Predicting Review Helpfulness on IMDB Movie Reviews

Gabor Angeli

Abstract

This report presents an approach to predicting the helpfulness of a movie review, taking into account features on the content of the article, the sentiments present in the article, and the context of the article with respect to previous postings. Results are compared loosely against the features presented in the previous work of **?**), which most closely resembles this task; however, we test on a different domain (IMDB movie reviews instead of Amazon product reviews).

We show that we improve upon this baseline, achieving a Pearson correlation coefficient of 0.493.

1 Introduction

Automatically assigning helpfulness to an online review is a task which has many applications. For instance, many reviews have few or no human annotated ratings; this may be the case for infrequently visited reviews, or for new reviews. An automatic helpfulness prediction method could help rate such reviews.

The motivation for automatic assessment of review helpfulness is many-fold. One motivation is to predict helpfulness of reviews which do not yet have a well-established human-annotated rating. This is often the case for new reviews, which never receive user attention, since they are not displayed near the top of the page by merit of either being an early review, or being a well-liked review. On a similar note, reviews which are thoroughly reviewed tend to garner more attention than those which are not, and therefore the established reviews tend to receive the most additional helpfulness ratings, independently of whether they are in fact the best reviews.

Another motivation is for text summarization. In this case, reviews which are deemed to be the most helpful should receive a higher weighting in the summarization task. Current summarization systems tend to consider all reviews as equally informative a-priori.

A last motivation, given in **?**), is to assist corporations marketing their products assess which aspects of the product the public likes and dislikes. The assumption is that higher rated reviews express more prominent opinions, both positive and negative.

The task of assessing review helpfulness is approached as a regression task: Given a training corpus of reviews each with a number of helpful/not-helpful annotations, predict the helpfulness of a new review. We approach the task using features based on a set of basic features augmented with additional semantic and temporal features, including sentiment features.

The approach is tested on a subset of the IMDB corpus of movie reviews. Although the corpora used in previous work are not available to compare against, the system outperforms an informed baseline consisting of the features presented in **?**).

2 Previous Work

Previous work focuses on assessing helpfulness in the context of predicting the helpfulness ranking for a product (**?**), low-quality review filtering (**?**), or as a tool for sentiment summarization (**?**), among other applications.

The work by **?**) was the earliest work present in the literature on the subject of assessing review helpfulness. The central theme of the paper was to use SVM regression to predict the ordering of reviews using a fairly large feature set.

The dataset used by the paper was a corpus of Amazon reviews, in the categories of *MP3 Players* and *Digital Cameras*. The dataset consisted of 821 products and 33,016 reviews for *MP3 Players*, and 1,104 products and 26,189 reviews for *Digital Cameras*.

We mimic the approach presented in the paper, however we report results on the predicted percentage of people who would find a review helpful matched with the gold standard. That is, we predict the value of h, defined as:

$$h(r \in R) = \frac{rating_+(r)}{rating_+(r) + rating_-(r)} \qquad (1)$$

Unlike the paper, only the Pearson and not Spearman correlation coefficients are reported, as our modified task is not particularly suited for evaluation using Spearman correlation.

We also adopt their feature set as a baseline to expand from (see Section **??**).

Work by **?**) focused on detecting low quality reviews. The paper again uses an dataset from Amazon

reviews, using only the *Digital Cameras* portion of the corpus.

Several new factors are introduced in the paper; notably, the *winner circle bias* and the *early bird bias*. The first notes that reviews which have a high helpfulness rating are more likely to receive more helpful ratings. The second notes that early reviews are often thought disproportionately helpful.

The paper accounts for these factors by manually reannotating their training data; in contrast, we attempt to learn these biases via features. In addition, we adopt a regression task rather than the bad-review classification task of the paper.

Other work has been done on helpfulness analysis (see **?**); **?**); **?**); *inter alia*). Our approach however does not adopt significant aspects of these, and unfortunately does not compare against their results.

3 Approach

The task presented is predicting the helpfulness of a review given features on the review, its associated movie, and the other reviews posted.

Experiments were performed on a subset of the IMDB corpus (described in Section **??**) and trained using Linear Regression (described in Section **??**). Features used in training are described in more detail in Section **??**.

3.1 Data

The reviews used in this paper were taken from the IMDB corpus: a collection of movie reviews from the site `http://imdb.com`. The corpus contains a total of 45,772 movies and 1,808,564 reviews. The text of the reviews was cleaned of special (non-ascii) characters; if possible, these terms were replaced with the ascii base of the character (e.g. removing accents), otherwise the character was replaced with a space. Reviews were tokenized using the heuristic document preprocessor included in the `JavaNLP` distribution.

Due to practical limitations, only a subset of the corpus was used; this subset used the first 1,000 movies (33,697 reviews) for training, and the subsequent 250 movies (11,365 reviews) for testing. This is comparable in size to the corpora of previous work. Development was done on a smaller subset of 100 movies for training and 25 movies for test, tuning on the training data of this smaller subset.

Reviews which had no helpfulness feedback were discarded from both the training and test sets, resulting in a corpus of 25,399 training reviews and 8833 reviews for testing. Discarding reviews with few (e.g. less than 5) helpfulness responses from the training data did not appear to help performance, in part perhaps due to mistraining features which depend on the number of reviews.

A sample review is shown in Figure **??**.

Topic 1	Topic 2	Topic 3	...
Lon	Russia	Best	
Anne	French	Madhumati	
BURN	contest	,	
Paula	genre	Raja	
THE	Soviet	Anand	

Table 1: Sample topics produced by the LDA topic model, although the topics were not used. The model was trained on 2152 reviews using 10 topics, for 100 iterations.

3.2 Training

Training was attempted using both SVM and linear regression; linear regression was deemed more reliable and is thus the method used in the reported results. The values of the features in the dataset were scaled to between 0 and 1 before passing them into the regression model; training took around 20 minutes.

Both SVM and linear regression suffered from overfitting on prolific features such as the unigram and bigram features (see Section **??**). Linear regression did not scale up to larger training corpus sizes due to memory constraints, whereas SVM regression exhibited severe over-fitting even on large training sets. Therefore, the unigram and bigram features were not used.

The SVM regression model, when used, was based on a Radial Basis Function (RBF) kernel, with hyperparameters C (misclassification cost) set to 0.25 and γ (for the RBF kernel) set to 0.01.

3.3 Topic Modeling

Although not used in this implementation, a topic model was trained on the dataset in an attempt to automatically capture relevant themes in reviews.

Attempts at training an LDA topic model were made both treating reviews as documents within a single movie, and treating reviews as documents among all movies. In both cases, the topics appeared too disorganized to be used as useful features. When topics were logical, they tended to cluster proper nouns, notably names of actors, or countries.

An example of topics extracted can be found in Table **??**, taking an optimistic sampling of the topics present. These topics were extracted from a training corpus of 100 movies (2152 reviews).

4 Features

Features extracted of the reviews were chosen to capture three types of phenomena:

- **Linguistic and Meta Features**: Features over the structure and syntactic content of the review. Grouped with this category are also the meta-features over the review.

- **Semantic Features**: Features over the semantic content of the review

Summary: Loved It!
Date: 22 September 2007
Author: ur2290488
Location: New York, NY USA
Helpful: 9 out of 10
Rating: 10 out of 10

I saw this series mentioned in the magazine Entertainment Weekly and looked for it online. I was pleasantly surprised to see the episodes posted on YouTube. I just finished watching this particular episode and have to say for a "fan effort I thought it was far superior to any of the spin offs. I think Gene Roddenberry would be very pleased with what's up there on that screen. What kept going through my mind is that I was watching something that should have occurred thirty years ago. I always felt that the spin offs had been taken over by people who really were not fans of the original and that Gene Roddenberry's original vision had been butchered horribly. "Star Trek: New Voyages is what should have happened all those years ago!
[...]

Figure 1: Example review (truncated), including meta-data. This review was taken from the movie *Star Trek: New Voyages To Serve All My Days (2006)*; the movie's meta-data has been omitted.

- **Temporal Features**: Features over the temporal positioning of the review with respect to other reviews

These three types of features are described in more detail below. Note that not all of these features are used in evaluation; for example the n-gram features are too sparse to be useful. The complete list is given to illustrate techniques attempted, and to compare with the features of previous work.

In addition to these features, which are intended to be largely domain-independent and general purpose, a feature is defined over the average helpfulness of other reviews on a movie (see Section **??**).

4.1 Linguistic Features

These features mirror the feature set proposed by **?**), however similar features are proposed in **?**) and **?**).

Structural Features Features over the review's structure, formatting, and length. These features include the length of the review (`LEN`), the number of sentences (`SENcount`), the average sentence length (`SENavelen`), as well as the number of questions (`SENquestions`) and exclamations (`SENexclams`).

Lexical Features Local features over the unigram (`UGR` and bigrams (`BGR`) of the review. As per **?**), *tf-idf* statistics were computed on the n-gram counts and used as features.

Syntactic Features Features were extracted over the Part of Speech tags of the reviews (`SYN`). Notably, the percent of the review corresponding to each POS tag was counted and used as a feature. This is a generalization of the technique of **?**). POS tagging was done using the Stanford MaxEnt tagger, using the *left3words* model trained on the Wall Street Journal.

Meta-Features The reviewer's rating of the movie (`RATING`) on a scale from 1 to 10 was included as a feature.

Additionally, a feature was extracted to account for the *Winner's circle* bias described first in **?**). The feature (`WIN`) counts the total number of helpful/unhelpful ratings given to a review.

4.2 Semantic Features

Similar to the approach of **?**), we incorporate features over relevant *Product Features* (`PRF`); the term originates from the context of Amazon reviews. In the case of the IMDB dataset, these entailed relevant movie-related keywords, extracted automatically from `http://www.filmsite.org/filmterms.html`; these terms were scraped from the internet and cleaned of quotes, terms in parentheses, and special html characters. The scope of the terms ranged from familiar concepts (e.g. *satire*, *VCR*) to technical phrases (e.g. *pixillation*, *chiaroscuro*). In total, there are 414 such terms.

Furthermore, features were extracted over sentiment words attached to these *Product Features* (`SENTIprf`). The intention is to capture phrases such as *great plot* or *exciting role*. These features were extracted using a simple heuristic approach: a word appearing before a *Product Feature* is assumed to be a modifier on the feature. If it has sentiment-information, a feature is extracted over the sentiment type and the product feature in question. Sentiment information is taken from the *General-Inquirer*. All sentiment classes are used.

Furthermore, features are extracted over the prevalence of sentiment words in the document as a whole

General Inquirer Sentiment	Description
Positiv	positive outlook
Negativ	negative outlook
Strong	strength
Weak	weakness
Active	active orientation
Passive	passive orientation
EMOT	related to emotion
Means	means of attaining goals
Eval@	judgment and evaluation
FREQ	assessment of frequency
IAV	explanation of an action
HU	general references to humans
Econ@	commercial, industrial, or business
Milit	military matters
Polit@	clear political character
Role	human behavior patterns
Kin	kinship
Female	referring to women
MALE	referring to men
ANI	referring to animals

Table 2: List of *General Inquirer* terms used in sentiment features. Sentiments not in this list are ignored, due to the large number of categories which are not useful for this task.

(SENTI). These sentiments are again taken from the *General-Inquirer*; each word is classified into its possible sentiments, and the count of that sentiment in the review is incremented. Word senses are not disambiguated, although future work could make use of POS information to limit the sentiments extracted for each word.

Sentiment features were extracted over the count of each sentiment type based on the number of words in the review expressing that sentiment. Due to the large number of categories in the *General-Inquirer* database, only a subset are used as features. This subset was chosen by manual inspection of the correlations between the sentiment category and helpfulness on the first 1000 movies (the training set). The full list of sentiment categories used is given in Table **??**; in general, most of the main categories and categories which describe conventional sentiments are kept.

Lastly, a feature is defined as the percentage of the article which is written in all caps (CAPS), which tends to convey some information about the reviewer.

4.3 Temporal Features

The last aspect the system attempted to capture was a temporal relationship between reviews. Notably, in general, later reviews tend to be marked less helpful than early ones (this is described as the *early-bird bias* in **?**)). Features were extracted both attempting to capture this general trend, and – although somewhat naively – attempting to capture the conjecture that this drop off is from reviews repeating similar topics. That is to say that helpful reviews are ones that introduce new information.

In the first category, features are extracted over the review's index with respect to its post time (INDEX), as well as over the seconds elapsed since the first review was posted (TIME).

In the second category, a feature is extracted over the percentage of unique words in the review which do not appear in any previously posted review (NEWWORDS). This is a simplification of a feature over novel topics introduced in each review, as trained by an LDA topic model. However, the topics extracted by the model were found to be of poor quality, both within a single movie considering reviews as documents, and across movies considering either reviews or movies as documents (see Section **??**)

4.4 Other Review Helpfulness

The IMDB corpus exhibits the peculiar phenomenon that movies tend towards having either mostly helpful rated or mostly unhelpful rated reviews. That is, a review on a movie is likely to have a similar helpfulness percent as other reviews on that movie.

We describe this phenomenon with a feature (OTHER_RATINGS), which takes the value of the helpfulness of all of the other reviews for the given review's movie. Note that the particular review in question is not included in this average.

This feature proves particularly well correlated with helpfulness (see Figure **??**), however is also very specific to the phenomenon observed in the IMDB corpus. Furthermore, looking at the helpfulness of other reviews at test time has an air of cheating around it; for instance, this feature prohibits classifying multiple unlabeled reviews in bulk, as each review will depend on the result of classifying the others.

None the less, the feature is included in the results reported.

5 Results

5.1 Feature Sets Evaluated

Roughly following the classification of features from Section **??**, we can group the full feature set into three broad categories:

- KIM denotes the lexical features similar to the system of **?**), with the addition of the product feature (PRF and SENTIprf) features.

  In the absence of previous work to compare to, this feature set serves as the baseline system.

- KIM+TIME denotes the above feature set combined with the temporal features

- KIM+SEM denotes the KIM feature set combined with the remaining semantic features (SENTI and CAPS)

- KIM+TIME+SEM denotes the union of all of the above features

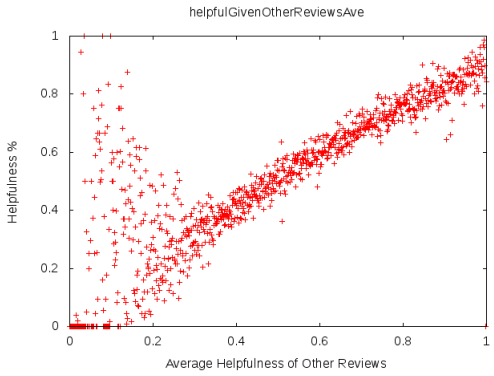

Figure 2: The correlation between the average helpfulness of other reviews in the same movie, and the review in question. For example, in the raw graph, a movie with three reviews rated at 0.0, 0.5, and 1.0 would be represented as three points {(0.75,0.0),(0.5,0.5),(0.25,1.0)}. To reduce clutter, this graph averages the Helpfulness (y) values for 1000 buckets of Other Review Helpfulness (x).

Feature Set	Pearson (train)	Pearson (test)
KIM	0.225	0.106
KIM+TIME	0.312	0.212
KIM+SEM	0.308	0.260
KIM+TIME+SEM	0.352	0.299
ALL	**0.486**	**0.493**
OTHER_RATINGS	0.435	0.458

Table 3: Pearson correlation for different feature sets (see Section **??**).

- OTHER_RATINGS denotes the feature over the average helpfulness rating of other reviews for the given movie (see Section **??**)
- ALL denotes the complete set of features. That is, KIM+TIME+SEM in addition to OTHER_RATINGS.

Each of these was evaluated using the Pearson correlation coefficient with respect to the gold labeled accuracy:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad (2)$$

5.2 Analysis of Results

The results of running the system on each feature set is given in Table **??**. A bar graph of the same results is shown in Figure **??**. The best performing feature set is, predictably, the ALL feature set with a Pearson correlation of 0.493. A graph of the output of the system on that feature set is showin in Figure **??**; since no pruning of the data was done based on number of helpful ratings, many 'Gold' points lie along common fractions.

A number of interesting conclusions arise from the results. Among them, the conspicuously low score of the KIM feature set, which would be expected to perform comparably to the dataset of **?**). A possible explanation for this is the lack of the unigram feature in our experiments, which was among the more expressive features presented in their paper. In fact, on small scale experiments, adding the unigram feature significantly improved performance on the training set, to the point of near-memorization however; this improvement did not translate to the test set. It's possible that the Amazon review dataset has a significantly smaller vocabulary (or equivalently, are shorter), making the feature effective on that dataset but not on the IMDB corpus.

Furthermore, the length of the review (LEN feature) – another feature which was found to be significant by **?**) – was not found to be overly useful in the IMDB domain. These two features account for two of the three features which they conclude are the most useful (the third being RATING). This would explain the dramatic difference in performance of the feature set between the two domains.

The second peculiar result is the disproportionate influence the OTHER_RATINGS feature has on the correlation. This likely indicates either that certain movies are more prone to have positive helpfulness feedback (e.g. 'nicer' people visit those pages); or else that there is another instance of the *winner's circle* bias occurring, where a user is hesitant to mark a review as helpful if there are few helpful votes overall on the page, and visa versa.

5.3 Classification

In addition to regression, the same model can perform classification of reviews into *helpful* or *not helpful* reviews, defined as being above or below a certain threshold. With a threshold of 0.5, the system achieves an accuracy of 69.9% on the training data, and 74.0% on the test data. The naive baseline for this task – in this case, always guessing *unhelpful* – achieves an accuracy of 54.2% on the training data, and 62.9% on the test data.

While this is not a groundbreaking feat of accuracy (11.1% improvement over majority guess), it serves as a proof of concept for the task, and as a sanity check for the correlation result.

6 Discussion

6.1 Motivation for New Features

This section aims to show that the new features introduced in this system are, in fact, correlated with helpfulness. Note that these correlations are not necessarily independent; also keep in mind that the plots of these correlations are over average values in buckets, and thus the correlation seems deceptively strong (in contrast, the raw data often appears as a uniform blob of dots). None the less, they provide some interesting insights.

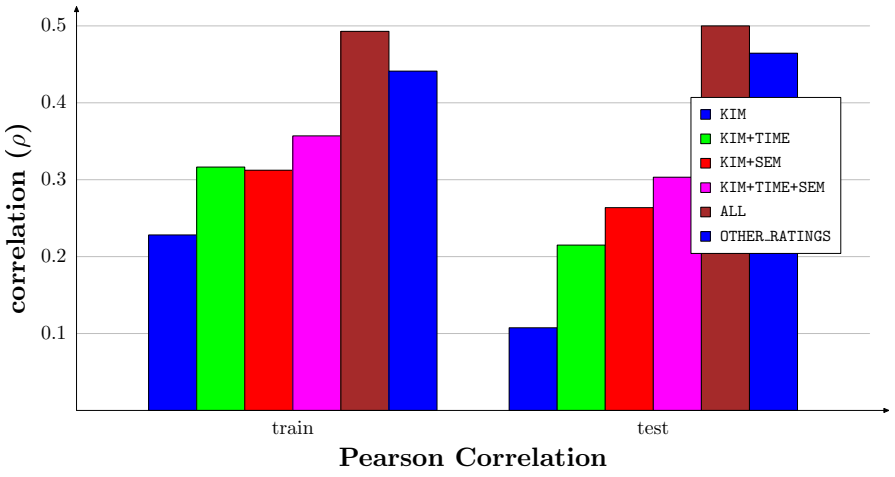

Figure 3: Pearson correlation for different feature sets (see Section **??**). The highest value is on the ALL feature set, denoted in brown. Note that adding both temporal and semantic features improve performance, and the addition of features compound on each other.

CAPS feature The conjecture the feature captures is that reviewers who often post in all-caps are likely to be less helpful, either because the reader dislikes the style, or because all caps is often an indicator of text that is written hastily and without proper thought. This conjecture is shown plausible (see Figure **??**); reviewers who use no or very few all-caps words are on average considered more helpful.

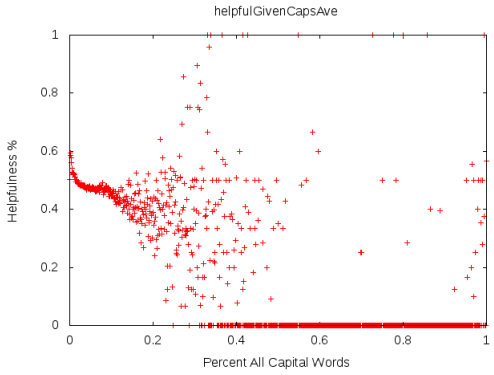

Figure 5: The correlation between the number of words in a review which are upper-case, that is contain an upper case character and no lower case characters, and the helpfulness of the review. To reduce clutter, this graph averages the Helpfulness (y) values for each of 1000 buckets (x).

TIME feature The conjecture captured by the feature is that reviews which are posted late happen to get rated less helpful. While this feature proved true for the INDEX feature (conforming to the exponential dropoff described in **?**), the TIME feature behaves somewhat strangely (see Figure **??**). Namely, helpfulness increases until around 6 years after the first post, and then decreases sharply until around 12 years after the first post. The reason for the discrepancy between this and the normal review index-wise exponential drop off is be interesting.

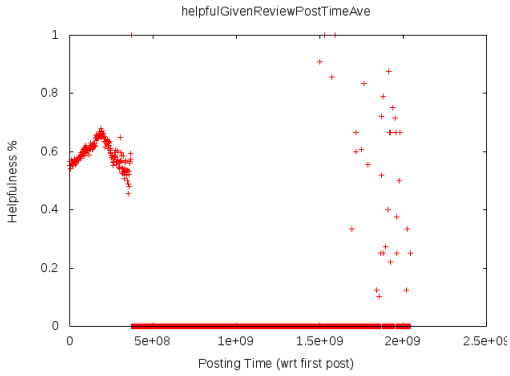

Figure 6: The correlation between the number of seconds since the first review of the movie was posted, and the helpfulness of the review. To reduce clutter, this graph averages the Helpfulness (y) values for each of 1000 buckets (x).

NEWWORDS **feature** This feature provides a naive substitute for analysis of new concepts being introduced in a review. The conjecture is that reviews which introduce new ideas (and, consequently, likely have many new words) are more helpful. This correlation is plotted in Figure **??**; in general the assumption seems valid that reviews which introduce new words are more helpful, although the shape of the function is not quite linear.

6.2 Future Directions

The project leaves open a number of interesting questions. Perhaps most apparent would be the question of why the OTHER_RATINGS feature is so effective, and