrific challenges and meets a lot of strange characters. We are reminded a little of "Alice in Wonderland." I have a problem with almost all nightmare movies: They aren't as suspenseful as they should be because they don't have to follow any logic. Anything can happen, nothing needs to happen, nothing is as it seems and the rules keep changing. Consider, for example, the scene in "Labyrinth" where Sarah thinks she is waking up from her horrible dream and opens the door of her bedroom. Anything could be outside that door.*
>
> *Therefore, we're wasting out psychic energy by caring. In a completely arbitrary world, what difference does anything make? "Labyrinth" is intended as another extension of Henson's muppetry, in which the creatures he creates are more scary and real than ever*
>
> *But they are still Muppets, and I think the Muppet idea works better when humans visit the Muppet world (as in the Muppets movies), rather than when Muppets turn up in the human world.*
>
> *One of the key characters in this film is Toby (played by Toby Froud). Froud is a midget who has been given a Muppet head to wear. And although the head is a good special-effects construction, I kept wanting to see real eyes and real expressions. The effects didn't add anything.*
>
> *One other problem is that the movie is too long. Without a strong plot line to pull us through, all movies like this run the danger of becoming just a series of incidents. There's no structure to the order of the adventures. Sarah does this, she does that, she's almost killed here, almost trapped there, until at last nothing much matters. Great energy and creativity went into the construction, production and direction of this movie, but it doesn't have a story that does justice to the production.*

Figure 3: Roger Ebert's review of *Labyrinth*.

Use of a direct word type based measure would be infeasible, considering the average length of a movie review, and how infrequently the same descriptions and adjectives are used (as the concern here was to discover subjectivity/objectivity, not topicality). To address this, a "subjectivity/objectivity" ratio was introduced, using a corpus of subjective and objective sentences, obtained from http://www.cs.cornell.edu/people/pabo/movie-review-data/ (Pang and Lee, 2004). Unigram language models of these subjective and objective sentences (using the Java *EmpiricalUnigramLanguageModel* class) were produced, giving $P_{Subjective}(s)$ and $P_{Objective}(s)$, the probability of sentence s being subjective or objective. These language models were used to produce features for the classifier, along the lines of (Beeferman et al., 1997).

The subjective vs. objective likelihood of the i^{th} sentence in a review, s_i, can be stated as,

$$\frac{P(Subjective \mid s_i)}{P(Objective \mid s_i)} = \frac{P(s_i \mid Subjective)P(Subjective)}{P(s_i \mid Objective)P(Objective)}$$

Since no prior data was tagged to indicate which sentences were subjective or objective, the priors were treated as being equal. Since the derived unigram language models give the probability of the sentence being subjective or objective,

$$\frac{P(s_i \mid Subjective)P(Subjective)}{P(s_i \mid Objective)P(Objective)} = \frac{P(s_i \mid Subjective)}{P(s_i \mid Objective)} = \frac{P(Subjective \mid s_i)}{P(Objective \mid s_i)} = \frac{P_{Subjective}(s_i)}{P_{Objective}(s_i)}$$

The log of this likelihood value was taken to normalize extreme values, giving a “subjective/objective ratio.”

$$subjObjRatio = \log\left(\frac{P_{Subjective}(s_i)}{P_{Objective}(s_i)}\right)$$

Thus positive values indicate a stronger likelihood of the sentence being subjective, while negative values indicate stronger objectivity. A graph of each sentence’s subjectivity/objectivity ratio, from Roger Ebert’s review of *Labyrinth* shows that indeed there are trends of subjective and objective language that match the flow of the review (Figure 4). Also interesting to note is the average subjectivity/objectivity ratio for the assigned quote was found to be 9.5537.

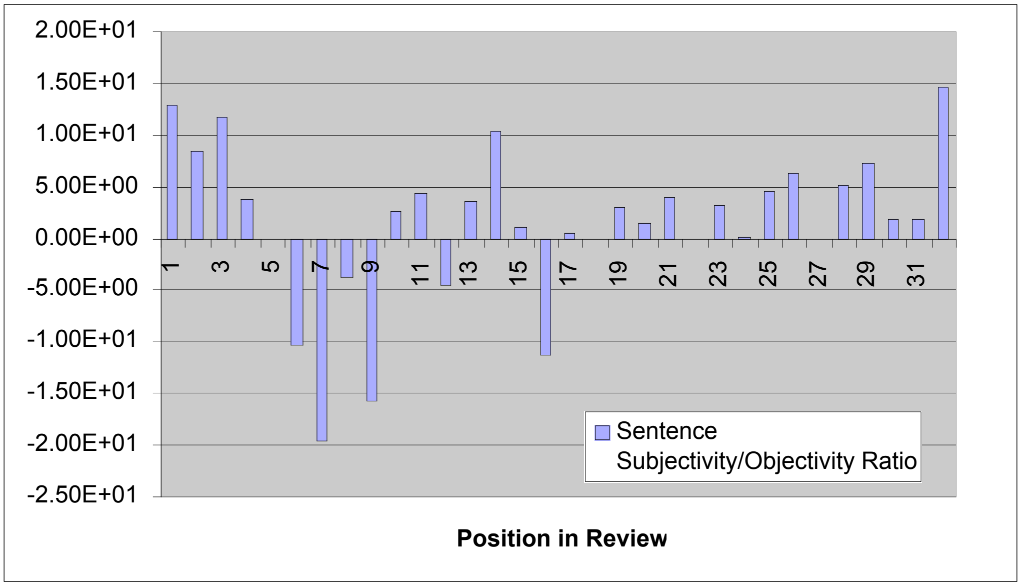

Figure 4: Subjectivity/Objectivity Ratio for Roger Ebert’s review of *Labyrinth*.

In order to capture the subjective/objective tendencies for regions of text, an adjustable span of size n was used, with the subjectivity/objectivity ratios from n sentences before and after the current sentence averaged together.

$$spanValue = \frac{\log\left(\frac{P_{Subjective}(s_{i-n})}{P_{Objective}(s_{i-n})}\right) + \ldots. + \log\left(\frac{P_{Subjective}(s_i)}{P_{Objective}(s_i)}\right) + \ldots + \log\left(\frac{P_{Subjective}(s_{i+n})}{P_{Objective}(s_{i+n})}\right)}{2n+1}$$

This is essentially the logarithm of the geometric average of the likelihoods of sentences in a $2n+1$ sized sentence span.

Since paragraph breaks can be indicative of a shift in tone, they are given a ratio score of 0, corresponding to a likelihood ratio of 1. In addition, spans that cross beyond the beginning or end of the review were given ratio scores of 0 as well.

In order to obtain a usable feature for the exponential model, the ratio is assigned a label based on its sign and magnitude (with "good" step sizes and cutoffs obtained by eyeballing the data).

6. Filtering

Feature sparseness can be a source of noise that corrupts results (Toutanova and Manning, 2000), so a filter was implemented to remove features that were seen *n* times or less in the training data. Asides from increasing classifier accuracy, this also lowered the number of feature weights the classifier had to learn, reducing the amount of computation time and resources needed.

7. Quote Selection and Evaluation

For each review in the test set, the probability of each constituent sentence being a "fresh" or "rotten" quote is determined, using the trained classifier. All sentences are ranked based on the sum of the "fresh" and "rotten" probabilities, with the sentence having the highest combined "fresh" and "rotten" mass identified as the guessed quote. Paragraph breaks do not have their quote probabilities computed, and are excluded from consideration.

Due to time constraints, the evaluation measure used was how accurately system identified the Rotten Tomatoes assigned quote for the given review. In spite of the issues given for using this as a measure, it does provide a common basis for comparison with similar studies.

In addition to determining the single best quote, a window of the top 5 review quote candidates was kept, and the accuracy of that window identifying the assigned quote was kept, the "Window Accuracy." Asides from giving higher accuracies values, use of this window can some insight into how well the system is performing overall: though the assigned quote may not be the most "appropriate," it is probably "pretty good" nonetheless, and should be ranked highly in the ratings. Using a window to detect the quote, in conjunction with the assigned quote detection accuracy, can give an indication of how well the classifier is ordering the sentences. Also, nothing prevents a potential automated system from displaying more than one quote as it's summary, and this can give an idea of how well such a system would perform.

8. Results and Discussion

Performance of the system using combinations of the basic features is given in (Table 1). The unigram language models for subjective and objective text were also used to identify the quote, in order to provide a baseline for comparison. As expected, the addition of bigrams and positional features gave increases in accuracies.

Included for comparison are accuracies for choosing the first line of the review and the last line as the selected quote. This is motivated by the observation that the most likely quote candidates do indeed appear at the beginning of a review, or at it's conclusion.

Features	Accuracy (% Correct)	Window Accuracy (% Correct)
Unigram	21.0	53.5
Unigram, Bigram	22.5	57.0
Unigram, Positional	27.5	65.5
Unigram, Bigram, Positional	31.5	65.5
Subj/Obj Ratios Only	19.5	N/A
First Line Only	10.0	N/A
Last Line Only	12.5	N/A

Table 1: Basic Features

The performance of subjectivity/objectivity ratio averages taken over increasing span sizes is given in (Table 2). What is interesting to note is despite having an accuracy of 26% for span sizes 0 and 2, a span size of 2 gives better performance in the window accuracy.

Features	Accuracy (% Correct)	Window Accuracy (% Correct)
Unigram, Span Size 0	26.0	61.5
Unigram, Span Size 1	22.5	58.5
Unigram, Span Size 2	26.0	63.0
Unigram, Span Size 3	24.0	63.0
Unigram, Span Size 4	22.5	58.5
Unigram, Span Size 5	24.5	62.5

Table 2: Span Size Performance

As for why span sizes 0 and 2 performed well, one possible explanation is at span size 0, the added information from tagging individual sentences with their subjectivity/objectivity ratio gave an increased boost. However, increasing the span by one gave a decrease, as the span was unable to straddle paragraph breaks (recall that paragraph breaks are considered to be empty lines, when indexing a review, but are not taken into consideration when computing probability masses). By increasing the span size to 2, a span can cross over a paragraph break and capture some information from the previous paragraph. Given how small paragraphs tend to be in reviews, a span size of 2 is probably able to capture an entire paragraph's worth of sentences. Increasing the span sizes give a decrease in accuracies, probably because too many sentences were spanned to provide useful data.

Also note that though the accuracies for detecting the assigned quote for span sizes 0 and 2 were identical (26%), the window accuracies were not. As stated previously, the sentences are ranked based on how likely they are to be the assigned quote, and the use of the window can capture the quality of this ranking. Given this, a span size of 2 was considered to be the optimal setting for this feature.

The effect of feature filtering was explored, with the optimal threshold found at 3 occurrences (Table 3).

Filter Threshold	Accuracy (% Correct)	Window Accuracy (% Correct)
0	31.5	65.5
1	30.5	64.0
2	32.5	66.0
3	33.0	66.0
4	32.0	67.0
5	32.0	67.0
6	31.5	67.5
7	31.5	67.0

Table 3: Filtering, using Unigram, Bigram, and Positional Features

Given the above results, an “best-found” combination of all basic features, a span size of 2, and filter threshold of 3 gave an accuracy of 33% (Table 4).

Features	Accuracy (% Correct)	Window Accuracy (% Correct)
Unigram, Bigram, Positional, Span Size 2, Filter Threshold 3	33.0	69.0

Table 4: Best Settings.

What is interesting to note is the increase accuracy for guessing the assigned quote seems to peak at a certain point, despite the fact that each feature individually increased performance. Perhaps some feature overlap was occurring between the use of the spans and the lexical features, though this was not reflected in the validation data (an overall increase was found). Also, the window accuracy has increased for using the best-found features-set, indicating that the quality of the quote candidate ordering has increased as well.

The impact of increasing sizes of training data was also explored, using the best-found settings (Table 5).

Training Size (No. reviews)	Accuracy (% Correct)	Window Accuracy (% Correct)
1000	26	62.5
2000	26	69.0
3000	29.5	68.5
4000	30.0	69.0
5000	28.5	68.0
6000	31.0	68.5
7000	33.5	68.5
8000 (full set)	32.5	67.0

Table 5: Training Data Size.

Despite the general trend of increasing accuracies, the increase is not strictly monotonic. One possible explanation for these fluctuations is though the training, development/validation, and test sets were divided randomly, the subsets of the training reviews used to conduct this test were not. The training reviews were divided based off of a directory listing of the training data directory. Each training review was stored as a separate file, and named after its review ID. It is likely that the review IDs are assigned on a per-author basis. Coupled with the fact that some authors are more prolific than others (most notably Roger Ebert), this could have caused some skew in the data.

Another test that was conducted was to sum up the mass for “fresh” and “rotten” from the sentences in the window, and to assign a rating for the review based on which label had a higher mass. This, along with the Window Accuracy, was evaluated over an increasing window size. Again, the best-found settings were used. Note that the rating accuracy over all sentences in the review was 75.5%.

Window Size	Window Accuracy (% Correct)	Rating Accuracy (% Correct)
1	33.0	70.0
2	44.5	74.0
3	54.5	72.0
4	62.5	73.5
5	69.0	73.0
6	72.5	76.5
7	76.0	75.5
8	79.0	76.0
9	82.0	76.0
10	84.5	75.5

Table 6: Increasing Window Size.

As expected, the window accuracy increases the window size. The point of greatest accuracy increase occurs from window sizes 1-5. This indicates that although the Rotten Tomatoes assigned quote may not always be the top choice, it is usually highly ranked.

What is not immediately expected is the quality of the rating does not appear to be (significantly) better on the average, compared to summing the masses over all the sentences in the review. One would expect these sentences to be the most indicative of the review's sentiment, and classifying over those should have less noise, and given improvement over using the entire review, as the results of (Pang and Lee, 2004) would indicate. However, closer analysis shows that the "fresh" and "rotten" probabilities assigned to each sentence already take into account the subjectivity and objectivity of that sentence, so the idea of the sums over the entire review giving the best rating should not be surprising.

Also interesting to note is the top rated sentences, when placed in order, can also act as a stripped version of the original review. For instance, the top three quote-candidates for Labyrinth read, when placed in positional order can act as a highly condensed version of the original review that focuses on why the reviewer disliked the film (Figure 5).

The movie is an impressive production that is often good to look at.
One other problem is that the movie is too long.
Great energy and creativity went into the construction, production and direction of this movie, but it doesn't have a story that does justice to the production.

Figure 5: Top 3 Quotes for *Labyrinth*, in order.

In general, the classifier seems to be able to capture the quote in the window, and the rating of the review. However, there were a number of cases where the window was not able to capture the quote, and the rating was missed. Oftentimes these reviews did not contain any overtly negative elements. On the other hand, the review was written in an entirely sarcastic manner. A negative review for *Waterboy* generally read along the lines of its assigned quote: *"How nice it would be, I thought, to give Adam Sandler a good review for a change."* Other reviews that the system missed completely were what could be best described as "gentle put-downs." The reviewer generally maintained a positive tone throughout the review, making efforts to praise elements of the movie (to the point that it would not be unreasonable to mistake it for a "fresh" review), but ultimately expressed mixed feelings and a negative sentiment. For example, this quote is indicative of the tone of a review for *Assassination Tango*: *"But I have seen countless movies about assassins and not a few about the tango, and while Duval's movie doesn't entirely succeed, what it attempts is intriguing."*

9. Conclusions and Future Directions

As stated earlier, an accuracy metric using the Rotten Tomatoes assigned quote may not be the best metric to use, given that a review can contains several good quote candidates. A better evaluation would have been to tag all sentences as being either subjective or objective, and to have the classifier attempt to discover them, using precision and recall for evaluation. Though this may seem to be an impossibly subjective task, especially when using input from multiple humans, (Hearst 1997) gives some interesting strategies for combining text-tagging data from multiple humans.

The coherence measure used was obviously not very sophisticated, and future avenues of investigation include the use of better methods to smooth, detect, and take advantage of trends from this data. Some of the methods described for use with the TextTiling algorithm may be of use (Hearst 1997). The underlying model for determining the subjectivity/objectivity ratio can also be improved, such as using better language models.

Due to the nature of the domain, there are obviously large amounts of unique proper nouns, and adding in part-of-speech tagging information would be a simple extension to try.

Finally, though detecting sarcasm and other such "thwarted expectation" language seems to be an "AI complete" problem, more sophisticated discourse extraction and co-reference resolution techniques could be used to try to discover features that would be good indicators for this type of text (Pang et al., 2002). Perhaps techniques used for question-answering problems may yield some interesting results.

10. Acknowledgement

The author would like to thank the crew at Rotten Tomatoes for allowing him to make use of their hard-earned data (all hand-picked,100% human), and for their expertise in all movie matters.

11. References

Beeferman, Doug, Adam Berger, and John Lafferty. 1997. Text Segmentation Using Exponential Models.

Beineke, Philip and Trevor Hastie, Christopher Manning, and Shivakumar Vaithyanathan. 2004. An Exploration of Sentiment Summarization.

Fingal, Dan, Jeff Michels, and Jamie Nicolson. 2004. CS224N Final Project: Summarizing Movie Reviews.

Hearst, Marti A. 1997. TextTiling: A Quantitative Approach to Discourse Segmentation.

Manning, Christopher and Dan Klein. 2003. Optimization, Maxent Models, and Conditional Estimation without Magic. Tutorial at HLT-NAACL 2003 and ACL 2003, http://www-nlp.stanford.edu/software/classifier.shtml.

Manning, Christopher and Dan Klein. 2005. CS224N Assignment 3: MaxEnt Classifiers and POS Tagging, http://www.stanford.edu/class/cs224n/hw/hw3.pdf

Pang, Bo, and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts.

Pang, Bo and Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of EMNLP*, pp. 79-86.

Porter, Martin. 1980, An algorithm for suffix stripping. Program, Vol. 14, no. 3, pp 130-137.

Reynar , Jeffrey C. and Adwait Ratnaparkhi. 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing, March 31-April 3, 1997. Washington, D.C.*

Toutanova, Kristina and Christopher Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger.

CS224N/Ling237 Final Project – Additional Report
Picking the Fresh from the Rotten: Quote and Sentiment Extraction from Rotten Tomatoes Movie Reviews.

Eric Yeh

1. Summary

This report describes additional work performed after the CS224N final project was submitted, and comments returned. Per comments from the original report, the test and validation sets were increased to 1000 reviews each, and feature tests from the original project were retested. Several new features were tested, motivated by a need to better describe the flow of subjective vs. objective language in a review, and by language that could help identify and frame the tone of a sentence. In addition, a new metric, the Average Window Position, was introduced in order to describe how well the actual human selected "Rotten Tomatoes" quote ranked in comparison to the other candidates, based on the given method.

2. Larger Test and Validation Sets

The original test size of 200 reviews proved to be too vulnerable to random fluctuations, so the test and validation sets were increased to 1000 reviews each. This did reduce the size of the training set to 7124 documents, but this was felt to remain a "sufficiently large" training set.

3. New Metric: Average Window Position

Due to the fact that the choice of the "Rotten Tomatoes quote" indeed is highly subjective, and that most reviews contain more than one possible candidate, a window of the top five quote candidates was monitored. If the quote was in that window, it was counted as a "hit," with the results being measured as Window Accuracy. When used with the actual quote selection accuracy, it was supposed to give an idea of how well the given method was ranking the selected quotes, as one objective was to produce a useful ranking of sentences that can serve as good one-line summaries of the review.

Continuing with the assumption that the Rotten Tomatoes quote is a "pretty good" representation of the ideal quote, the average rank of the quote is recorded, for quotes captured in that window. The idea is to convey how that quote would be distributed in that window, thereby giving an idea of how close the "near misses" were.

4.1 New Features: Subjective/Objective "Spikes"

A utility was constructed that displayed graphical representations of feature data on a per line basis. The features graphed were the subjective/objective ratio for the line, the window, and the probability of the line being the quote (per the classification method). to allow quick scanning of how each of those values behaved across reviews. A visual inspection of the output revealed that though the subjective/objective ratio span values were nicely smoothed, in some instances those values were significantly different from the ratio value at that line (Figure 1).

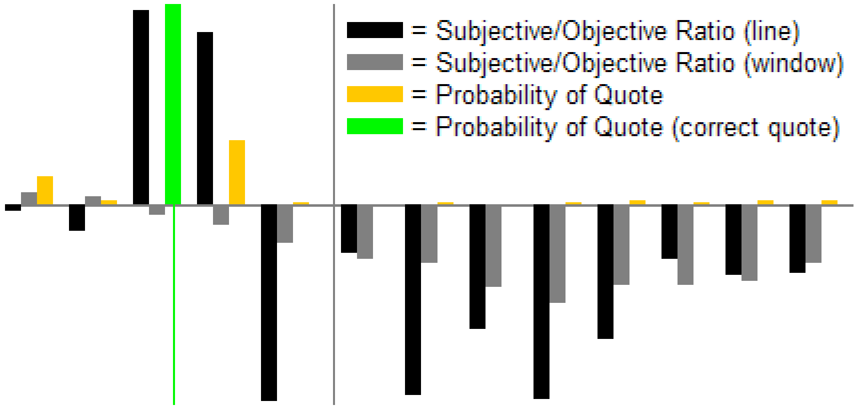

Figure 1: Subjectivity/Objectivity single-line ratios, and window ratios, for each line in a review. Each group of three bars corresponds to one line of the review, starting from the left.

To return the ability to represent how much any one line's ratio can "spike" in contrast with its neighbors, the subjective/objective ratio for each line is introduced as a feature, in addition to the span. The discretization method chosen is the same as the one used for discretizing the span.

4.2 New Features: Framing Language

Oftentimes the sentences that Rotten Tomatoes quotes are extracted from begin with language that frames those sentences quite differently in tone from the rest of the review. Some highly emphatic "to the point" quotes begin with certain phrases, such as "This movie …" or "A surprisingly …" (Figure 2).

"This movie's dumb."

"One of the most enjoyably inane movies of the season, this faux Southern Gothic offers an embarrassment of geek pleasures."

"A thoroughly adult comedy that, refreshingly, isn't toned down for the PG-13 brigade."

Figure 2

A significant number of Rotten Tomatoes quotes also begin with phrases indicative of a change or pivot in tone, compared with the surrounding context (Figure 3).

"Despite some serious flaws, and issues with the plot, "The Transporter 2" is a fun, silly, over the top, action film, that contain some truly inventive moments."

"Aside from constantly wondering how Jacquet's team pulled off capturing a never-ending series of amazing shots, you're left with the constant, inescapable thought that it ain't easy being a penguin."

"While it has its jumpy moments, it's really more of a suspense movie and mystery than a horror film."

Figure 3

In lieu of implementing a more sophisticated phrase extractor, a feature was added that simply re-added the first-*n* unigrams and first-*n* bigrams as distinct features. Though crude, the intent was to glean a rough intuition of how viable a more sophisticated strategy could possibly perform.

5. Results and Discussion

Given the ability of the larger test set to absorb random fluctuations, the "best" set from the initial project was used to re-evaluate the effect of filtering. The best feature-set was composed of unigram, bigram, positional, and a subjective/objective span size 2 (Table 1).

Filter Threshold	Accuracy (% Correct)	Window Accuracy (% Correct)	Average Window Position
3	27.4	68.7	2.2285
5	27.7	68.7	2.2368
10	27.8	68.8	2.2267
15	27.9	68.9	2.2322
20	28.1	69.0	2.2322
25	28.3	68.9	2.2264
30	28.4	69.0	2.2333
35	28.4	69.0	2.2290
40	28.3	69.1	2.2344
45	28.4	68.9	2.2235
50	28.2	69.0	2.2319
55	28.3	68.9	2.2322
75	27.8	68.5	2.2175
100	27.9	68.3	2.1991
200	26.4	68.4	2.2368

Table 1

To fully explain why performance appeared to peak around what appears to be an unusually high filter threshold of 30 warrants another study in itself. For the purposes of this report, a filter threshold of 30 was used, as it offered a good balance between classification performance over "full" features and the amount of time needed to run a trial.

Unfortunately, the standard deviation for the Average Window Postion was around 2.1-2.2, indicating that the Rotten Tomatoes quote was likely distributed relatively evenly amongst the evaluation window throughout the tests.

Because of the increased testing size, the effect of varying subjective/objective span sizes was reassessed (Table 2).

Features	Accuracy (% Correct)	Window Accuracy (% Correct)	Average Window Position
Unigram, Span Size 0	22.0	60.1	2.3380
Unigram, Span Size 1	21.2	61.6	2.4140
Unigram, Span Size 2	20.8	60.9	2.3826
Unigram, Span Size 3	20.4	60.7	2.4135
Unigram, Span Size 4	21.0	60.2	2.3588
Unigram, Span Size 5	22.3	60.7	2.3278
Unigram, Span Size 6	22.4	61.0	2.3377
Unigram, Span Size 7	21.7	60.6	2.3845
Unigram, Span Size 8	21.0	59.2	2.3209

Table 2

Again, there is a performance spike at span size 0, with performance decreasing slightly until span size 4 is reached. A possible explanation is the average number of lines per paragraph is 4.229 lines (including the paragraph break, which is also read by the feature extractor). A span size of around 4 or 5 would allow the feature to straddle just about the preceding and following paragraphs (albeit as a single value). Visual inspection of the graphical representations of the reviews shows that most paragraphs generally maintain a subjective or objective tone on the average. By extending the span to capture the general tone of the neighboring paragraphs, the feature could be capturing the per paragraph behavior.

The original feature-set, including the "best" found from the original report, were retested (Table 3).

Features	**Accuracy (% Correct)**	**Window Accuracy (% Correct)**	**Average Window Position**
Unigram	20.1	57.1	2.3520
Unigram, Bigram	21.9	61.8	2.4223
Unigram, Positional	24.1	67.1	2.2802
Unigram, Bigram, Positional	27.1	68.4	2.2997
Unigram, Bigram, Positional, Span Size 2 (Previous Best)	28.4	69.0	2.2333
First Line Only	13.3	N/A	N/A
Last Line Only	12.3	N/A	N/A
Subj/Obj Ratios Only	19.8	N/A	N/A

Table 3

The effectiveness of the subjective/objective spike feature with varying span sizes was tested, first in conjunction with unigrams only (Table 4), second with the “base set” of unigrams, bigrams, and positional features (Table 5). Per the unigram and subjective/objective span test, the best improvements were seen at span widths of 5.

Span Width (SubjObj Spike + Unigram)	**Accuracy (% Correct)**	**Window Accuracy (% Correct)**	**Average Window Position**
0	21.9	64.8	2.4290
1	22.7	63.5	2.3685
2	22.8	64.2	2.3769
3	23.0	63.4	2.3312
4	23.3	62.3	2.2777
5	24.5	63.1	2.2995
6	24.5	63.3	2.3002
7	24.0	63.2	2.3354
8	24.4	62.6	2.2875

Table 4

Span Width (All Features + SubjObj Spike)	**Accuracy (% Correct)**	**Window Accuracy (% Correct)**	**Average Window Position**
0	27.9	70.5	2.2681
1	27.9	71.8	2.3106
2	28.1	71.8	2.3301
3	27.2	70.6	2.3031
4	25.8	69.9	2.2976
5	29.2	70.1	2.2596
6	28.1	69.9	2.2504
7	28.0	70.1	2.2896
8	27.3	69.4	2.2709

Table 5

Framing language features, the first-*n* unigrams the first-*n* bigrams, were first tested without the use of the subjective/objective ratios, and had the unigram, bigram, and positional features activated. Using the first-*n* unigrams gave mixed results (Table 6), while the first-*n* bigrams gave consistently better results (Table 7). Given the features’ lack of sophistication, and the high filter threshold, it is not surprising that performance gains were marginal.

Number of first-n unigrams	**Accuracy (% Correct)**	**Window Accuracy (% Correct)**	**Average Window Position**
1	27.8	69.2	2.2832
2	27.0	68.5	2.2365
3	26.1	68.1	2.2291
4	28.6	69.5	2.2317
5	26.9	68.7	2.2213

Table 6

Number of first-n bigrams	**Accuracy (% Correct)**	**Window Accuracy (% Correct)**	**Average Window Position**
1	27.8	69.2	2.2832
2	27.2	68.8	2.2951
3	27.6	68.8	2.2485
4	27.7	68.3	2.2313
5	27.6	68.2	2.2317

Table 7

However, combining the subjective/objective ratio features with the framing language features results in consistently poorer performance than use of either alone. It is possible that the grossness of both the framing and subjective/objective language features resulted in conflicting features, and finer tuning both of them would result in better interplay between them. However, further data analysis needs to be performed to give a more adequate explanation.

Overall, the best performing feature set consists of using the unigram, bigram, positional, and the combined subjective/objective ratio features (Table 8).

Features	**Accuracy (% Correct)**	**Window Accuracy (% Correct)**	**Average Window Position**
Unigram, Bigram, Positional, Span Size 5, Spike	29.2	70.1	2.2596

Table 8

6. Conclusion and Future Directions

Increasing the test size resulted in what could be viewed as more "realistic" values for the results, with the effects of the subjective/objective ratio span decreased. As introducing the ratio spike showed, better and more sophisticated analysis still needs to be performed over the subjective/objective ratios.

The slight gain in performance from the use of simple framing language features indicates some potential in identifying key phrases that can help identify potential quotes. Indeed, chunk parsing via the use of simple cascaded finite state automata has had demonstrated success in extracting significant relations and entities from unstructured text (Hobbs et al., 1996). It is not inconceivable to utilize this type of technology for creating more sophisticated models and measures for subjective and objective language. Chunk-parsing style techniques have been used before to determine phrases that are indicative of reviewer sentiment (Turney 2002). Another possible approach for learning phrases (or categories of phrases) could utilize techniques used for learning link grammars (Temperley 2005).

7.References

Hobbs, Jerry R., Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1996. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural Language Text. *http://www.ai.sri.com/~appelt/fastus-schabes.html.*

Temperley, Davy, Daniel Sleator, and John Lafferty. 2005. Link Grammar. *http://www.link.cs.cmu.edu/link/*.

Turney, Peter D. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), Philadelphia, Pennsylvania, USA, July 8-10, 2002. pp 417-424. NRC 44946.*