Identifying Health-Violating Restaurants with Online Reviews*

Mikel Joaristi
Computer Science Dep.
Boise State University
Boise, ID, USA
mikeljoaristi@
boisestate.edu

Edoardo Serra
Computer Science Dep.
Boise State University
Boise, ID, USA
edoardoserra@
boisestate.edu

Francesca Spezzano
Computer Science Dep.
Boise State University
Boise, ID, USA
francescaspezzano@
boisestate.edu

ABSTRACT

Nowadays, detecting health-violating restaurants is a serious problem due to the limited number of health inspectors in a city as compared to the number of restaurants. Rarely inspectors are helped by formal complains, but many complaints are reported as reviews on social media such as Yelp.

In this paper we propose new predictors to detect health-violating restaurants based on restaurant sub-area location, Yelp reviews content, and Yelp users behavior. The resulting method outperforms past work, with a percentage of improvement in Cohen's kappa and Matthews correlation coefficient of at least 16%. In addition, we define a new method that directly evaluates the benefit of a classifier on the ability of an inspector in detecting health-violating restaurants. We show that our classification method really improves the ability of the inspector and outperforms previous solutions.

1. INTRODUCTION

Foodborne illness is prevented by the Department of Public Health by periodically inspecting the health conditions of restaurants all over the country. However, annual inspections are not always able to check all the restaurants in a city because of a limited number of inspectors as compared to the number of restaurants and not so many formal complaints are available to help them to prioritize restaurant to inspect.

In this paper we study the problem of predicting restaurant health inspection outcomes by leveraging restaurant and inspection metadata and information extracted from social media. Our goal is to make use of public available information to help government health inspectors in carrying out their job more efficiently. Moreover, we want to provide inspectors with a system that helps to prioritize what business to inspect first based on the health risk they could potentially pose in the near future.

We use restaurant on-line reviews and information about the users writing these reviews extracted from Yelp, a business recommendation social media where the users can review a business and give advice to other users. These reviews often contain informal complaints about a visited restaurant.

The prediction problem turns out to be challenging because of the following reasons. First of all, datasets are often unbalanced w.r.t. the class of interest (restaurants not passing the inspection). As a consequence, high accuracy results reported in past work often coincide with the percentage of passed inspections. Second, public available data are noisy. There are cases of finding very good reviews for a restaurant that did not pass health inspection and vice versa. These are usually fake reviews, or the content is not related with health problems.

Moreover, previously proposed approaches usually concentrate on one specific city [9] or the performances are shown on small datasets [17]. In addition, many evaluation metrics such as Cohen's kappa and Matthews correlation coefficient (MCC) taking into account the data unbalance, or the class of interest (e.g. recall of not passed inspections) are not shown. Thus, it is very difficult to have a clear picture of how these methods perform.

The contributions of this paper are the following:

- We used public available information to collect a new dataset for the city of Las Vegas containing (i) 17K restaurant health inspection records and metadata, and (ii) 13K Yelp reviews and information for 65K users.
- We evaluate existing methods according to many performance measures and on two different datasets: our Las Vegas dataset and an existing dataset for the city of Seattle.
- We define a novel set of features for the prediction task based on restaurant sub-area location, Yelp reviews content, and Yelp users behavior.
- We show that our approach significantly improves over past work in terms of Cohen's kappa, Matthews correlation coefficient, and recall for the class of not passed inspections, while it remains comparable in accuracy and precision.
- We further study the effectiveness of the classifier in helping public inspectors in finding health-violating restaurants. We propose an algorithm simulating the inspector job of choosing which restaurants to inspect. We show that using a classifier is always better than conducting inspections without any guidance, and that our classifier is more helpful in retrieving unhealthy restaurants in comparison with

*Extended Abstract. A full version of this paper to appear in Proceedings of the 2016 IEEE-ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2016).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

BigNet 2016 Indianapolis, USA, October 24-28, 2016

© 2016 ACM. ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

previous work.

2. RELATED WORK

Recently, there has been huge attention towards helping public health surveillance by mining social media [6]. These works deal with monitoring and tracking flu activity and predicting influenza levels [11, 8, 12, 4, 1], allergy surveillance [13], and tracking foodborne illness [5]. However, few works have studied the relation of social media reviews and the outcome of health inspections.

Sadilek et al. [16] studied the relation of user comments in Twitter and the outcome of restaurant health inspections. They used users' geo-location data to link the users to restaurants they might have been to, and then analyze if there is any trace of health related complaints. They found a correlation of 0.3 with the official inspection data. However, many relevant comments may not be considered as they could have been written hours later a user visited the restaurants and in a different geo-location.

Kang et al. [9] are the first who studied the problem of predicting restaurant health inspection outcomes from Yelp reviews. They applied basic text analysis (unigrams and bigrams) together with other features such as review count, non-positive review count, cuisine type, zip code, average review rating, and inspection history (previous inspection score and average of past inspection scores). They classified passed from not passed inspections for the city of Seattle by using a support vector machine. They defined an inspection as passed if the inspection grade is above a threshold $\theta \in \{0, 10, 20, 30, 40, 50\}$. In their paper, the best accuracy result is obtained with $\theta = 50$, but in this case the resulting dataset, that we called Seattle_50, is highly unbalanced (see Figure 1). Consequently, even tough they have a very high accuracy of 0.97% (see Table 2), their Cohen's kappa and MCC are very poor (0.04 and 0.046, resp.). Then, the high accuracy is mainly due to the fact that the dataset is unbalanced. In this paper we compare our approach with the one proposed by Kang et al. on more than one dataset and show that we improve Cohen's kappa, MCC, and recall of not passed inspections.

Recently, Schomberg et al. [17] conducted a pilot study detecting unhealthy restaurants in the cities of San Francisco and New York. They used a very small dataset (755 restaurant inspections for NY and less than 1543 for SF) and a very simple model mainly using a bag of words with few keywords on Yelp reviews. We did not compare with this approach as the list of all unigrams is already considered in [9], and our model automatically extracts topics dealing with health related problems from reviews.

3. DATA

In this paper, we use three datasets dealing with restaurants in the cities of Las Vegas and Seattle which integrate health inspections data with restaurant reviews from Yelp.

The first one is a *new* dataset we built for the city of Las Vegas. This dataset, we refer to as the Las Vegas dataset, is composed of two parts:[1]

(i) Historical records of all restaurant health inspections carried out in the city of Las Vegas from 1990 to Jan. 2016. These records are collected from the Southern Nevada Health District (SNHD) web page[2] and contain information about restaurant name, location, category, inspection date, grade (*passed* or *not passed*) assigned during the inspection, demerits, inspection type and the list of all violations of that particular inspection together with the severity of each violation[3];

(i) Las Vegas restaurant metadata and reviews, and users information extracted from Yelp[4]. For each restaurant we extracted name, location, grades, and a list of reviews (from 2005 to 2015). Each review data contains text, grade, date, and votes from other users. Finally for each Yelp user who made a review we extracted name, friends list, and compliment information from other users.

The Las Vegas dataset contains more than 1,200 businesses, 17,000 inspections, 13,000 reviews and 65,000 Yelp users.

In addition, we used the dataset from [9] built for the city of Seattle which has been downloaded from their website[5]. This dataset, which we call the Seattle dataset, contains the same informations as our Las Vegas dataset, from 2006 to 2013. This dataset contains more than 1,200 businesses, 9,000 inspections, 150,000 reviews (from 2004 to 2013), and 43,000 Yelp users. Kang et al. ignored the inspection passed/not passed labels[6] and considered an inspection passed if the inspection grade score more than or equal to 50 points. Thus, for comparison, we also used this version of the Seattle dataset which we call the Seattle_50 dataset.

As shown in Figure 1, none of the three datasets we used in this paper is balanced. In particular, the Seattle_50 dataset is extremely unbalanced, while the Seattle dataset is slightly unbalanced (about 60% of the restaurants passed the inspection, and 40% not).

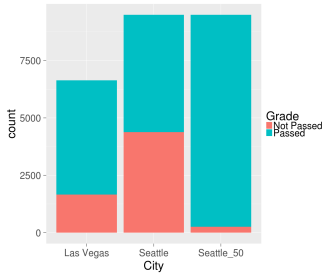


Figure 1: A visualization of the unbalance of all the three datasets: Las Vegas, Seattle, and Seattle_50.

4. RESTAURANTS INSPECTION OUTCOME PREDICTORS

In this section, we propose new features for the task of predicting restaurant health inspection outcomes. These features involve (i) restaurant location, (ii) Yelp restaurant reviews, and (iii) Yelp users behavior. Some of these fea-

[1]The Las Vegas dataset is available at https://sites.google.com/site/YelpRestaurantInspections/

[2]SNHD web page: http://southernnevadahealthdistrict.org

[3]http://southernnevadahealthdistrict.org/food-operations/restaurant-inspection.php

[4]We used the dataset available at https://www.yelp.com/dataset_challenge

[5]http://www3.cs.stonybrook.edu/~junkang/hygiene/

[6]These labels can be found at https://data.kingcounty.gov/Health/Food-Establishment-Inspection-Data/f29f-zza5

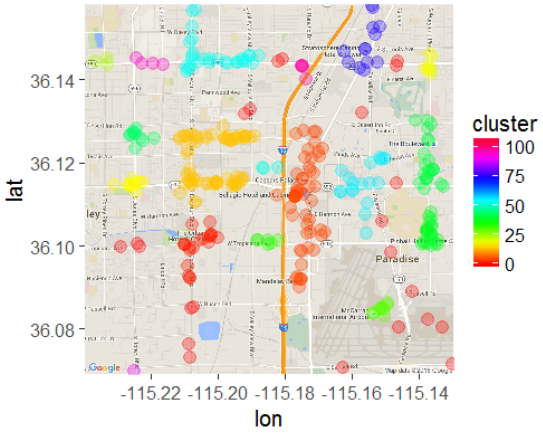


Figure 2: The different area clusters. In the middle (orange), Las Vegas Strip area.

tures are improvements upon the features defined by Kang et al. [9].

4.1 Restaurant Location-based Features

The idea behind considering restaurant location to predict the health inspection outcome is that restaurants in a rich area may have more selected customers and higher quality service and be more expensive than restaurants in a poor or conflictive area. This implies that restaurant in a good area will be more careful with health hazards and will pass the inspections more often.

Kang et al. [9] used the *zip code* of the restaurants to define its location. However, the size of the area defined by the zip code may be too big to define a meaningful location. Then, we propose to divide the whole area defined by the zip code into smaller sub-areas, and to locate each restaurant in one of these sub-areas. We used the DBSCAN [7] clustering algorithm to define zip code sub-areas given restaurants latitude and longitude. There are two main reasons why we choose a density based clustering algorithm instead of a distance based one. The first reason is that with a density based clustering algorithm there is no need to select the number of clusters to find. The second and main reason is that a density based algorithm can find clusters of arbitrary shape, in contrast, a distance based algorithm is bias towards circular shapes. In our particular problem this allows us to find shapes like streets and individuate, for instance, two parallel streets as two different areas. Figure 2 shows the sub-areas individuated by DBSCAN in Las Vegas. As we can see, the restaurants in the Las Vegas Strip (in orange) can be easily differentiated from the other nearby areas.

4.2 Yelp Restaurant Reviews-based Features

The main idea oaf our paper is to use information extracted from Yelp to predict if a restaurant will pass or not a health inspection. The features presented in this subsection use Natural Language Processing to analyze the content of Yelp users' reviews. Kang et al. [9] preliminarily considered this aspect also and included unigrams and bigrams occurrences extracted from the text of Yelp users' reviews as features to predict the restaurant health inspection outcome.

In this paper we proceed a step further and use Latent Dirichlet Allocation (LDA) [2] to extract the main latent topics from Yelp reviews and trying to differentiate reviews dealing with health-related topics (e.g. the filthiness or health condition of a restaurant) from the other topics that are not important for predicting the health status of a restaurant, for instance, reviews regarding the restaurant employees attitude.

We preprocessed the reviews text as follows. We removed punctuation marks, stop words, and filtered out all non-English or misspelled words by using the SCOWL dictionary[7] to reduce the size of the word list considered. Next, we performed a lemmatization step by using CoreNLP [14] to group together words having the same canonical form, and then the same meaning.

After preprocessing the Yelp reviews text, we used LDA to extract meaningful topics from these reviews as follows. Since we are interested in finding topics that are related with the restaurants' health status, given the training set, we separated all the reviews in two groups: (1) *positive reviews*, i.e. reviews having a review score of 4 or more stars, and (2) *negative reviews*, i.e. reviews having a review score of 3 or less stars. Then, we trained two separate LDA models (40 topics each), one for the positive reviews and one for the negative ones with the aim of discovering topics specific for each class (passed or not passed inspection). Table 1 compares some of the topics obtained from learning the two separate LDA models from the ones we can obtain by extracting a single LDA model on all the reviews together. When we extract topics from all the reviews together, we retrieve that reviews are talking about several restaurants types, as coffee shops, Italian restaurants, etc. without giving any health related information. On the other hand, when we look at the topics extracted by separating the positive reviews from the negative ones, we see that health or bad food related topics are retrieved from the LDA learned on the negative reviews only.

Thus, we included a total of 80 topic related features, 40 topics extracted from the LDA trained on the positive reviews (*positive topics*) and 40 topics extracted from the LDA trained on the negative reviews (*negative topics*), as features for our classification task. More specifically, for each inspection, we grouped together into two super-reviews all the positive (resp. negative) reviews for the restaurant involved in the inspection from the day after the previous inspection up to the day of the considered inspection. Then, the value of each topic related feature t for that restaurant is the probability (assigned by the two extracted LDA models) that the positive (resp. negative) super review is talking about the positive (resp. negative) topic t.

4.3 Yelp Users Behavior-based Features

The last group of features we propose leverages users information we can extract from Yelp.

Average normalized review score. This feature computes the average of the normalized review scores given by the Yelp users to restaurant r. More specifically, the review score given by user u to r is normalized by considering the maximum and minimum score given by u among all its ratings.

[7]Spell Checker Oriented Word Lists (SCOWL): http://wordlist.aspell.net/

LDA computed in all the reviews	
Topic label	Words
General good related	location, fast, fresh, clean, always...
Coffe shop related	coffe, tea, ice, milk, hot, chocolate...
Italian related	pasta, italian, sauce, meatball, calamari...
Two separate LDA models Positive and Negative	
Positive LDA	
Topic label	Words
Good experience realted	good, price, night, like, place...
Mexican related	taco, mexican, salsa, burrito, food...
Cake related	pancake, red, velvet, chocolate, sweet...
Negative LDA	
Topic label	Words
Health related	dirty, floor, bathroom, old, disgust...
Late delivery related	order, call, time, delivery, wait, cold...
Bad food related	advise, microwave, spect, raw, tasteless...

Table 1: Example topics of the trained LDA models (Las Vegas dataset).

The reason is that there are, for instance, users whose usual review scores range only between 2 and 4 stars, (Yelp stars range is between 1 and 5), and then, for these users, 2 star is a very bad score and 4 is a very good score (equivalent to 1 and 5 in the Yelp range, respectively). Let $s(u,r)$ be the review score given by user u to restaurant r, then the normalized review score $s_N(u,r)$ is

$$s_N(u,r) = \frac{5 \times (s(u,r) - s_{min}(u))}{s_{max}(u) - s_{min}(u)}$$

where $s_{min}(u)$ (resp. $s_{max}(u)$) is the minimum (resp. maximum) score given by u.[8] Observe that, by normalizing the review score we can also recognize a really bad score or an extraordinary good one, e.g. a score of 1 (resp. 5) on a usual score range from 2 to 4 stars.

Given a restaurant r, the average normalized review score is computed as

$$ANRS(r) = \frac{1}{|U(r)|} \sum_{u \in U(r)} s_N(u,r)$$

where $U(r)$ is the set of all users who rated r.

Restaurant authority. HITS (Hyperlink-Induced Topic Search) is a link analysis algorithm to rate Web pages [10]. For each page, two scores are computed: *hub* and *authority*. A good hub is a page that points to many other pages, while a good authority is a page linked by many different hubs. In our case, we consider a bipartite graph whose nodes are Yelp users and restaurants and there is an edge (u,r) from a user u to a restaurant r if u reviewed r. Then, we computed the hub score for each user and the authority score for each restaurant as follows

$$h(u) = \sum_{r \in R(u)} w(u,r) \times a(r)$$
$$a(r) = \sum_{u \in U(r)} w(u,r) \times h(u)$$

where $R(u)$ is the set of all restaurants reviewed by user u and $U(r)$ is the set of all users who reviewed restaurant r, and $w(u,r)$ is a weight on the edge (u,r). We consider three values for $w(u,r)$

(1) $w(u,r) = 1$: in this case, hub and authority scores are computed according to the standard HITS algorithm. A restaurant with high authority score is a popular one as it is highly reviewed from users that make many reviews. However, in order to distinguish whether the restaurant is popular because it is a well or badly reviewed one, we consider the other two following edge weights.

(2) $w(u,r) = s(u,r)$: in this case the restaurant authority increases if the restaurant receives good reviews from users with high hub score, and the hub score of a user increases if the user gives good scores to good restaurants. A restaurant with high authority is then very good one.

(3) $w(u,r) = 5 - s(u,r)$: in this case the restaurant authority increases if the restaurant receives bad reviews from users with high hub score, and the hub score of a user increases if the user gives bad scores to bad restaurants. A restaurant with high authority is then very bad one.

If the classical HITS formula is used (i.e. $w(u,r) = 1$), we denote by $a(r)$ the authority of the restaurant r. If $w(u,r) = s(u,r)$ (resp. $w(u,r) = 5 - s(u,r)$), we denote the corresponding restaurant authority by $a_s(r)$ (resp. $a_{5-s}(r)$).

Thus, we consider the restaurant authority as feature for our classification task. In particular, we included all the three versions we defined, i.e. $a(r)$, $a_s(r)$, and $a_{5-s}(r)$.

Users Filtering. Yelp Elite users are people that are awarded a Elite status for their reviews quality, authenticity and contribution to the network. This means that these particular users are considered by the Yelp network to be trustworthy. Thus, we used this information to fight the fake reviews. There are two types of fake reviews: some are really good with the intend of increasing the restaurant overall rating, while others are really bad and try to do the opposite, i.e. lowering the restaurant rating. Thus, to fight this phenomenon, we filtered out reviews by using the information about the Elite users as follows. Given a restaurant r we got all the reviews for r and computed a weighted average of all the review scores where the weight of a review is 3 if it is done by an Elite user, and 1 otherwise. After the weighted mean is computed, all reviews that are 2 standard deviations away from the weighted mean value are filtered out as we consider them as fake reviews.

5. EXPERIMENTS

We implemented the features we propose in this paper and the ones proposed by Kang et al. [9] and compared them on the prediction task by using different classification algorithms, namely Linear SVM and Logistic Regression with *l1* and *l2* regularization, and Random Forest, from the Scikit-learn library [15]. We used the sample weighting to deal with class imbalance.[9] All results reported in the paper are obtained with the best classifier in each case. To extract the latent topics from Yelp reviews, we used the LDA implementation where Gibbs sampling is used for parameter estimation and inference. Performances are evaluated according to accuracy, precision, recall, F1 score, Cohen's kappa coefficient, and Matthews correlation coefficient (MCC). The last two measures are considered as the datasets we used are unbalanced.

We compared the performances of all our features with the ones of the features proposed by Kang et al. via 10-fold cross validation. All the features were computed using the information between the actual inspection date and the

[8]To extract the general trend of user u review scores, we computed $s_{max}(u)$ and $s_{min}(u)$ after removing unusual scores that are more than 2 standard deviations away from the review scores mean value. Note that this filter is not applied in the computation of $ANRS(r)$ measure.

[9]We did consider SMOTE over-sampling techinique [3] but sample weighting performed better.

	Accuracy	Cohen's kappa	MCC	Precision (*passed*)	Precision (*not passed*)	Recall (*passed*)	Recall (*not passed*)	F1 Score (*passed*)	F1 Score (*not passed*)
Seattle_50									
All Kang	**0.9729**	0.0401	0.0460	0.9784	**0.0983**	**0.9943**	0.0333	**0.9863**	0.0492
All Our	0.8921	**0.1021**	**0.1374**	**0.9846**	0.0825	0.9037	**0.3794**	0.9423	**0.1348**
Improvement (%)	(-8.31)	(154.61)	(198.70)	(0.63)	(-16.07)	(-9.11)	(1039.34)	(-4.46)	(173.98)
Kang + Our	0.9664	0.0337	0.0374	0.9783	0.0799	0.9876	0.0378	0.9829	0.0490
Seattle									
All Kang	0.5559	0.1022	0.1024	0.5836	0.5196	0.6157	0.4860	0.5989	0.5019
All Our	**0.5713**	**0.1410**	**0.1414**	**0.6077**	**0.5334**	0.5781	**0.5635**	0.5922	**0.5477**
Improvement (%)	(2.77)	(37.96)	(38.09)	(4.13)	(2.66)	(-6.11)	(15.95)	(-1.12)	(9.13)
Kang + Our	0.5597	0.1003	0.1026	0.5789	0.5277	**0.6813**	0.4177	**0.6242**	0.4611
Las Vegas									
All Kang	0.7104	0.1655	0.1697	0.7851	0.4045	**0.8453**	0.3078	**0.8135**	0.3450
All Our	0.7057	**0.2439**	**0.2447**	**0.8146**	**0.4233**	0.7861	**0.4657**	0.7999	**0.4428**
Improvement (%)	(-0.66)	(47.37)	(44.20)	(3.76)	(4.65)	(-7.00)	(51.30)	(-1.67)	(28.35)
Kang + Our	**0.7113**	0.1855	0.1897	0.7910	0.4165	0.8360	0.3390	0.8121	0.3680

Table 2: Comparison between our features and previous work. All improvement percentages are w.r.t. Kang et al. [9] features.

	Accuracy	Cohen's kappa	MCC	Precision (*passed*)	Precision (*not passed*)	Recall (*passed*)	Recall (*not passed*)	F1 Score (*passed*)	F1 Score (*not passed*)
Las Vegas									
Location	0.5920	0.1153	0.1243	0.7952	0.3141	0.6136	0.5276	0.6922	0.3932
$ANRS(r)$	0.5617	0.0227	0.0244	0.7579	0.2637	0.6102	0.4171	0.6746	0.3212
$a(r)$	0.6364	0.1894	0.2014	0.8207	0.3586	0.6597	0.5665	0.7301	0.4377
$a_s(r)$	0.6472	0.1783	0.1853	0.8095	0.3597	0.6931	0.5100	0.7450	0.4192
$a_{5-s}(r)$	0.6243	0.1906	0.2069	0.8270	0.3562	0.6317	0.6020	0.7136	0.4452
LDA topics	0.7030	0.2297	0.2303	0.8092	0.4174	0.7895	0.4447	0.7990	0.4297
All	0.7057	0.2439	0.2447	0.8146	0.4233	0.7861	0.4657	0.7999	0.4428

Table 3: Feature ablation: our features on Las Vegas dataset.

previous inspection date.

Table 2 shows classification results with all our features in comparison with the features from Kang et al. [9], and the combination of the two approaches. We see that our approach results better, over all the three datasets, in terms of Cohen's kappa, MCC, and recall for the class *not passed*. The obtained accuracy is comparable with the one of the competing approach, except in the case of Seattle_50 which is the most unbalanced one. However, due to the unbalance of the datasets, Cohen's kappa and MCC are more significant measures than accuracy in this case. In particular, we obtained relative improvements for these two measures that go from 38% to 199%. Moreover, the higher recall for the class *not passed* indicates that our approach is better in retrieving unhealthy restaurants (we have relative improvements going from 16% to 1038%). The combination with the competing set of features decreases the performances of our approach. All the improvements (positive or negative) shown in Table 2 are statistically significant with a p-value$<$ 0.05 or less (t-paired test), except for the case of the precision for the class of not passed inspections.

Table 3 reports the feature ablation for our case on the Las Vegas dataset. We note that the best features are the LDA topics and the restaurant authorities w.r.t. to all the measures considered, except the recall for for the class *not passed*, where the location-based features are better. In general, among the three restaurant authority features, $a_{5-s}(r)$ is the best one. [10]

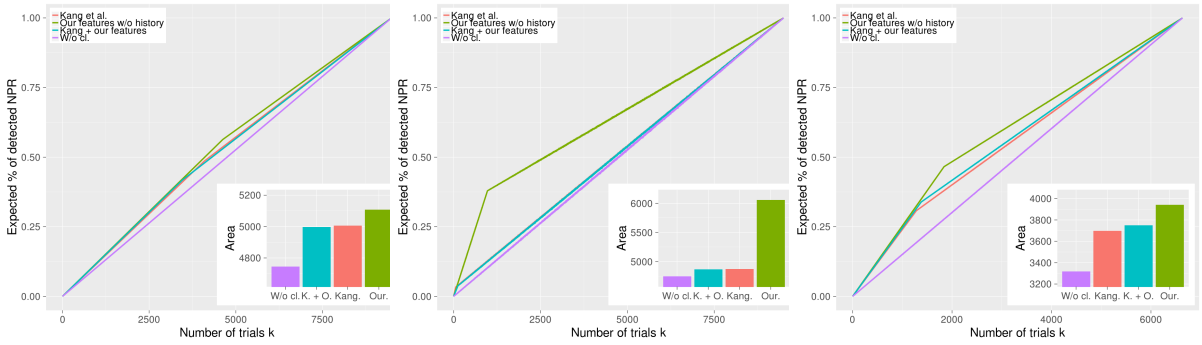
The experiments in this section prove that our approach is generally better than the one of Kang et al., but the classification problem turned out to be very tough. Even if we obtained high improvements for some measures, it is difficult to understand the effectiveness of these approaches in helping restaurant health inspectors in detecting unhealthy restaurants. Thus, to further investigate this case, we propose a new methodology analysis in the following section.

6. INSPECTOR SIMULATION MODEL

In this section we define a new method to analyze the effectiveness of the above classifiers. The restaurant health inspection outcome prediction problem is affected by two main issues such as unbalanced data and data that does not completely characterize the entire domain of the inspections. For instance, Yelp contains reviews that can be positive or negative, but not necessary correlated to the result of the health inspection. An example of bad review for a restaurant that passed the health inspection is *"The pizza have a burned taste and cold. The garlic fries is worst. Won't see me again"*, while a good review for a restaurant not passing the inspection is *"Good food at right price."* Also features as restaurant location or cuisine type suffer of the same issue. In the previous section we showed several measures to evaluate the performance results, but most of them do not clearly summarize the real situation. Thus, we define a new method to evaluate the performances of the classifiers.

We define an inspector simulation model where there is exactly one inspector that does at least k inspections in sequence. We distinguish two cases by considering if the inspector uses a classification model or not. Our main goal is to compute the expected number of restaurants that do not pass the inspection among the restaurants checked by the health inspector. This kind of analysis allows us to evaluate the help of the classifier in early identifying the restaurants

[10] All the observed patterns are the same also for Seattle and Seattle_50 datasets. Because of lack of space, we do not report the corresponding tables.



(a) Seattle (b) Seattle_50 (c) Las Vegas

Figure 3: Inspector simulation curves to the vary of number of trials k. "W/o cl." means that no classifier is used to compute the curve.

Algorithm 1 Inspector simulator.

```
1: procedure simWithoutClassifier(P,NP,k)
2:     NPR = 0;
3:     for simulation from 1 to 10,000 do
4:         NPR = NPR+COMPUTE(P,NP,k)
5:     end for
6:     return NPR/(NP*10,000);
7: end procedure

8: procedure simWithClassifier(TN,FP,FN,TP,k)
9:     NPR = 0;
10:    for simulation from 1 to 10,000 do
11:        k1 = min(k, FN + TN)
12:        NPR = NPR+COMPUTE(FN,TN,k1)
13:        NPR = NPR+COMPUTE(TP,FP,k − k1)
14:    end for
15:    return NPR/((TN+FP)*10,000);
16: end procedure

17: procedure compute(P,NP,k)
18:    NPR = 0
19:    for inspection from 1 to k do
20:        b= random number in [0, 1);
21:        if (b < NP/(NP+P)) then
22:            NP = NP − 1;
23:            NPR = NPR + 1;
24:        else
25:            P = P − 1;
26:        end if
27:    end for
28:    return NPR;
29: end procedure
```

that will not pass the health inspection, also under the assumption of having a limited number of trials k.

First, we consider the case where the inspector does not use any classifier, that is modeled by the method simWithoutClassifier in Algorithm 1, where P is the number of passed inspections and NP is the number not passed inspections. This method initially sets the number of not passed inspections NPR to 0 (line 2), and successively, for each of the 10,000 simulation runs, it adds up to NPR the number of not passed inspections found by the inspector in the current simulation and returned by the method Compute (line 4).

The method Compute generates, for each of the possible k trials, a random number b. If b is lower than $\frac{NP}{NP+P}$, then it simulates the fact that the inspector inspected a restaurant that does not pass the inspection, and, consequently, NP is decreased by one unit (line 22) and NPR is increased by one unit (line 23). Otherwise, if $b \geq \frac{NP}{NP+P}$, it means that the inspector selected a restaurant that passes the inspection, and then the algorithm decreases P by one unit (line 25).[11] Finally, the method will return the number of not passed inspections found in these k trials (line 28). Next, the method simWithoutClassifier computes the expected number of not passed inspections among all the simulations ($e = \frac{NPR}{10,000}$), and returns the percentage of not passed inspections ($\frac{e}{NP}$) for a given k (line 6).

Second, we consider the case where the inspector is helped by a classifier, then the inspector first will inspect the restaurant that the classifier predicts as not passed ($FN + TN$) and then, if the there are still trials available, he will inspect the restaurants predicted as passed[12]. Thus, to simulate this behavior, we give in input the confusion matrix of the classifier (TN,FP,FN,TP)[13] to the method simWithClassifier. This method, similarly to the method simWithoutClassifier, initializes $NPR = 0$ and iterates over 10,000 simulation runs. At each iteration, it computes $k1 = \min(k, FN+TN)$ trials (line 11) as the minimum between the total number of k available trials and the number of not passed inspections predicted by the classifier. Next, it does the simulation by using the method Compute and adds up to the variable NPR the value returned (line 12). Intuitively, it means that the inspector first focuses on restaurant classified as not passed by the classifier. Then, if other trials are available ($k - k1 > 0$) the inspector will search on the remaining restaurants (line 13). Finally, the method will return the expected number of not passed inspections among all the simulations divided by the actual number of passed inspections ($TN + FP$) (line 15).

Figure 3 shows the expected number of not passed inspections over all passed inspections to the vary of k (k varies from 1 to the size of the test set), when the inspector does not use any classifier (purple line - simulated with the method simWithoutClassifier from Algorithm 1) and when he uses our classifier (green), the classifier from Kang et al. (red), and the combination of both (blue), simulated with the method simWithClassifier. All the three

[11] The decrement of either P or NP indicates that each restaurant is inspected once for each simulation.

[12] Whatever is the value of classification accuracy or any other measure, it is always possible to have false positives.

[13] The confusion matrix is computed on the test set.

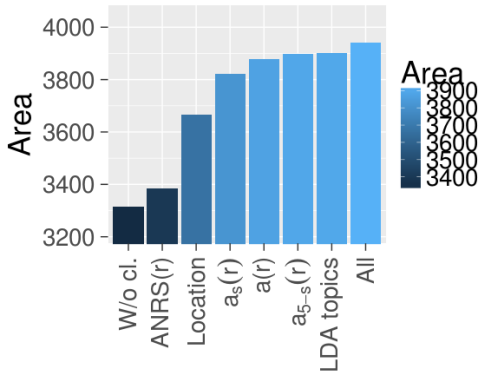


Figure 4: Our features ablation on Las Vegas dataset accouding to the area under the simulation curve. "W/o cl." means that no classifier is used to compute the curve.

datasets are considered. Within each plot there is a histogram showing the area under these curves. The higher is the area, the better is the classifier in helping the inspector to find unhealthy restaurants. Also, the slop of the curve denotes how fast is the inspector in finding these restaurants. First of all, we observe in Figure 3 that using a classifier is always better than conducting inspections at random without any guidance, even if the classifier prediction results are not so high. Moreover, we see that our classifier always outperforms the one Kang et al. and each improvement is statistically significant with a p-value < 0.01. Figure 4 shows the feature ablation for our approach in Las Vegas dataset according to the area under the simulation curve. We confirm the most important features are the LDA topics and the restaurant authority $a_{5-s}(r)$ for all the three datasets (the plots for the Seattle datasets are not reported due to the lack of space).

7. CONCLUSIONS

In this paper we studied the problem of prioritizing restaurant health inspections by using classification techniques trained on public health inspection records and Yelp data. We built a new dataset for the city of Las Vegas. We evaluated existing methods on our Las Vegas dataset and an existing dataset for the city of Seattle according to many well-established performance measures. The result that came out was not satisfactory, then we defined a novel set of features for the prediction task based on restaurant sub-area location, Yelp reviews content, and Yelp users behavior. We showed that our approach significantly improves over past work in terms of Cohen's kappa, Matthews correlation coefficient, and recall for the class of not passed inspections, while it remains comparable in accuracy and precision. In addition, we further studied the effectiveness of the classifier in helping public inspectors in finding health-violating restaurants by proposing an algorithm simulating the inspector job of choosing which restaurants to inspect. Our results show that using a classifier is always much better than conducting inspections without any guidance, and that our classifier is more helpful in retrieving unhealthy restaurants than previous work.

8. REFERENCES

[1] E. Aramaki, S. Maskawa, and M. Morita. Twitter catches the flu: detecting influenza epidemics using twitter. In *EMNLP*, pages 1568–1576, 2011.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of machine Learning research*, 3:993–1022, 2003.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, pages 321–357, 2002.

[4] L. Chen, K. Tozammel Hossain, P. Butler, N. Ramakrishnan, and B. A. Prakash. Flu gone viral: Syndromic surveillance of flu on twitter using temporal topic models. In *ICDM*, pages 755–760, 2014.

[5] S. C. Collaborative. Foodborne chicago. https://www.foodbornechicago.org.

[6] M. Dredze, M. J. Paul, S. Bergsma, and H. Tran. Carmen: A twitter geolocation system with applications to public health. In *AAAI/HIAI*, pages 20–24, 2013.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise.

[8] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2009.

[9] J. S. Kang, P. Kuznetsova, M. Luca, and Y. Choi. Where not to eat? improving public policy by predicting hygiene inspections using online reviews. In *EMNLP*, pages 1443–1448, 2013.

[10] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *(JACM*, 46(5):604–632, 1999.

[11] A. Lamb, M. J. Paul, and M. Dredze. Separating fact from fear: Tracking flu infections on twitter. In *HLT-NAACL*, pages 789–795, 2013.

[12] K. Lee, A. Agrawal, and A. Choudhary. Real-time disease surveillance using twitter data: demonstration on flu and cancer. In *KDD*, pages 1474–1477, 2013.

[13] K. Lee, A. Agrawal, and A. N. Choudhary. Mining social media streams to improve public health allergy surveillance. In *ASONAM*, pages 815–822, 2015.

[14] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[16] A. Sadilek, S. Brennan, H. Kautz, and V. Silenzio. nemesis: Which restaurants should you avoid today? In *HCOMP*, 2013.

[17] J. P. Schomberg, O. L. Haimson, G. R. Hayes, and H. Anton-Culver. Supplementing public health inspection via social media. *PLoS ONE*, 11(3), 03 2016.