ClearView: Data Cleaning for Online Review Mining

Amanda Minnich*, Noor Abu-El-Rub*, Maya Gokhale†, Ronald Minnich‡ and Abdullah Mueen*
*University of New Mexico, aminnich@cs.unm.edu †Lawrence Livermore National Laboratory ‡Google Inc.

Abstract—**How can we automatically clean and curate online reviews to better mine them for knowledge discovery? Typical online reviews are full of noise and abnormalities, hindering semantic analysis and leading to a poor customer experience. Abnormalities include non-standard characters, unstructured punctuation, different/multiple languages, and misspelled words. Worse still, people will leave "junk" text, which is either completely nonsensical, spam, or fraudulent. In this paper, we describe three types of noisy and abnormal reviews, discuss methods to detect and filter them, and, finally, show the effectiveness of our cleaning process by improving the overall distributional characteristics of review datasets.**

I. Introduction

Online reviews form a unique source of unbiased information about products and services for consumers, manufacturers, distributors, sellers, packagers, and shippers. Online reviews have been analyzed and mined for over a decade to extract useful knowledge such as opportunities to improve service [1] and business planning [2]. Hosting sites have also evolved to collect various forms of information from mass consumers, such as star-rating, review-text, and helpfulness-rating.

Online reviews are typically noisy and contain many types of abnormalities. In Figure 1, we show a set of noisy reviews that contain no information whatsoever. The top row shows a set of un-intelligible reviews where meaningless sequences of English characters are written as reviews in the Google Play site. It also shows a review in Russian, which, while valid content, is not meaningful to English-speaking audiences. The second row shows repeated text, inconsistently rated reviews in which the sentiment in the text is the opposite of the rating, and an app containing promotional spam and non-Unicode characters masquerading as standard English characters.

Noisy reviews appear in almost every kind of hosting system including tourism, e-commerce[3], real-estate[4], and mobile apps[5]. Surprisingly, there is no formal cleaning process for online reviews that can be generically applied before presenting them to consumers or mining them for knowledge discovery. The absence of clean reviews may lead to *flawed* marketing strategies[6] and *lack of trust* in customers[7].

In this paper, we discuss various types of abnormalities that exist in different review sites and develop filtering techniques to clean them. Our methods use natural language processing (NLP) parsers and classifiers targeting three kinds of noise. To evaluate the efficacy of our cleaning process, we look at the distributions of features that have been shown to identify abnormally behaving hotels[7]. We show that our cleaning technique standardizes these feature distributions and improves the quality of the reviews for knowledge discovery processes.

II. Background

Online reviews for products and services are written by consumers with the intention of helping people make informed decision when selecting a purchase. Natural variations in reviews occur because of the diverse backgrounds of writers. In addition, fake reviews, incentivized reviews, and revengeful reviews introduce unique anomalous variations.

We identify three major types of variations that appear in almost all review datasets. We describe the categories below.

a) Syntactic Noise: Reviewers often make syntactic errors. For example, misspelled words, nonsense words, and slang are commonly used in the review space. Another example is using non-standard characters to write English words, with the intention of defeating content-based filtering.

b) Semantic Noise: Reviewers often write incorrect sentences that are not intelligible. Such reviews can be the result of automated text completion during typing, or can simply be due to the negligence of the reviewer.

c) Rating Noise: Sometimes the review text consists of well-formed, meaningful sentences, but the star-rating accompanying the text does not match the text sentiment. Such reviews are confusing for the reader and not trustworthy.

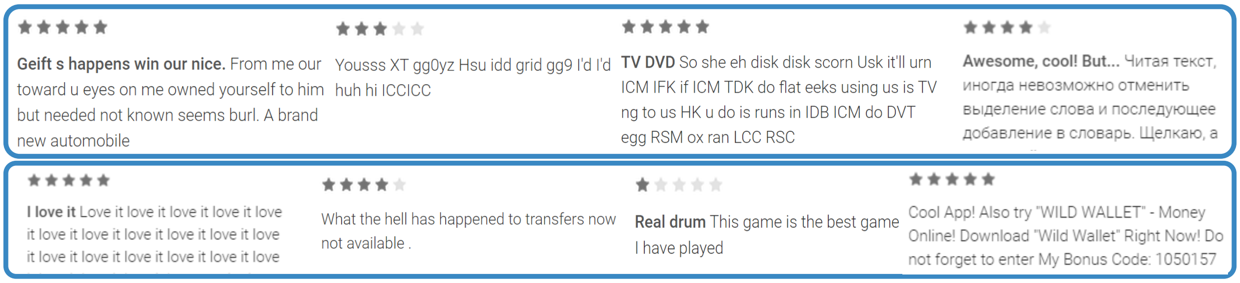
We develop detection and filtering techniques for each type of noise.

III. Syntactic Cleaning

We perform two types of syntactic cleaning: character-level cleaning and word-level cleaning.

a) Character-level cleaning: When examining samples of Google Play reviews, we find many reviews that contain non-Unicode characters masquerading as normal text. We believe that this is done in an attempt to evade keyword filters which are used to detect spammers. For example, in the reviews for the app Key Ring: Cards Coupon & Sales, a user left the review shown in Figure 1 bottom right. Although it looks normal in the Google Play site, when the text is processed by the LaTeX compiler it reads: "`Cool ! Also try "WILD WLL" - Mon Online! Downld "Wild Wllt" Right Now! Do not forget to ntr  nus d: 1050157`." This is because the review contains many words that are hiding non-Unicode characters. These are words that a filter may be looking for because they indicate a spamming behavior: *money, wallet, code, bonus, enter,* and *app*. This user has left 11 other reviews with this same signature. We catch such reviews by checking if the characters in the review are printable, which is defined as digits, letters, punctuation, and whitespace. If any characters are not printable (including non-English characters), we view

IEEE/ACM ASONAM 2016, August 18-21, 2016, San Francisco, CA, USA



★★★★★ **Geift s happens win our nice.** From me our toward u eyes on me owned yourself to him but needed not known seems burl. A brand new automobile

★★★☆☆ Yousss XT gg0yz Hsu idd grid gg9 I'd I'd huh hi ICCICC

★★★★★ **TV DVD** So she eh disk disk scorn Usk it'll urn ICM IFK if ICM TDK do flat eeks using us is TV ng to us HK u do is runs in IDB ICM do DVT egg RSM ox ran LCC RSC

★★★★☆ **Awesome, cool! But...** Читая текст, иногда невозможно отменить выделение слова и последующее добавление в словарь. Щелкаю, а

★★★★★ **I love it** Love it love it love it love it love it love it love it love it love it love it love it love it love it love it love it love

★★★★☆ What the hell has happened to transfers now not available .

★☆☆☆☆ **Real drum** This game is the best game I have played

★★★★★ Cool App! Also try "WILD WALLET" - Money Online! Download "Wild Wallet" Right Now! Do not forget to enter My Bonus Code: 1050157

Fig. 1: (top-row) Unintelligible reviews and a review in Russian. (bottom-row) Repeated text, positive and negative twisters, and non-Unicode text.

the review as not informative and filter out the review. Such a strict filter removes a large number of syntactically unusable reviews.

b) Word-level cleaning: At the word level, we check for black-listed keywords, abnormal repetitions, and meaningless words. There are only a few valid reasons to have a long sequence of alpha-numerals in a review. Reviewers may identify the price, model, and feature of a product in their review to describe their experience. However, such alpha numerals are also a sign of abusive reviews. For example, personal ID or code for referral rewards and promotions are advertised via reviews, e.g. `Please enter my code 8zl12j to help us both get rewards!`. We clean such reviews based on a set of black-listed keywords that most often represent abusive behavior, for example, `promo-code`, `invitation code`, and `http`, among many others.

We also check for abnormal repetition of one or few words. Some reviewers just copy and paste words such as `good`, `great`, and `nice` many times. Such reviews contain very little information compared to the length of the review.

Another type of useless review we see is reviews comprised mainly of nonsense words. There are many ways meaningless words are written. Sometimes authors use English letters to write another language, which, while obviously containing meaning, does not contain meaning to an English-only speaker. Authors also invent spellings, such as the use of `nyc` to represent the word `nice`, the use of `gr8` to represent the word `great`, and so on. Reviews that primarily consist of such words contain little-to-no usable information about the product, especially from a text analysis perspective. Using the *Enchant* library from Python, we identify the percentage of words in the sentence that are in its large corpus of English words. Since reviews often contain misspelled words or colloquial words not in this corpus, we set a threshold for our filter: if less than a threshold of the words in a review are in this corpus, then the review is filtered out. This threshold is a tunable parameter that can be set depending on the end usage of the text: sentiment classification needs a high threshold, compared to robust clustering which requires a lower threshold. For our experiments, we use a threshold of 90% for TripAdvisor and 50% for Google Play. This is because the reviews in the Google Play dataset have on average more misspelled words than those in the TripAdvisor dataset.

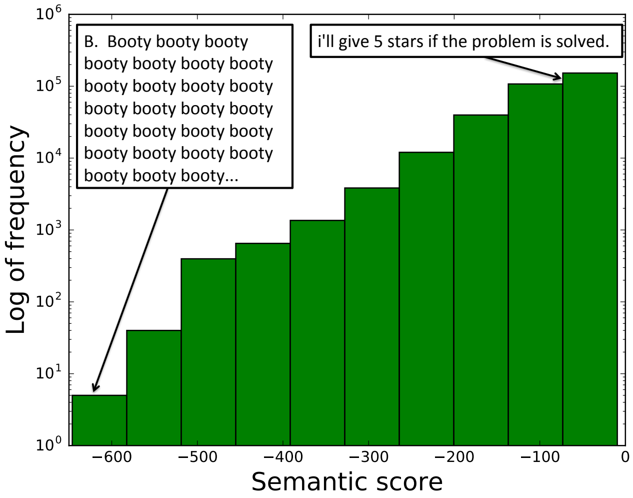


Fig. 2: The distribution of semantic scores.

IV. Semantic Cleaning

To identify nonsensical reviews, we analyze the semantic structure of the sentences in a review. We use the Stanford CoreNLP Parser[8] to label words with part-of-speech tags and to parse sentences into tree structures. We use the confidence score of the parser as a measure of the semantic correctness of a sentence. The lower the score, the less likely that the sentence is valid. For example, for the first review in Figure 1, the second sentence contains mostly valid words but does not make any sense: "`From me our toward u eyes on me owned yourself to him but needed not known seems burl.`" The score generated for this sentence is -174.3. If we compare this with a sentence found in many reviews: "`I highly recommend it.`", we get a much higher score of -33.9 because the tree it forms is highly probable. In Figure 2, we show the distribution of scores of the sentences of all reviews. Note the log scale in the y-axis.

V. Rating Cleaning

The second row of Figure 1 shows reviews in which the sentiment of the text does not match the rating.

We develop a method to clean such reviews by using an ensemble of sentiment classifiers. We use the sentiment classifiers to iteratively label the sentiment of the reviews to obtain the maximal agreement with the user-given ratings. When a writer and the classifiers agree on the sentiment, the review is more likely to be high quality. However, when they mismatch, it can be either writer error or classifier error. For cleaning purposes, precision is more important than recall rate. Hence, we remain strict on absolute consensus.

The process starts with a classifier trained on the Standard Tree Bank[9]. This tree bank contains more than 11,000 sentences from movie reviews on RottenTomatoes.com. 215,000 individual phrases of these sentences were manually labeled from "Very Negative" to "Very Positive." We train a classifier on these data and then generate sentiment labels for our reviews using this classifier. We then form a training set of reviews whose sentiment scores match the user-given ratings. Reviews whose sentiment scores do not match their ratings form the new test set. To improve the classifier, we then train the current classifier using the new training set and evaluate its performance on the new test set. We continue this process iteratively, adding matched reviews to the training set at every iteration. When the process converges, we have an extremely *overfitted* sentiment classifier that has almost memorized the noisy set, excluding the reviews remaining in the test set. The leftover test set is more likely to contain erroneous cases, as even an overfitted classifier has failed to classify its members correctly.

We overfit two classifiers using the above process and filter out any reviews that have a single disagreement between the sentiment labels and the user-given rating.

VI. Experimental Evaluation

a) Data Description: Two datasets were used for the experiments described in this section. The first dataset consists of reviews from TripAdvisor.com. We collected all the reviews and associated information for almost all of the US hotels on this site. For the second dataset, we collected reviews and their associated information from nearly all of the apps in the Google Play Store.

b) Classifiers: The first classifier, Stanford's Sentiment Classifier in the CoreNLP suite[10][11], is widely acknowledged to be a top-performing sentiment classifier. This recursive neural tensor net classifier stores sentences in a parsed tree format, rather than the typical bag-of-words approach. This allows the classifier to take the sentence structure into account when classifying sentiment. While this approach is quite accurate, it is also quite slow, requiring weeks to train and classify our full datasets.

The second classifier we use is based on [12]. This classifier is a simple Naive Bayes classifier that is smart enough to handle both negation and double negation, and adds a negated form of the word to the opposing class's word bank (e.g. if "good" occurs in a review with a positive label, it adds "not_good" to the negative class). This algorithm also uses bigrams and trigrams in addition to unigrams to further improve performance. Lastly, low-occurring words are pruned at the end of every training round.

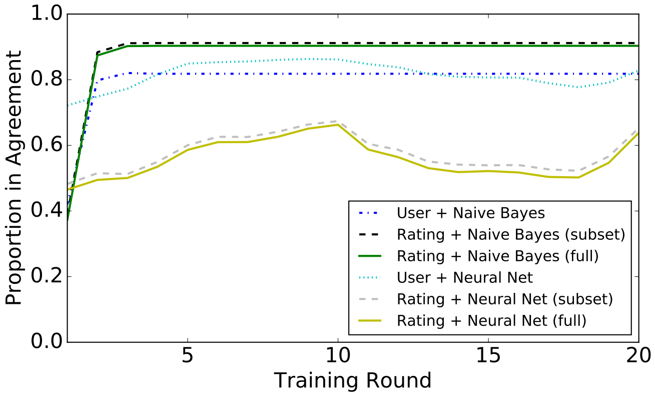


Fig. 3: Iterative training results for TripAdvisor

We added in the capability for 5-class classification and iterative training to both of these classifiers, as well as input and output pipelines including customizable performance visualizations. Note that both of the classifiers rate individual sentences; we aggregate the ratings to calculate one score for a review that may contain many sentences based on the sentence scores normalized by the sentence lengths.

A. Results

	Full Dataset	Percent Filtered	Agreement Before Filtering	Agreement After Filtering
TripAdvisor	3,167,036	30%	17.7%	59.9%
Google Play	21,112,036	70%	18.7%	36.9%

TABLE I: Results of filtering process.

We run two datasets through the ClearView pipeline and filter out 874,275 malformed reviews from the TripAdvisor dataset and 10,357,430 reviews from the Google Play dataset.

We then perform semantic filtering and rating validation for a randomly selected subset of 60,000 reviews from each of our filtered datasets. We also randomly select 10,000 reviews from each subset to evaluate in our user study.

For our user study, we use the Amazon Mechanical Turk Marketplace to have readers evaluate the sentiment of 10,000 reviews from each dataset. We require 3 different users to score each review and average their scores. They can pick a score from 1-5, just like the ratings on the review websites. We then compare those to the review's rating and the sentiment classifier rating over the 20 rounds. This user study allows us to validate the performance of the sentiment classifier. Table I shows our results.

Figure 3 shows the convergence of the sentiment classifiers over 20 training rounds for the TripAdvisor dataset. Note that the Naive Bayes classifier starts out with lower accuracy but surpasses the performance of the neural net classifier after a few training iterations. Furthermore, the entire 20 rounds ran in a few minutes for this Naive Bayes classifier, as opposed to days for the neural net. We find that Google Play reviews are much more difficult to properly classify. They contain many more colloquialisms than the TripAdvisor reviews and have less consistent structure.

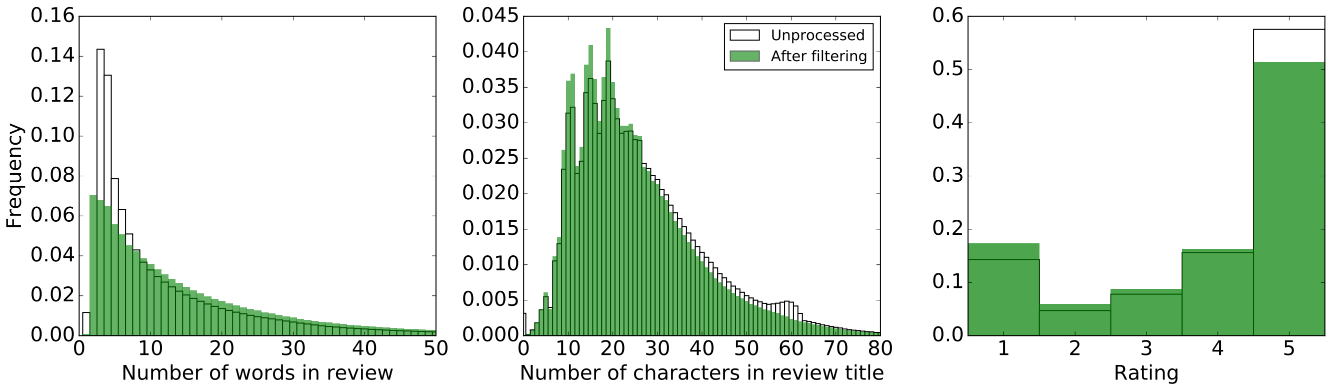


Fig. 4: (left) Distribution of the number of words in Google Play reviews before and after filtering. (middle) Distribution of the number of characters in TripAdvisor titles before and after filtering. (right) Rating distributions of Google Play reviews before and after filtering

Review	Writer	Turker	NLP
I cant log in...i try to update the game but i still cant log in. please fix this bug.	5	1	1.67
beautiful view of the lake and really enjoyed sitting on our private balcony. free wifi and no problems...	1	4	4.19

TABLE II: Reviews identified as inconsistent through iterative sentiment classification. More examples available at[13].

Figure 4 shows feature distributions before and after filtering. These features were shown in [7] to be effective in characterizing anomalous hotels on three different online review websites. In all three subfigures, filtering either reduces the skewness of the distribution or removes an abnormal bump, even though there was no specific filtering done on title/ review length or rating.

VII. Sensitivity and Scalability

a) Sensitivity: In semantic cleaning, we have several thresholds that can be set depending on the needs of the user. The percentage of non-printable characters, the percentage of misspelled words, and the minimum semantic score can all be set separately, allowing for customization of this pipeline.

b) Scalability: The Stanford sentiment classifier is unusable for the large set of reviews we consider in this work. For example, our training data consisted of a collection of 1.75GB for the TripAdvisor dataset and 6.62GB for the Google Play dataset. Processing such data using a single classification process would take on the order of days.

To calibrate the runtime of a single iteration, we run the entire workload on a single "fat" server with 2TB memory, 120 hyperthreaded Xeon E7-4870 1.2GHz cores. Both input and output files were accessed in tmpfs[14], making all file I/O in-memory. We use a taskbag model in which a pool of compute server processes (i.e. workers) iteratively fetches and executes work from the taskbag. The workers invoke the Stanford NLP sentiment classifier with preset parameters and input/output file names tagged with the id of the fetched task. The whole job is self load-balancing and the task dispatch algorithm was written in 200 lines of *Go* code. One iteration to classify the review set takes 30 hours.

VIII. Related Work

Current research on online reviews can broadly be classified based on methodology and application. Researchers have developed topic discovery [15] and sentiment classification [16] methods from review text. Connecting users, products and reviews in an information network is another promising method to analyze reviews [17]. Researchers have applied these methods for fraud detection [18] and recommending products [15]. Our proposed work is the first attempt to filter and normalize useless reviews.

References

[1] "Using Online Reviews to Improve Your Business." https://www.reputationloop.com/online-reviews-improve-your-business/.

[2] "Review your business performance." http://www.infoentrepreneurs.org/en/guides/review-your-business-performance.

[3] "TripAdvisor." https://www.tripadvisor.com.

[4] "Zillow." http://www.zillow.com.

[5] "Google Play Store." https://play.google.com/store/apps.

[6] "83% Of SEOs Believe Focusing On Reviews Delivers Good ROI." http://searchengineland.com/local-search-marketers-83-seos-believe-focusing-reviews-delivers-good-roi-220077.

[7] A. J. Minnich, N. Chavoshi, A. Mueen, S. Luan, and M. Faloutsos, "TrueView: Harnessing the Power of Multiple Review Sites," in *WWW'15*, pp. 787–797, 2015.

[8] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *ACL'03*, pp. 423–430, 2003.

[9] R. Socher *et al.*, "Parsing With Compositional Vector Grammars," in *EMNLP'13*, 2013.

[10] C. D. Manning *et al.*, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55–60, 2014.

[11] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP 2013*, pp. 1631–1642, 2013.

[12] V. Narayanan, I. Arora, and A. Bhatia, "Fast and accurate sentiment classification using an enhanced naive bayes model," *CoRR*, vol. abs/1305.6143, 2013.

[13] "ClearView." http://cs.unm.edu/~aminnich/clearview.

[14] P. Snyder, "tmpfs: A virtual memory file system," in *Proceedings of the Autumn 1990 EUUG Conference*, pp. 241–248, 1990.

[15] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *RecSys'13*, pp. 165–172, 2013.

[16] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *EMNLP'02*, pp. 79–86.

[17] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *ICWSM'13*, pp. 2–11, 2013.

[18] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," in *WWW'12*, pp. 191–200, 2012.