# Identifying Humor in Reviews using Background Text Sources

**Alex Morales** and **ChengXiang Zhai**
Department of Computer Science
University of Illinois, Urbana-Champaign
amorale4@illinois.edu
czhai@illinois.edu

## Abstract

We study the problem of automatically identifying humorous text from a new kind of text data, i.e., online reviews. We propose a generative language model, based on the theory of incongruity, to model humorous text, which allows us to leverage background text sources, such as Wikipedia entry descriptions, and enables construction of multiple features for identifying humorous reviews. Evaluation of these features using supervised learning for classifying reviews into humorous and non-humorous reviews shows that the features constructed based on the proposed generative model are much more effective than the major features proposed in the existing literature, allowing us to achieve almost 86% accuracy. These humorous review predictions can also supply good indicators for identifying helpful reviews.

## 1 Introduction

The growth of online feedback systems, such as online reviews in which users can write about their preferences and opinions, has allowed for creativity in the written communication of user ideas. As such, these feedback systems have become ubiquitous, and it's not difficult to imagine a future with smart systems reacting to user's behaviour in a human-like manner (Nijholt, 2014). An essential component for personal communication is the expression of humor. Although many people have studied the theory of humor, it still remains loosely defined (Ritchie, 2009), this leads to difficulties in modelling humor. While the task for identifying humor in text has been previously studied, most approaches have focused on shorter text such as Twitter data (Mihalcea and Strapparava, 2006; Reyes et al., 2012, 2010) (see Section 6 for a more complete review of related work). In this paper, we study the problem of automatically identifying humorous text from a new kind of text data, i.e., online reviews. In order to quantitatively test whether the review is humorous, we devise a novel approach, using the theory of incongruity, to model the reviewer's humorous intent when writing the review. The theory of incongruity states that we laugh because there is something incongruous (Attardo, 1994), in other words, there is a change from our expectation.

Specifically, we propose a general generative language model to model the generation of humorous text. The proposed model is a mixture model with multinomial distributions as component models (i.e., models of topics), similar to Probabilistic Latent Semantic Analysis (Hofmann, 1999). However, the main difference is that the component word distributions (i.e., component language models) are all assumed to be known in our model, and they are designed to model the two types of language used in a humorous text, including 1) the *general background model* estimated using all the reviews, and 2) the reference language models of all the *topical aspects* covered in the review that capture the typical words used when each of the covered aspects is discussed. Thus the model only has the parameters indicating the relative coverage of these component language models. The idea here is to use these parameters to assess how well a review can be explained by collectively by the reference language models corresponding to all the topical aspects covered in the review, which are estimated using an external text source (e.g., Wikipedia).

We construct multiple features based on the generative model and evaluate them using supervised learning for classifying reviews into humorous and non-humorous reviews. Experiment re-


*Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 503–512
Copenhagen, Denmark, September 7–11, 2017. 

sults on a Yelp[1] review data set show that the features constructed based on the proposed generative model are much more effective than the major features proposed in the existing literature, allowing us to achieve almost 86% accuracy. We also experimented with using the results of humorous review prediction to further predict helpful reviews, and the results show that humorous review prediction can supply good indicators for identifying helpful reviews for consumers.

## 2 Referential Humor and Incongruity

In this section we describe some observations in our data that have motivated our approach to solving the problem. In particular, we show that humorous reviews tend to reference aspects which deviate from what is expected. That is, in funny reviews, the authors tend to use referential humor, in which specific concepts or entities are referenced to produce comedic effects, which we call *aspects*. Here we define *referential humor* to be a humorous piece of text which references aspects outside of the typical context, in our case restaurant reviews. For the rest of the paper we use humorous and funny interchangeably.

Our study uses review data from Yelp. Yelp has become a popular resource for identifying high quality restaurants. A Yelp user is able to submit reviews rating the overall experience of the restaurants. The reviews submitted to Yelp tend to have similar context, in particular they mention several aspects rating the quality of the restaurant such as food, price, service and so on. This information is expected from the reviewer in their review, however it is not always the case since there is no requirement for writing the review. Yelp users are able to vote for a review in several criterion, such as funny, cool, and useful. This gives the users an incentive for not only creating informative reviews but possibly entertaining reviews.

In Figure 1, we show a humorous review, randomly sampled by using our classifier with a high probability of being funny, where the reviewer asserts that the food has extreme medicinal properties. The reviewer refers to "Nyquil" a common cold medicine to express the food's incredible ability to cure ailments. This appears almost surprising since it would not normally be mentioned in restaurants reviews. To identify the intended humor, we can use the references the reviewer

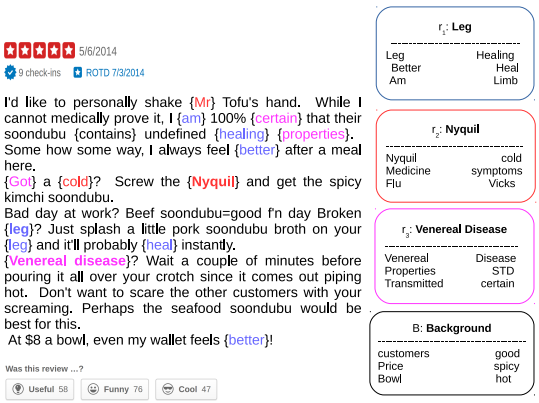


Figure 1: A funny review (left), with $K_d = 3$, aspect topics (right) contain words in their corresponding language model, probabilities removed for clarity, the colored (bracketed) word correspond to a different aspect assignment.

makes, e.g. Nyquil, as clues to what she is emphasising, e.g. the savory soondubu, by making such comparisons, e.g. the heavenly taste and amazing price. Yelp users seem to consider funny reviews which tended to deviate from what was expected into things which would seem out of place.

## 3 Language Models as a Proxy for Incongruity

Motivated by the observations discussed in the previous section (i.e., reviewers tend to reference some entities which seem unexpected in the context of the topic of the review), we propose a generative language model based on the theory of incongruity to model the generation of potentially humorous reviews. Following previous work on humor, we use the definition of incongruity in humor as "what people find unexpected" (Mihalcea and Strapparava, 2006), where "unexpected" concepts are those concepts which people do not consider to be the norm in some domain, later we formalize unexpectedness using our model.

We now describe the proposed model in more detail. Suppose we observe the following references to $K_d$ topical aspects $A_d = \{r_1, r_2, ..., r_{K_d}\}$ in a review $R_d = [w_1, w_2, ..., w_{N_d}]$, where each $r_i$ corresponds to an aspect reference (i.e. NyQuil in our running example), and $w_i \in V$, where $V$ is the vocabulary set. The model generates a word, for some review, at a time, which talks about a specific aspect or is related to the language used in Yelp more broadly; we call the latter the background language model. Thus a word is generated from a mixture model, and its probability is an interpola-

[1] www.yelp.com

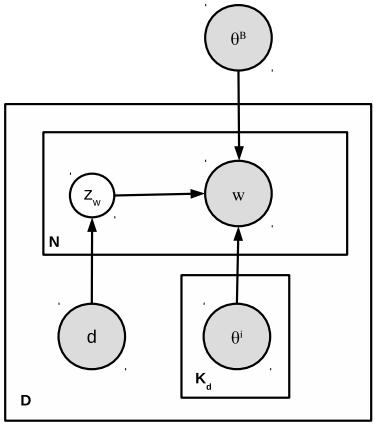


Figure 2: Generation model for reviews, where the $d$th review has $K_d$ aspects in the review. The shaded nodes here are the observed data and the light node $z$ are the latent variables corresponding to aspect assignments.

tion of the background language and the language of the references as shown in Figure 2.

These aspects provide some context to the underling meaning of a review; the reviewers use these aspects for creative writing when describing their dining experience. These aspects allow us to use external information as the context, thus we develop measures for incongruity addressing the juxtaposition of the aspect's context and the review. The review construction process is represented in a generative model, see Figure 2, where the shaded nodes represent our observations, we have observed the words as well as the referenced aspects which the reviewer has mentioned in their review. The light nodes are the labels for the aspect which has generated the corresponding word. Since the background language model, denoted by $\theta^B$, is review independent, we can simplify the generative model by copying the background language model for each review, thus we can focus on the parameter estimation for each review in parallel.

A key component to the success of our features is the mesh of background text from external sources, or *background text sources*, and the reviews. In our example, Figure 1, Nyquil is a critical component for understanding the humor. However it is difficult to understand some references a reviewer makes without any prior knowledge. To do so, we incorporate external background knowledge in the form of language models for the referenced aspect present in the reviews. If the reviewer has made $K_d$ references to different aspects $A_d$ in review $R_d$, then for each $r_i$ there is a corresponding language model $\theta_w^{r_i} = P(w|\theta^{r_i})$ over the vocabulary $w \in V$. For simplicity, we describe the model for each document, and use the notation $\theta_w^i$ and $\theta^i$ for the corresponding language model of $r_i$.

### 3.1 Incorporating Background Text Sources

As described before, some features we will use to describe incongruity correspond to the weights of the mixture model used to generate the words in the review, which take into account the language of the references she will make or allude, as shown in Figure 2. The probability that an author will generate a word $w$, for the $d$th review given corresponding aspects $\Theta = \{\theta^B, \theta^1, ..., \theta^{K_d}\}$, is

$$P(w, d, \Theta) = \sum_{z=0}^{K_d} P(w, z, d, \Theta) =$$

$$\sum_{z=0}^{K_d} P(w|z, \Theta)P(z|d) = \lambda\theta_w^B + (1-\lambda)\sum_{i=1}^{K_d} \pi_i\theta_w^i$$

Note $K_d$ indicates the different aspects the reviewer will mention in a review, $R_d$, and hence it can vary between reviews. $\theta_w^B = P(w|z = 0, \Theta)$ is the probability that the word will appear when writing a review (e.g. background language model) and $\theta_w^i$ can be interpreted as word distributions over aspect $i$. Here $\lambda = P(z = 0|d)$ is the weight for the background language model and $\pi_i = \dfrac{P(z = i|d)}{1 - P(z = 0|d)}$ denotes the relative weights of the referenced aspect's language models used in the review. We denote our parameters for review $R_d$ as $\Lambda_{R_d} = \{\pi_1, ..., \pi_{K_d}, \lambda\}$. Note that the parameter set varies depending on how many references the review makes. In order to estimate $P(w|\theta^i)$, we first need to find the aspects that the user is mentioning in their reviews. In general aspects can be defined as any topics explicitly defined in external background text data; in our experiments we define aspects as Wikipedia entities. In subsection 5.1, we describe one way of obtaining these aspects, but first we describe the estimation methodology.

### 3.2 Parameter Estimation

To estimate our parameters $\Lambda_{R_d}$, we would like to maximize the likelihood of $P(R_d)$, which is the same as maximizing the log-likelihood of $P(R_d)$.

That is

$$\hat{\Lambda} = \text{argmax}_{\Lambda} \log P(R_d|\Lambda)$$
$$= \text{argmax}_{\Lambda} \sum_{w \in V} c(w, R_d) \log \left(P(w, d, \Theta)\right)$$

Here $c(w, R_d)$ represents the number of occurrences of the word $w$ in $R_d$. In order to maximize the log-likelihood we use the EM algorithm (Dempster et al., 1977), to compute the update rules for the parameters $\lambda$ and $\pi_1, ...\pi_{K_d}$. For the E-Step, at the $n+1$th iteration we have

$$P(z_w = 0) = \frac{\theta_w^B \lambda^{(n)}}{\left(\sum_{l=1}^{K_d} \theta_w^l \pi_l^{(n)}\right)(1 - \lambda^{(n)}) + \theta_w^B \lambda^{(n)}}$$

$$P(z_w = j) = \frac{\theta_w^j \pi_j^{(n)}}{\sum_{l=1}^{K_d} \theta_w^l \pi_l^{(n)}}$$

Where $z_w$ is a hidden variable indicating whether we have selected any of the aspect language models, or the background language model, when generating the word $w$. The update rules for the M-Step are as follows:

$$\lambda^{(n)} = \frac{\sum_{w \in V} c(w, R_d) P(z_w = 0)}{n}, \pi_j^{(n)} = \frac{\sum_{w \in V} c(w, R_d) P(z_w = j)(1 - P(z_w = 0))}{\sum_{l=1}^{K_d} \sum_{w \in V} c(w, R_d) P(z_w = l)(1 - P(z_w = 0))}$$

We ran EM until the parameters converged or a small threshold was reached. Note there is some similarity to other topic modelling approaches like PLSA (Hofmann, 1999). PLSA is a way to soft cluster the documents into several topics, in doing so a word distribution for each topic is learned. In our work we make the assumption that the "topics" are fixed, namely they are the aspects which the reviewer mentions in their review. Note that, we can similarly derive update rules for an different topic model such as LDA (Blei et al., 2003), however prior work, (Lu et al., 2011), shows that LDA does not show superior performance over PLSA empirically for a number of tasks.

## 4 Features construction

Since we are interested in studying discriminative features for humorous and non-humorous reviews, we set up a classification problem to classify a review into either humorous or non-humorous. In classification problems the data plays a critical role; here the labels are obtained from the funny votes in our Yelp dataset, and we describe how we created the ground-truth in Section 5. Here in this section, we discuss the new features we can construct based on the proposed language model and estimated parameter values.

### 4.1 Incongruity features

A natural feature in our incongruity model is the estimated background weight, $\lambda$, since it indicates how much emphasis the reviewer puts in their review to describe the referenced aspects, we denote this feature by **A1**. Another feature is based on the relative weights for the referenced aspect's language models. There tends to be more 'surprise' in a review when the reviewer talks about multiple aspects equally, this is because the more topics the reviewer writes about the more intricate the review becomes. We use the entropy of the weights $H(R_d) = -\sum_{i=1}^{K_d} \pi_i \log \pi_i$ as another incongruity score and label this feature as **A2**.

### 4.2 Unexpectedness features

Humor often relies on introducing concepts which seem out of place to produce a comedic effect. Thus we want to measure this divergence from the references and the language expected in the reviews. Hence a natural measure is the KL-divergence measure the distance between the background language model and the aspect language models. We use the largest deviation, $\max_i\{D_{KL}(\theta^i||\theta^B)\}$ as feature **D2**. For this feature we tried different combinations such as a weighted average, but both features seemed to perform equally so we only describe one of them.

By considering the context of the references in the reviews we can distinguish which statements should be considered as humorous, thus we also use the relative weight for each aspect to measure unexpectedness. Formally we have $U_j = \pi_j D_{KL}(\theta^j||\theta^B)$, lastly we will denote $\max_i\{U_i\}$ these set of features as **U2**.

### 4.3 Baseline features from previous work

For completeness, we also include a description of all the baseline features used in our experiments; they represent the state of the art in defining features for this task. These features described below do not use any external text sources (leveraging external text sources is a novel aspect of our work), and they are more contextual and syntactical based features. We describe some of the most

promising features, which have previously shown to be useful in identifying humor in text.

**Context features:** Due to the popular success of context features by Mihalcea and Pulman (2007) we tried the following features content related features: **C1**: the uni-grams in the review.[2] **C2**: length of the review. **C3**: average word length. **C4**: the ratio of uppercase and lowercase characters to other characters in the review text.

**Alliteration:** Inspired by the success that Mihalcea and Strapparava (2006) had using the presence and absence of alliteration in jokes, we developed a similar feature for identifying funny reviews. We used CMU's pronunciation dictionary [3] to extract the pronunciation to identify alliteration chains, and rhyme chains in sentences. A chain is a consecutive set of words which have similar pronunciation, for example if the words words "scenery" and "greenery" are consecutive they would form a rhyme chain. Similarly, "vini, vidi, visa" also forms another chain this time an alliteration chain. We used the review's total number of alliteration chains and rhyme chains and denote it by **E1**. Note that there could be different lengths of chains, we experimented with some variations but they performed roughly the same, for simplicity we did not describe them here.

**Ambiguity:** Ambiguity in word interpretation has also been found to be useful in finding jokes. The reasoning is that if a word has multiple interpretation it is possible that the author intended another interpretation of the word instead of the more common one. We restricted the words in the reviews to only nouns and used Wordnet [4] to extract the synsets for these words. Then we counted the average number of synsets for each of these words, finally we took the mean score for all the words in the reviews. We call these features lexical ambiguity and denote it by **E2**.

## 5 Experimental Results

For our experiments we obtained the reviews from the Yelp Dataset Challenge[5], this dataset contains over 1.6 million reviews from 10 different cities. We also crawled reviews from Yelp in the Los Angeles area which is not included in the Yelp Dataset Challenge. This dataset was particularly interesting since the readers are able to vote whether a review is considered *cool*, *funny*, and/or *helpful*. It also allows the flexibility for the reviewers to write longer pieces of text to express their overall rating of a restaurant.

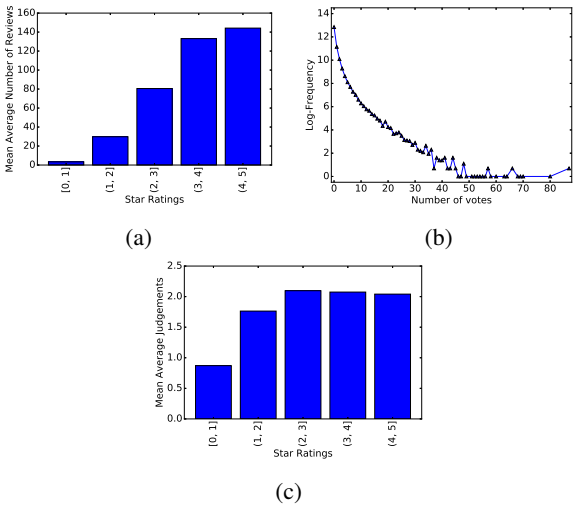


Figure 3: (a) Mean average number of reviews for restaurants falling in five different star rating ranges. (b) Log occurrences of funny votes per review. (c) Mean average voting judgements for restaurants in different star ratings.

### 5.1 Identifying Aspects in Reviews

We use recent advancements in Wikification, which aims to connect important entities and concepts in text to Wikipedia, it is also known as disambiguation to Wikipedia. In particular we use the work of Ratinov et al. (2011), in order to obtain the Wikipedia pages of the entities in the reviews, we call these *aspects* of the review. Using the Wikipedia description of the aspects we can compute the language models for each aspect. Using *mitlm*, the MIT language modeling toolkit by Hsu and Glass (2008), we apply Modified Kneser-Ney smoothing to obtain the language models from the Wikipedia pages obtained from review's aspects.

### 5.2 Preliminaries and Groundtruth Construction

In Figure 3 we give an account of data statistics based on a random sample of 500,000 reviews, focusing on the funny voting judgements and the star rating distributions. In Figure 3a, we notice that on average the highly rated restaurants tend to have more reviews. Since users would

[2] We also considered content-based features derived from PLSA topic weights, however the unigram features outperform these features, thus we exclude them for lack of space.

[3] www.speech.cs.cmu.edu/cgi-bin/cmudict

[4] http://wordnet.princeton.edu/

[5] http://www.yelp.com/dataset_challenge

| Features | | Classifiers | | |
|---|---|---|---|---|
| | | Naive Bayes | Perceptron | AdaBoost |
| Content Related Features | **C1** | 69.92 (0.545) | 57.62 (1.084) | 69.44 (0.485) |
| | **C2** | 51.33 (1.250) | 50.35 (0.763) | 50.56 (1.155) |
| | **C3** | 50.86 (0.812) | 50.00 (0.012) | 50.59 (1.122) |
| | **C4** | 53.85 (0.486) | 50.03 (0.172) | 51.41 (1.205) |
| Alliteration | **E1** | 50.81 (0.408) | 50.11 (0.301) | 50.28 (1.195) |
| Ambiguity | **E2** | 51.53 (0.677) | 50.39 (0.857) | 51.78 (1.533) |
| Incongruity | **A1** | 81.32 (0.974) | 81.32 (0.974) | 81.32 (0.974) |
| | **A2** | 83.68 (0.623) | 83.68 (0.623) | 83.68 (0.623) |
| Divergence Features | **D2** | 84.55 (0.550) | 83.68 (0.627) | 84.23 (0.561) |
| Unexpectedness | **U2** | 83.68 (0.627) | 83.68 (0.627) | 83.68 (0.627) |
| Combination features | **A1 + D2** | 84.55 (0.549) | 83.68 (0.627) | 84.35 (0.548) |
| | **A2 + D2** | 84.55 (0.549) | 84.00 (0.579) | 84.41 (0.496) |
| | **D2 + U2** | 84.55 (0.549) | 84.00 (0.579) | 84.40 (0.549) |
| | **A2 + D2 + U2** | 84.55 (0.550) | 83.89 (0.593) | 84.35 (0.590) |
| | **D2 + U2 + C1** | 78.28 (0.545) | 79.63 (0.534) | 83.18 (1.109) |
| | **A2 + D2 + C1** | 78.87 (0.546) | 82.68 (0.353) | 85.61 (0.900) |
| | **A1 + D2+U2+C1** | 78.62 (0.671) | 79.63 (0.528) | 85.77 (0.843) |
| | **A2 + D2+U2+C1** | 78.87 (0.546) | 81.60 (0.703) | 85.60 (0.968) |

Table 1: Classification accuracies, using 5-fold cross validation, the 95% confidence is given inside the parenthesis.

prefer to dine in a restaurant expecting to get a better overall experience, they create a feedback on the reviews for those highly rated restaurants. This "rich-get-richer" effect has been also been recently observed in other social networks (Su et al., 2016) and a more detailed analysis is out of scope of this paper. We observe that most of the reviews receive a low number of funny votes in Figure 3b, with $\mu = 0.55$, where $\mu$ is the average funny rating. Computing the restaurant's average funny votes, then taking the mean by the star ratings for each category range, see Figure 3c, which seems to be consistently increasing across the different star ratings. Note that this also includes the restaurants with zero funny votes, by excluding these we found that the ratings were more consistently stable on about 2.1 votes. Thus regardless of restaurant rating, the funny reviews distribution are stable on average. Considering the prevalence of noise in the voting process, we also analysed those reviews with more than one funny vote ($\mu = 3.90$), and with more than two votes ($\mu = 5.54$).

To construct our ground-truth data, we took all of the reviews at least five funny votes, which indicates the review was collectively funny, and considered those as humorous reviews, we considered all the reviews with zero funny votes as non-humorous reviews. We obtained 17,769 humorous reviews and 856,202 non-humorous, from which we sampled 12,000 reviews from each category, and another 5,000 reviews was left for a development dataset, to obtain a corpus with 34,000 reviews total. In total we collected 2,747 wikipedia pages with an average of about 247 sentences per page. In our work we focused on identifying distinguishing features and relative improvement in a balanced dataset and while the true distribution may be skewed, we leave the unbalance distribution study for future work.

Finally we use five-fold cross validation to evaluate all the methods. Due to the success of linear classifiers in text classification tasks we were interested in studying the Perceptron and Adaboost algorithms, we also used a Naive Bayes classifier which has been shown to perform relatively well in humor recognition tasks (Mihalcea and Strapparava, 2006). We used the Learning Based Java (LBJava) toolkit by Rizzolo and Roth (2010) for the implementation of all the classifiers and used their recommended parameter settings. For the Averaged Perceptron implementation, we used a learning rate of 0.05 and thickness of 5. In Adaboost, we choose BinaryMIRA as our weak

learner to do our boosting on. We also considered SparseWinnow and SparseConfidenceWeighted to be our weak learner as well, but the boosting performance for those two learners is marginal on the development set.[6] All experiments were run on an Intel Core i5-4200U CPU with 1.60GHz running Ubuntu.

### 5.3 Predicting Funny Reviews

We report the results of the features in Table 1. First we can compare the accuracies of the individual features. For the content related features we see that the best features is **C1**, which is consistent to what others have found in humor recognition research (Mihalcea and Pulman, 2007). The other content related features are based on some popular features for detecting useful reviews, however we notice that in the humor context it is not very effective. The performance of the contextual features could indicate that humor is not specific to a particular context and thus comparing different context between humorous and non-humorous text will not always work.

For the alliteration and ambiguity features which were reported to be very useful in short text, such as one-liners and on Twitter, are not as useful in detecting humours reviews. The reason is pretty clear since when writing a funny review, the reviewer does not worry about the limitation of text and thus their humor does not rush to a punch-line. Instead the reviewer is able to write a longer more creative piece, adhering to less structure. The features based on incongruity and unexpectedness, do really well in distinguishing the funny and non-funny reviews. For incongruity the best feature is **A2**, achieving about the same accuracy as unexpectedness features of about 83% accuracy.

The best feature was **D2** achieving an accuracy of around 84% accuracy. The features seem to be consistent over all of our classifiers. This indicates that incorporating background text sources to identify humor in reviews is crucial, and our features we can indirectly capture some common knowledge, e.g. prior knowledge. In particular it provides evidence that humor in online reviews can be better categorized as referential humor (Ritchie, 2009) rather then shorter jokes. The results also suggest that we can use these features to help predict the style of humorous text.

Exploring this would be an interesting venue for future work. When we combine our features for the classification task and find that the best combination is the incongruity features with the divergence features. We do not report the results for features **E1, E2** and other context features, **C2, C3, C4**, since their performance when combined with other features did not add to the accuracy of the more discriminant feature. The divergence feature **D2** plays a big role in the accuracy performance. This is in line with our hypothesis that the more uncommon language used the more it is possible to be for a humorous purpose.

It is interesting to see that AdaBoost performed the best out of all three classifiers achieving about 86% accuracy, especially when more features were added, the classifier was able to use this information for improvement. While Naive Bayes and the Perceptron algorithm did not make such improvement achieving about 85% accuracy.

### 5.4 Ranking Funny Reviews

From the data we noticed that funny reviews tend to be voted highly useful,in particular we noticed a correlation coefficient of 0.77. Although it would have been easy to use the useful votes as a feature to determine whether the review is funny/not funny, these scores are only available after people have been exposed to these reviews. To test how well the features worked when identifying helpful reviews, in a more realistic setting, we formulated a retrieval problem. Given a set of reviews, $\mathcal{D} = \{R_1, R_2, ..., R_m\}$ and relevant scores based on usefulness, $U = \{u_1, u_2, ..., u_m\}$, is it possible to develop a scoring function such that we rank the useful reviews higher? For this task we used the classification output of Naive Bayes, $P(\text{funny}|R_i)$ where $i$ is the current example under consideration, for our scoring function and trained with the best performing features in the original dataset. We used a with-held dataset crawled from restaurants in Yelp in the Los Angeles area containing about 1,360 reviews with 260 reviews labelled as helpful and the other reviews labelled as not helpful. To obtain the ground truth we used the useful votes in Yelp similar to how we constructed the funny labels, using a threshold of 5 votes minimum to be considered helpful. This experiment reveals two things about our features for detecting humorous reviews. First we see that the preci-

[6] Since our main goal is to understand the effectiveness of various features we did not further tune these parameters since they are presumably orthogonal to the question we study.

| K | Precision @ K |
|---|---|
| 1 | 1.00 |
| 10 | 0.50 |
| 25 | 0.48 |
| 50 | 0.44 |
| 100 | 0.45 |
| 200 | 0.54 |

Table 2: Precision of useful reviews.

sion is around 50%, see Table 2, this is more than two times better than random guess which is about 19% and second that our features can be used to filter out some useful reviews.

## 6 Related Work

Although there has been much work in the theory of humor by many linguists, philosophers and mathematicians (Paulos, 2008), the definition of humor is still a debated topic of research (Attardo, 1994). There have been many applications from computational humor research; for instance, creating embodied agents using humor, such as chat bots, which could allow for more engaging interactions and can impact many domains in education (Binsted et al., 2006). Existing work on computational humor research can typically be divided into humor recognition and humor generation.

In humor generation, some systems have successfully generated jokes and puns by exploiting some lexical structure in the pun/joke (Lessard and Levison, 1992; Manurung et al., 2008; McKay, 2002). The HAHAcronym project was able to take user inputs and output humorous acronyms and it achieves comical effects by exploiting incongruity (Stock and Strapparava, 2002). Work in automatic generation of humor is limited to particular domains, usually only generating short funny texts.

One of the earliest work on humor recognition in text data is the work of Mihalcea and Strapparave (2006), trying to identify *one-liners*, short sentences with a humorous effect. They frame the problems as a classification problem and develop surface features (alliteration, antonym, and adult slang) as well as context related features. They ultimately proposed that additional knowledge such as, irony, ambiguity, incongruity, and common sense knowledge among other things would be beneficial in humor recognition, but they do not further pursue these avenues. Although they are able to distinguish between humorous and non-humorous one liners, in longer of texts such as reviews it is not so clear that these features suffice. Instead we make use of the creative writing structure of the reviewers by looking at the referenced entities in their reviews.

Although verbal irony can be humorous, and an active topic of research (Wallace, 2013), it is often defined as the "opposite to what the speaker means", and combining features for identifying both humor and irony has been studied (see, e.g., Reyes et al. (2012)). In the work by Reyes et al. (2012), the authors defined the unexpectedness feature as semantic relatedness of concepts in Wordnet and assuming that the less the semantic relatedness of concepts the funnier the text. In our work we use a similar definition but applying it to the "topical" relatedness of the referenced aspects and the background language model. The authors demonstrate that irony and humor share some similar characteristics and thus we can potentially use similar features to discriminate them. There has been some early work in identifying humor features in web comments (Reyes et al., 2010), in these comments the users are able to create humor through dialogue thus making the problem more complex. More recently there was a workshop in SemEval-2017 [7], which focus is on identifying humorous tweets which are related, typically as a punchline, to a particular hashtag.

Kiddon and Brun (2011) aimed to understand "That's what she said" (TWSS) jokes, which they classify as double entendres. They frame the problem as metaphor identification and notice that the source nouns are euphemisms for sexually explicit nouns. They also make use of the common structure of the TWSS jokes to the erotic domains to improve 12% in precision over word-based features. In our work we try to explicitly model the incongruity of the reviewer, by doing so we are able to distinguish the separate language used by the user when introducing humorous concepts. Recently there has been work in consumer research, to identify the prevalence of humor in social media (McGraw et al., 2015). The main focus was to examine the benign violation theory, which "suggest that things are humorous when people perceive something as wrong yet okay". One of their finding suggests that humor is more prevalent in complaints than in praise, thus motivating

[7] http://alt.qcri.org/semeval2017/task6/

the usage of automatic humor identification methods for restaurants regardless of its popularity.

While there is a breadth of work in identifying helpful reviews and opinion spam in reviews (Jindal and Liu, 2008) as well as deceptive opinion spam (Ott et al., 2011), and synthetic opinion spam (Sun et al., 2013); we show that humour can also be used to identify helpful reviews.

## 7 Conclusion

We have studied humorous text identification in a novel setting involving online reviews. This task has not been studied in the previous work and is different than detecting humorous jokes or one-liners, this allows for creative and expressive writing since the reviewer is not limited in text. In this problem we cannot directly apply the ideas that others have developed in order to identify the humorous reviews. Instead features that are based on the theory of incongruity are shown to outperform previous features and are effective in the classification task. Our model introduces a novel and way to incorporate external text sources for humor identification task, and which can be applied to any natural language provided there is a reference database, i.e. news articles or Wikipedia pages, in that language. We also show that the features developed can also be used to identify helpful reviews. This is very useful in the online review setting since there tends to be a cumulative advantage, that is the "rich get richer" effect which limits the exposure that the users get to other helpful reviews. Thus identifying these types of review early can potentially diversify the types of reviews that the users read.

Although we used a background language model on the entire corpus to capture a sense of expectation, there could be other ways to do this. For example, we could develop neural network embeddings to capture the entities descriptions in the reviews. Another direction would be to use topic models and see whether reviewers are more inclined to compare different types of references when talking about certain aspects of restaurants or other products. A different approach to identifying helpful reviews would be to create entertaining and informative summaries.

## Acknowledgments

The first author was supported by the University of Illinois, Urbana-Champaign College of Engineering's Support for Underrepresented Groups in Engineering (SURGE) Fellowship and the Graduate College's Graduate Distinguished Fellowship.

## References

Salvatore Attardo. 1994. *Linguistic theories of humor*, volume 1. Walter de Gruyter.

Kim Binsted, Anton Nijholt, Oliviero Stock, Carlo Strapparava, G Ritchie, R Manurung, H Pain, Annalu Waller, and D O'Mara. 2006. Computational humor. *Intelligent Systems, IEEE*, 21(2):59–69.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc.

Bo-June Hsu and James Glass. 2008. Iterative language model estimation: efficient data structure & algorithms. In *Proceedings of Interspeech*, volume 8, pages 1–4.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 219–230. ACM.

Chloe Kiddon and Yuriy Brun. 2011. That's what she said: double entendre identification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 89–94. Association for Computational Linguistics.

Greg Lessard and Michael Levison. 1992. Computational modelling of linguistic humour: Tom swifties. In *ALLC/ACH Joint Annual Conference, Oxford*, pages 175–178.

Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. 2011. Investigating task performance of probabilistic topic models: an empirical study of plsa and lda. *Information Retrieval*, 14(2):178–203.

Ruli Manurung, Graeme Ritchie, Helen Pain, Annalu Waller, Dave O'Mara, and Rolf Black. 2008. The construction of a pun generator for language skills development. *Applied Artificial Intelligence*, 22(9):841–869.

A Peter McGraw, Caleb Warren, and Christina Kan. 2015. Humorous complaining. *Journal of Consumer Research*, 41(5):1153–1171.

Justin McKay. 2002. Generation of idiom-based witticisms to aid second language learning. *Stock et al.(2002)*, pages 77–87.

Rada Mihalcea and Stephen Pulman. 2007. Characterizing humour: An exploration of features in humorous texts. In *Computational Linguistics and Intelligent Text Processing*, pages 337–347. Springer.

Rada Mihalcea and Carlo Strapparava. 2006. Learning to laugh (automatically): Computational models for humor recognition. *Computational Intelligence*, 22(2):126–142.

Anton Nijholt. 2014. Towards humor modelling and facilitation in smart environments. *Advances in Affective and Pleasurable Design*, pages 260–269.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics.

John Allen Paulos. 2008. *Mathematics and humor: A study of the logic of humor*. University of Chicago Press.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics.

Antonio Reyes, Martin Potthast, Paolo Rosso, and Benno Stein. 2010. Evaluating humour features on web comments. In *LREC*.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.

Graeme Ritchie. 2009. Can computers create humor? *AI Magazine*, 30(3):71.

Nick Rizzolo and Dan Roth. 2010. Learning based java for rapid development of nlp systems. In *LREC*.

Oliviero Stock and Carlo Strapparava. 2002. Hahacronym: Humorous agents for humorous acronyms. *Stock, Oliviero, Carlo Strapparava, and Anton Nijholt. Eds*, pages 125–135.

Jessica Su, Aneesh Sharma, and Sharad Goel. 2016. The effect of recommendations on network structure. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1157–1167. International World Wide Web Conferences Steering Committee.

Huan Sun, Alex Morales, and Xifeng Yan. 2013. Synthetic review spamming and defense. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1088–1096. ACM.

Byron C Wallace. 2013. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, pages 1–17.