Journal of
*Theoretical and Applied
Electronic Commerce Research*

MDPI

Article

TipScreener: A Framework for Mining Tips for Online Review Readers

Hanyang Luo [1], Wugang Song [2] and Wanhua Zhou [2,*]

[1] Institute of Big Data Intelligent Management and Decision, College of Management, Shenzhen University, Shenzhen 518060, China
[2] College of Management, Shenzhen University, Shenzhen 518060, China
* Correspondence: 2060132016@email.szu.edu.cn

Abstract: User-generated content explodes in popularity daily on e-commerce platforms. It is crucial for platform manipulators to sort out online reviews with repeatedly expressed opinions and a large number of irrelevant topics in order to reduce the information processing burden on review readers. This study proposes a framework named TipScreener that generates a set of useful sentences that cover all of the information of features of a business. Called tips in this work, the sentences are selected from the reviews in their original, unaltered form. Firstly, we identify information tokens of the business. Second, we filter review sentences that contain no tokens and remove duplicates. We then use a convolutional neural network to filter uninformative sentences. Next, we find the tip set with the smallest cardinality that contains all off the tokens, taking opinion words into account. The sentences of the tip set contain a full range of information and have a very low repetition rate. Our work contributes to the work of online review organizing. Review operators of e-commerce platforms can adopt tips generated by TipScreener to facilitate decision makings of review readers. The convolutional neural network that classifies sentences into two classes also enriches deep learning studies on text classification.

Keywords: tips mining; information overload; user-generated content

check for updates

Citation: Luo, H.; Song, W.; Zhou, W. TipScreener: A Framework for Mining Tips for Online Review Readers. *J. Theor. Appl. Electron. Commer. Res.* **2022**, *17*, 1716–1740. https://doi.org/10.3390/jtaer17040087

Academic Editor: Eduardo Álvarez-Miranda

Received: 24 October 2022
Accepted: 2 December 2022
Published: 5 December 2022

1. Introduction

For merchants on e-commerce platforms, good use of customer feedback such as online reviews can help companies better understand customer expectations and correct deficiencies [1]. For platform consumers, online reviews written by previous users can provide them with more realistic information about merchants and products. However, the convenience of customers being able to express their opinions has also led to an explosion of information. Managers often receive duplicate messages in a pile of text, such as in a hotel where 40 percent of the tens of thousands of reviews are of guests blaming the front desk staff for their poor attitude [2]. While this can highlight the seriousness of the problem, receiving the same information too often can fatigue readers and reduce their attention for other information. At the same time, people always add off-topic content to their comments, and it requires some effort to weed out redundant and irrelevant information. If all of the key information that is not duplicated can be extracted and then made available for human reference, it will greatly improve the efficiency of companies in responding to the market, and review readers can make consumption decisions more efficiently. Therefore, the purpose of this paper is to propose a framework which can refine sentences that cover all of the most useful information possible from all of the reviews of a product or entity. These sentences are called "tips".

From a customer's perspective, reviews posted by previous adopters reflect more realistic and comprehensive information than the merchant's presentation [3]. Online user-generated content has become the most important source of information about their products when shopping online and the most critical factor influencing their purchase decisions [4]. However, it is impossible for customers to process such a large amount of information comprehensively. They can only selectively read the information at the top of the review list. Studies have shown that more than 90% of customers make a purchase decision based on no more than 10 reviews, while 68% make a purchase decision based on only 1–6 reviews [5,6]. Thus, even if the information is sufficient, the reader cannot use it effectively. How to organize the user-generated content to present information in a more comprehensive and readable way that efficiently delivers information to readers has received considerable critical attention from e-commerce operators.

Both platforms and researchers have come up with ways to summarize customer opinions to address the above issues. Most of the existing platforms and forums such as Ctrip.com and autohome.com have designed display interfaces that sort reviews or discussions by the number of votes representing the information's helpfulness. However, the number of votes can be affected by the number of readings due to time factors. Suppose there is a newly opened hotel. When the second customer makes a comment, the first one has already received a lot of votes. At this time, it is unfair to display their comments in order according to the number of votes. Some platforms allow customers to rate dimensions such as hotel location, service attitude, price and sanitation, as well as car appearance, engine dynamics, prices, and space. While managers can see shortcomings clearly and customers can get a general idea of the product by a comparison of scores, it is difficult to distinguish between similar products to highlight some specific advantages since the dimensions are relatively large. To allow readers to have a more in-depth understanding of product details, dynamic web pages attached with tags have currently become a popular information classification layout. Different attributes and keywords are set into more detailed tags so that customers can select the information they are interested in. The problem is that tags are not updated in time. They are always limited to the listed words mentioned frequently, and it is difficult to detect some novel and innovative features.

Recently, a few works have proposed methods to generate high-quality tips from the overwhelming number of user-generated contents [7–9], which share a very similar goal with our research. Tips are short text snippets that provide valuable insights into specific aspects of the business being reviewed. Typically, they are a one-sentence text distilled from the original online reviews without being altered, saving efforts for small-screen device users or people short in time. Given the right approach, extracted tips will provide a concise overview of the product's strengths and weaknesses. In this study, we adopt a hybrid method of supervised and unsupervised learning to screen helpful information and generate tips from online customer reviews, aiming at addressing the challenges of redundant information proliferation. Specifically, we propose TipScreener, a framework for searching a general tip-set and a unique tip-set of a business's reviews. The first set provides information about features shared with other similar businesses. The second set provides information on the unique characteristics of the business. Correspondingly, the tokens of each that represent the features are named general tokens and unique tokens in this paper. In our framework, we first identify similar businesses to the one in the study. Then, we recognize general tokens and unique tokens from their online reviews. The tokens are information identifiers that we use to select review sentences. The final results are the two smallest sets of tips containing as much information delivered by previous adopters. To reduce time complexity, inspired by Timoshenko and Hauser [10], we use a deep learning classifier to initially screen unnecessary sentences in the set of original review sentences, greatly improving the efficiency of tokens and sentence matching. Although the training process of supervised learning requires certain labor costs, it also reduces the time of subsequent informative sentences identifying and improves the quality of tips (e.g., filtering out fake reviews).

The rest of this paper is organized as follows. In Section 2, we review the work related to approaches to overcoming information overload, with a focus on the field of user-generated content on e-commerce platforms. In Section 3, we present our supervised and unsupervised mixed algorithm 'TipScreener', which generates two meaningful sets of tips from customer reviews. Next, an experiment on restaurant reviews is conducted and evaluations of the proposed algorithm are presented in Section 4. We finally conclude our work in Section 5 with a discussion on managerial implications and future work.

2. Literature Review

We review studies on approaches to extracting information from user-generated content. Given that the state-of-the-art deep learning method is involved in the framework, we also give a brief overview of text information classification and convolutional neural network (CNN).

2.1. Methods to Extract Information from User-Generated Content

Human memory and information processing capacity are limited, but the information acquired or received by the environment is always much higher than the information it can consume, bear, or need, and a lot of redundant information seriously interferes with its accurate analysis and correct selection of relevant useful information. The effect is particularly severe when consumers browse product reviews on e-commerce platforms. Scoring is one of the first methods used by almost all e-commerce platforms to make product information clearer at the product level dimension. Some studies suggest that it is more reasonable to create a fine-grained scoring system for each text review through topic modeling and sentiment analysis than to ask customers to rate each dimension of the product directly [11–15]. Original scores can also be calibrated by the scoring model based on text reviews [16]. Considering individual differences, Wang et al. [17] involve the factor of the history of users' emotional expression in their review scoring prediction model. In addition to online reviews, discussions on vertical forums such as MForum are also important sources of public attitudes about a particular product, which can provide references for business practitioners to improve their products [18]. Scoring prediction models can make up for these user-generated contents that are not rated. However, the dimensions are usually too general and only allow for rough side-by-side comparisons. It is difficult for consumers to learn in detail about why the score of some dimensions are low (let alone fundamentally address the problem of review information overload). Due to the insufficient expressiveness of numerical scores, some recommendation systems use a combination of ratings, topic summarizing and representative reviews to present product information for readers to understand in multiple directions [19]. Specifically, they use natural language processing (NLP) techniques to extract information from product reviews submitted by users. Three types of information streams are presented to consumers including but not limited to (1) a scoring system on costs and efficacy, (2) a summary of common topics about the product, and (3) representative reviews of the product. However, these representative reviews are always filtered based on the most commonly discussed product features by customers and fail to present a full range of more specific information.

In addition to the duplication and redundancy that comes with too much information, consumers are plagued by fake reviews. Quality evaluation of reviews can help platforms filter spam content, unhelpful opinions, as well as highly subjective and misleading information, thereby reducing information overload. Usefulness voting is one of the traditional ways to review quality evaluation. However, not all reviews are fairly evaluated for usefulness [20]. There is a potential "the rich get richer" effect, where the most popular comments accumulate more and more votes [21]. Most studies take each review as an independent text document, extract features from the text, and learn functions to evaluate the quality of reviews based on these features [22–25]. In order to enhance the accuracy of the filtering mechanism, Lu et al. [26] consider social network interaction in their quality evaluation function to assess the quality and authenticity of reviewers and their reviews.

Summarization is also a popular way to address user-generated information overload. Cremonesi et al. [27] propose a framework to summarize all customer reviews that capture salient aspects of crowd judgment and construct at least suboptimal solutions. Five techniques are compared in the work, namely, LSA [28], TextRank [29], LexRank [30], Luhn [31], Edmunson [32], among which the last one has the best performance. They separate reviews into positive and negative groups based on ratings, and then the techniques extract no more than four sentences from the original reviews to produce a summary. But most reviews contain both praise and suggestions for improvement and could not be simply classified by ratings, so it could have been misclassified in the first place. Compared with it, the work of Hu and Liu [33] identifies opinion sentences for product features before sentiment orientation classification. They use the matching of opinion words and feature words to automatically generate positive and negative direction sentences, which requires much attention to the rules of human languages. Our algorithm does not classify sentence emotion, since the direction of opinion words needs to be judged according to the user's own situation. For example, it's hard to judge whether a "large screen" is good or not for different computer users. Some of the special features may not have opinion words in the reviews. Therefore, we first recognize sentences as informative by the existence of tokens. The general capabilities of these sentences are then ranked according to whether there is an opinion word in the sentences to eliminate redundancy.

The method proposed by Guy et al. [7] to extract short practical tips from a large number of reviews is the first to suggest tip extraction. In the training stage of their work, 30 editors are first tasked with selecting a gold set of tips from city guide tips written by experts on TripAdvisor. Templates consisting of n-gram sequences that occurred in the gold tips are then classified into trivial and non-trivial types. After matching candidate reviews with the potential templates, manual annotation is involved in filtering out non-useful sentences. The training of data heavily relies on human intervention in the steps of reviews and template usefulness classification, which requires much manual cost on untrained domains. Based on this, Zhu et al. [8] present TipSelector in their work. TipSelector is a completely unsupervised algorithm that can be applied to customer reviews of any business entity. The algorithm generates tips by matching review sentences with tokens that are differentiated from similar enterprises, making tip information highly representative. But in many cases, information about the common characteristics of similar businesses is also meaningful. For example, a hotel may attract customers with tips such as "equipped with a coffee machine". But "bedding", a token shared by all hotels, is also what customers care about. Including such information in the tips can be more comprehensive. Based on the advantages of tips, we extend their work, with the main differences in our approach being as follows: (1) This study not only uses tokens as information carriers to screen sentences but also considers opinion words to select superior sentences. For example, "I like this computer screen very much" and "the computer screen is clean" seem to contain the same token, which would be considered equally important in [8], but obviously the latter is more informative since it has a specific description. (2) For each business, our algorithm generates tips that include information unique to it (named unique tip-set), as well as tips that include other general information (i.e., tips which are also important to other similar entities, named general tip-set).

2.2. Text Classification and Deep Learning

Our research framework involves the classification of useful and useless sentences, so in this section, we review the types of text classification and the classification method.

Types of text classification: With the emergence of network media, the automatic data processing of user-generated content has attracted the great attention of researchers. Text classification is an important task in this field. Text classification refers to the process of automatically determining one or more categories of unknown texts in a document set according to certain rules. Text can be classified based on many different methods. Classified by content is a common method for detecting spam emails [34], cyber grooming attacks [35] and so on, focusing on keywords that distinguish vicious content. Sometimes it's used in fields such as legal documents and financial documents classification. For example, Xiong et al. [36] automatically shunted Chinese legal case texts through text classification based on the secondary dimension reduction method and improved mutual information feature extraction based on LSA, which reduced the burden of staff. Classification based on sentiment is a very important focus of UGC research, such as the evaluation of people's attitude towards the introduction of new policies using data in social platforms such as Twitter [37], the impact of hot news or emergencies on people's emotions [38], and the study of customer product satisfaction on e-commerce platforms [39–41].

Methods for text classification: The most common machine learning algorithms for text classification include k-nearest neighbor [42], Naive Bayes [43,44], and support vector machine [45]. These classifiers can be used in almost all data mining domains such as image, video and audio analysis, human behaviors, bioinformatics, safety and security, etc. [46]. They have also been used by many studies as benchmarks to compare with their proposed models. However, they largely require manual feature extraction compared to state-of-the-art deep learning. Convolutional neural network (CNN) is one of the important models of deep learning networks, which has been first applied in the fields of image processing and achieved good results [47]. In recent years, many scholars also try to use CNN to extract text features and get rid of complex artificial feature engineering [48]. Although much of the literature suggests that CNN performs well at text classification tasks, there has been far less research on it than on image classification tasks. We searched the core collection of Web of Science with the keywords "text classification, CNN" and "image classification, CNN" and found that Their numbers from the advent of CNN to the end of this study (July 19, 2022) were 1570 and 20,710, respectively. As shown in Figure 1, the annual growth rates of the two are also significantly different. We believe that the application of CNNs to text classification can be richer since text can not only describe an image from multiple perspectives and stylistic expressions but also has its own features such as different languages. It is demonstrated that CNN is state-of-the-art with minimum tuning in many text analysis research such as relation extraction [49], named entity recognition [50], and sentiment analysis [51]. In Timoshenko and Hauser's study on opinion mining of oral users using online reviews, they trained a CNN to filter out uninformative review sentences and then used clustering to group sentences with similar opinions into a cluster, greatly improving the efficiency of opinion extraction [10]. At the end of the study, they compared the efficiency of the method with CNN and the method without CNN, and found that the former far outperformed the latter on datasets of different sizes. Therefore, in this paper, CNN is adopted in the framework, hoping to enrich the research in this field.

At present, most CNN models are used for text sentiment classification [52], topic detection [53], fake reviews filtering, etc. [54], and use to solve the problems of weak semantic meanings [55] and over-fitting [56]. Although there is some research on CNN text analysis at the sentence level, little attention has been paid to uninformative sentence screening, which can save time if adequately applied. In this study, inspired by Timoshenko and Hauser [10], CNN is used to filter uninformative texts which should not be included in the candidate tips, and our data prove that CNN performs at least as well as SVM [57] and LSTM [58].

In the task of English text classification using CNN, word embeddings are usually taken as the input of text and category probability as the output. Word embeddings are real-valued vector mappings (typically 20–300 dimensions) that have been trained to bring similar words close to each other in a vector space, in line with the distributional hypothesis that words appear in similar contexts and share semantic meaning [59]. Most favorably, high-quality word embeddings can not only capture semantic similarity but also be able to do vector calculations according to semantic relations [60]. Assuming v(a) is the word embedding of "word", then the high-quality embeddings trained from a corpus will have the following relationships [10,60]:

$$v(\text{ran}) - v(\text{walked}) + v(\text{waling}) \approx v(\text{run}), \tag{1}$$

$$v(\text{Germany}) - v(\text{Berlin}) + v(\text{France}) \approx v(\text{Paris}), \tag{2}$$

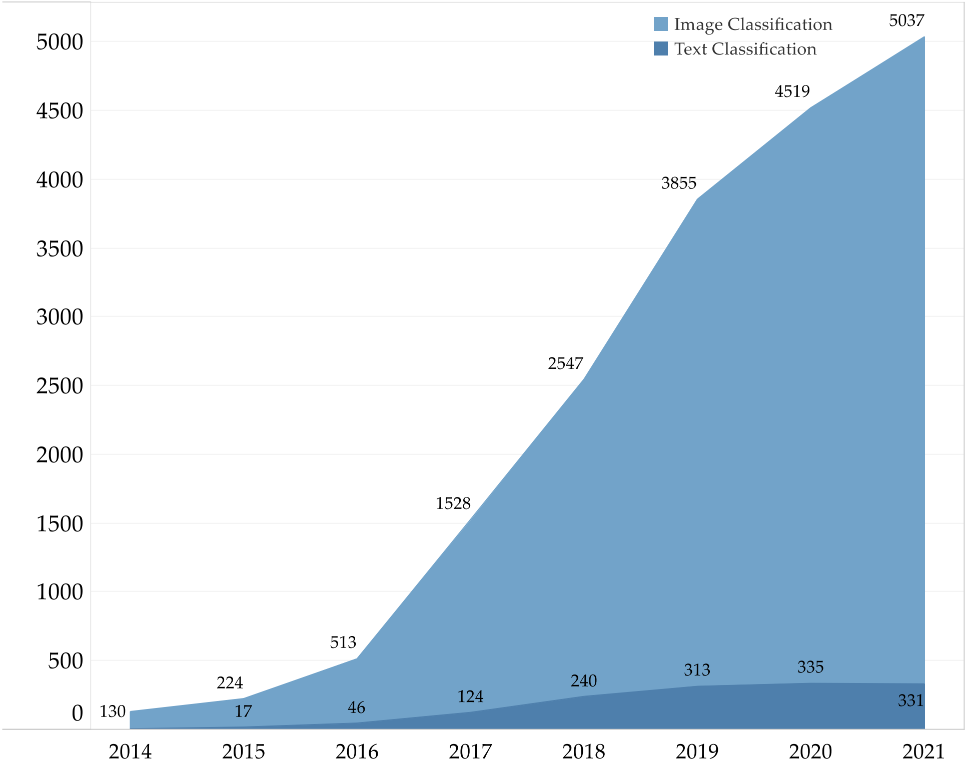


Figure 1. The number of CNN studies in the two applications.

In this work, we use a large online review corpus to train word embedding, which is then used as input to CNN to improve performance.

3. The TipScreener Framework

In this section, we introduce the TipScreener framework for mining concise and high-quality information to summarize the status or characteristics of businesses. For each business, we produce two sets of tips, one for displaying information about its unique features, and another for information about common features of similar entities. Algorithm A1 in the Appendix A presents the pseudocode of the TipScreener framework.

3.1. Identification of Similar Businesses

The fourth line of the algorithm requires that m similar businesses are found for each b. It is important to note that the value of m is not deterministic.

Set B can contain b from different market sectors, but considering that enterprises are often compared with those from the same industry in actual applications, this paper only selects catering as the candidate business. The process of a rule-based approach for identifying similar restaurants in our work is quite simple. Types of cuisine and geographical distance are used as the measurements to make pin-pinal discrimination for restaurants in set B (i.e., if the size of the set is n, $n(n-1)/2$ times of discrimination is required to find out similar businesses for all b). The first discriminant index is based on whether the cuisine of a candidate business b' is of the same type. If so, the next judgment is made. Otherwise, it is not taken as an element in B_b^m. In the second step, if the geographical distance between the b and b' is less than a certain value, the two restaurants are classified as similar businesses and b' is added to B_b^m. Operators can adjust the indicators as needed. For example, restaurants in the same mall are competitors even if their cuisines are different, so they can also be treated as similar businesses.

3.2. Identifying Information Tokens

In this stage, we introduce the recognition of the token set T_b. In this paper, a token is defined as the carrier of information and also the opinion target of review posters. Only sentences containing at least one token can convey information; otherwise, the sentences may lack a body or be semantically incomplete. Our algorithm looks for potentially meaningful review sentences based on token identification. Usually, sentence components that describe the characteristics of a product or service are nouns or noun phrases. Therefore, we chose three types of entities as tokens:

1. Named entities: named entities are the names of pre-defined categories such as mentions of persons (e.g., Jack, Rose), organizations (e.g., World Health Organization, World Trade Organization), locations (e.g., New York, Hannover), businesses (e.g., Hugendubel, Block James), etc. [61–64]. In the experiment of our research, name entities such as the name of dishes are very important tokens. Our study applies the named entity recognition system of the StanfordCoreNLP.
2. Compound nouns: collocations such as "power adapter" and "towel hanging" are composed of multiple nouns, which will become two different items if they are considered independent words separately. We extract compound nouns from the whole review corpus we obtained using the method in [10].
3. Other singleton nouns: this type of token identification method is consistent with [8], which satisfies a lower frequency limit (occurrence greater than 1%) and an upper limit (occurrence less than 99%) [65], and excludes spelling errors as well as generic terms (e.g., restaurant).

We extend the above set of tokens by adding their synonyms of whose parts of speech are nouns. To facilitate subsequent processing, we join each token with more than one word together with the "_" character and transform all letters into lower cases. To improve the quality of tokens, we also manually remove some words that are unsuitable to be considered as tokens (e.g., people will always mention the name of a waiter and praise their service, but the name itself does not have a particularly useful meaning, and tokens such as "staff" and "servant", which appear very frequently, already cover their information).

To construct the set of unique tokens T_b^{unique} for business b, we target those tokens that appear much more often in the reviews of b than in the reviews of other similar businesses. We build a table as shown in Table 1 for b and similar enterprises b', illustrating the frequency of token t appearing and the total frequency of other tokens appearing. A, B, C, and D denote the number of sentences of R_b that t appears in, the total number of sentences of R_b that other tokens appear in, the number of sentences of $R_{b'}$ that t appears in, the total number of sentences of $R_{b'}$ that other tokens appear in. Note that the sets of tokens other than t for b and b' are different.

Table 1. Table for Cochran–Mantel–Haenszel test.

	Frequency of *t*	**Frequency of Other Tokens**	**Row Total**
R_b	A	B	N_1
$R_{b'}$	C	D	N_2
Column total	M_1	M_2	T

R is defined as the odds ratio:

$$R = \frac{AD}{BC} \tag{3}$$

We propose the null hypothesis and alternative hypotheses as:

H0: *$R = 1$. i.e., the probability of t appearing in R_b is equal to the probability of t appearing in $R_{b'}$.*

H1: *$R \neq 1$. i.e., the probability of t appearing in R_b is not equal to the probability of t appearing in $R_{b'}$.*

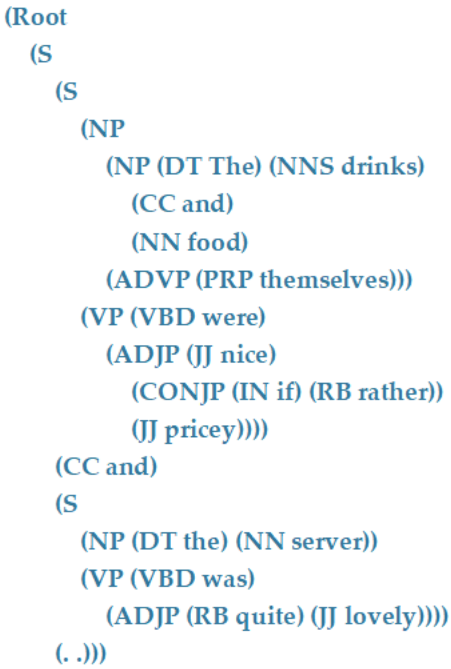
The chi-square statistic χ^2_{CMH} is calculated using sample data:

$$\chi^2_{CMH} = \frac{\left(A - \frac{N_1 M_1}{T}\right)^2}{\frac{N_1 N_2 M_1 M_2}{T^2(T-1)}} \sim \chi^2(1) \tag{4}$$

If $\chi^2_{CMH} > \chi^2(0.05, 1)$ = 3.841, the null hypothesis is rejected and the alternative hypothesis is accepted (i.e., t has different probabilities of occurring in the two review sets). In this case, if $R > 1$, it means that t is significantly more likely to appear in R_b than it is in $R_{b'}$. If b has a dominance ratio over t greater than 1 for all similar firms, then t is a unique token that can distinguish b from other similar businesses.

3.3. Matching Token and Opinion Words

According to [66], a syntactic tree can be generated for each complete sentence. For example, the syntactic tree of "the drinks and food themselves were nice if rather pricey and the server was quite lovely." is shown in Figure 2, where "NP" represents a noun phrase. We can divide each pair of token-opinion words according to the matching string "(NP". Though sometimes a token may not have opinion words in a sentence, we can easily separate it with other token-opinion words pair in this way. The syntactic tree in Figure 2 can be divided by "(NP" to get "drinks, food-nice, pricey" and "server-lovely "Since the former group consists of two tokens and two opinion words, we can get four pairs of token-opinion words: "drink-nice" "drink-pricey" "food-nice" "food-pricey". It should be noted that the noun after the NP is not necessarily a token, and each token needs to be lemmatized to its original form in the pairs.



```
(Root
  (S
    (S
      (NP
        (NP (DT The) (NNS drinks)
          (CC and)
          (NN food)
        (ADVP (PRP themselves)))
      (VP (VBD were)
        (ADJP (JJ nice)
          (CONJP (IN if) (RB rather))
          (JJ pricey))))
    (CC and)
    (S
      (NP (DT the) (NN server))
      (VP (VBD was)
        (ADJP (RB quite) (JJ lovely))))
    (. .)))
```

Figure 2. The syntactic tree of an example sentence.

3.4. Review Data Preprocessing and Preliminary Review Sentence Filtering

The algorithm proposed in this paper is applicable to languages with distinct word boundaries such as English, and the methods in the literature referenced in Sections 3.2 and 3.3 are similarly limited to such languages. It is also feasible to use advanced translation software to convert reviews from other languages into English before extracting tokens, opinion words and the next series of operations. In the data preprocessing stage, all reviews are first divided into sentences based on sentence separators such as periods, question marks, and exclamation marks. All letters are converted to lowercase and make sure there are no duplicate sentences. Since sentences that don't cover any token are considered as not containing information, they are eliminated (ReviewFilterOne). If the token in a sentence consists of more than one word, they are concatenated into a single string with "_" to facilitate subsequent matching.

3.5. Secondary Filtering of Review Sentences—A CNN Classifier

Tokens are the bearers of information, but a sentence containing tokens does not necessarily convey useful information. For example, "My girlfriend and I ordered a Hawaiian pizza" contains the token "Hawaiian pizza" but does not convey information useful to readers. A review sentence such as "I went in the afternoon but the Hawaiian pizza was sold out" tells the customer that Hawaiian pizza may be in high demand so it is better to go early to get it.

We use the CNN-static model of [67] with some parameter modifications to automatically extract text features and filter out uninformative sentences (ReviewFilterTwo). Three stages are presented in this step.

3.5.1. Stage 1. Data Processing

First, data processing is needed again. All stop words (e.g., "that" "and") and non-alphanumeric symbols except the underlines are eliminated from the sentences obtained in 3.4. We remove sentences with less than three words or longer than forty-five words. Long sentences are usually a typical example of missing punctuation.

3.5.2. Stage 2. Training Word Embeddings

We need to transform the unstructured text data into vector data as the input of CNN. In this paper, we use the skip-gram language model to train the word embedding of each word for the set of sentences obtained from stage 1. The skip-gram model determines the word vector of the current word by maximizing the output probability of the contextual word within a window of c in the model. Specifically, suppose I is the number of words in the corpus, V is the set of all feasible words in the vocabulary, and v_i is the d-dimensional real vector word embedding of the i-th word, we choose v_i to maximize:

$$\frac{1}{I}\sum_{i=1}^{I}\sum_{\substack{-c\le j\le c\\ j\neq 0}}\log p(word_{i+j}|word_i) \tag{5}$$

$$p(word_j|word_i) = \frac{\exp(v_j v_i')}{\sum_{k=1}^{|V|}\exp(v_k v_i')} \tag{6}$$

The word vector learned by a smaller context window can better reflect the word function and contextual semantic information, which is required by tips mining. We refer to the rare literature on useless sentence elimination [10] and set c to 5 and d to 20. In our application to the restaurant reviews dataset, the word vectors trained by this method capture semantic information. For example, the cosine distance between the word vectors of "light" and "brightness" is close.

3.5.3. Stage 3. Training a CNN Classifier and Filtering Uninformative Sentences

In line 17 and line 19 of Algorithm 1, matching tokens and sentences through SentenceSelector is np-complete, so we use ReviewFilterTwo's CNN classifier in advance to filter out some sentences that are uninformative even though they contain tokens. Some relatively novel or niche tokens appear less frequently, we therefore retain all sentences that contain tokens with frequencies less than 10 so that they will not be filtered out. Of the remaining sentences, some are labeled as informative ($y = 1$) or uninformative ($y = 0$) by domain experts as a training set. At this point, the input to the CNN is the word vector and the output is the predicted probability value of the label y. Figure 3 shows the structure of the CNN in our work.

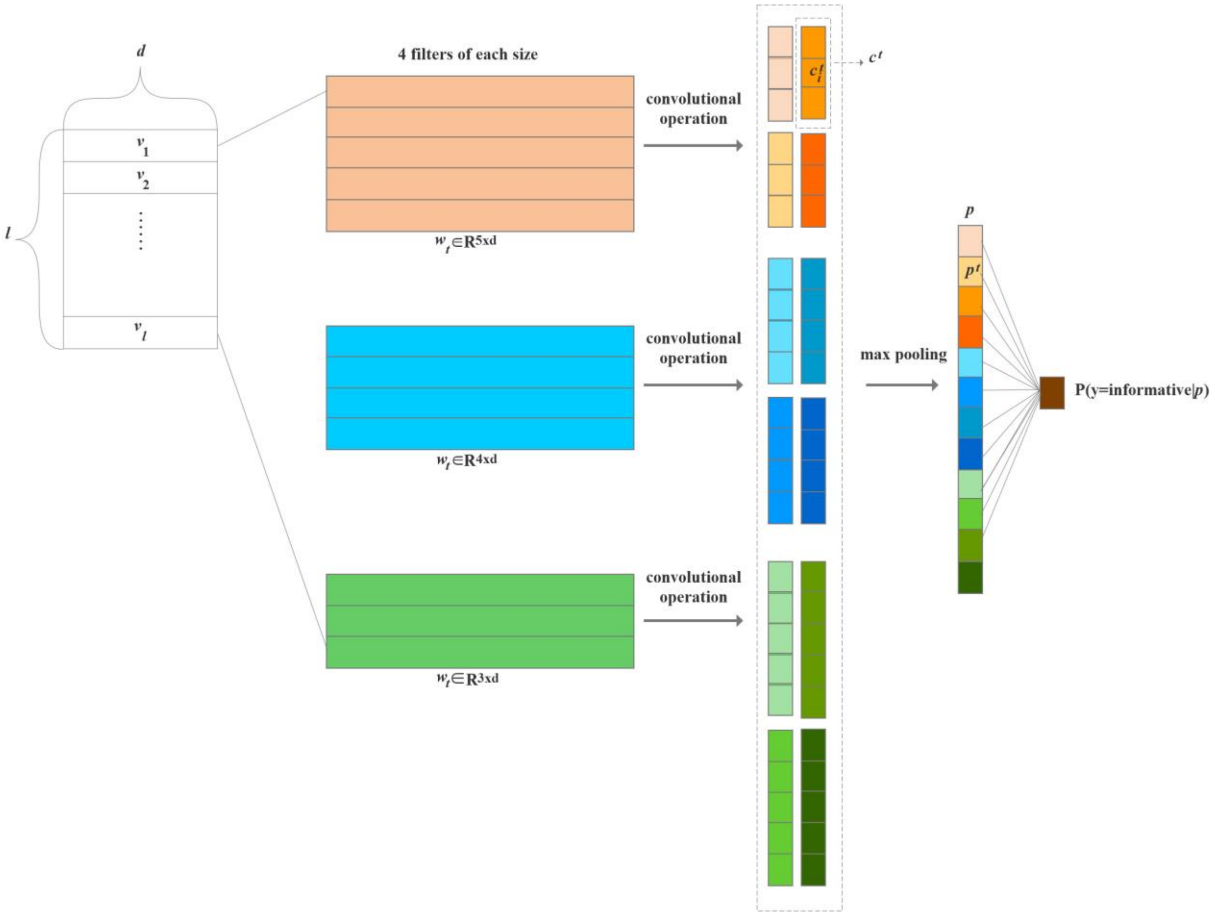


Figure 3. The structure of the CNN in this work.

Input Layer

As shown by the left-hand of Figure 3, a feature map (the word embedding matrix for the trained sentence) is used as the input for the representation of each sentence in the experiments. Future research can try to use word vectors trained by multiple models as multi-channel input feature maps or allow the CNN to update numeric representation in the training process. In this paper, a static input of word embeddings has achieved ideal results. Let the length of a sentence be l, then the matrix of the input can be represented as v:

$$v = [v_1, v_2, \ldots, v_l] \tag{7}$$

where v_i denotes the word embedding of the i-th word in the sentence.

Convolutional layer

This layer uses several filters in different sizes for feature selection and extraction of the input information. Each filter is a weight matrix with the same channel depth as the input feature map. Since the dimension of our input feature map is $l \times d$ (channel of 1), the filter should also be two-dimensional. Distinguishing from the pixel matrix of a picture, each row in the input sentence feature map represents a word, which is meaningless after slicing, so the length of the filter should be $d = 20$, and we set their height to h_t. Let $w_t \in R^{h_t \times d}$ be the matrix of the t-th filter, then it convolves the input feature map from top to bottom with a stride of 1 to generate a new feature map c^t:

$$c^t = \left[c_1^t, c_2^t, \ldots, c_{l-h_t+1}^t\right] \tag{8}$$

$$c_i^t = ReLU\left(w_t * v_{i:i+h_t-1} + b_t\right) \tag{9}$$

where $ReLU\ (\cdot)$ is a nonlinear activation function, and $ReLU(x) = \max\ (0, x)$,b_t is a bias term. In this paper, we set up filters with h_t of 3, 4, and 5, respectively, four of each (the number of these hyperparameters is set based on the previous research that work well on sentence classifications, e.g., [10,67]). Therefore, the parameters to be trained in this layer are the twelve matrixes w_t and the twelve b_t.

Pooling Layer

The pooling layer compresses all of the twelve feature maps obtained from the convolution layer into shorter vectors. Pooling layers can effectively reduce the number of parameters, speed up the computation and prevent overfitting. We apply the maximum pooling operation to generate vector p [67]:

$$p = [p_1, p_2, \ldots p_{12}] \tag{10}$$

$$p_t = \max\left\{c_1^t, c_2^t, \ldots, c_{l-h_t+1}^t\right\} \tag{11}$$

The pooling layer captures the most important features by selecting the maximum value in each new feature map. This layer has no parameters that need to be trained.

Fully connected layer with dropout and SoftMax output

The fully connected layer is equivalent to a traditional neural network. Dropout is the process of temporarily dropping a portion of units from the network according to a certain probability during the model training [68]. Figure 4 is an example of the layer before and after dropout. The introduction of dropout transforms the effect of one model into the sum of the effects of multiple models, which is more similar to multi-model voting and improves model stability and robustness. We set an output neuron in the last layer for classification, and set the dropout rate of the fully connected layer to 0.5. The final output of the model relies on the SoftMax function to predict the probability that the input sentence belongs to the informative type. The estimate of the probability that the sentence is informative, $\mathrm{P}(y = informative|p)$ is given by:

$$\mathrm{P}(y = informative|p) = \frac{1}{1 + e^{-\theta p}} \tag{12}$$

where θ is the parameter vector to be determined in the training process. When $\mathrm{P}(y = informative|p) > 0.5$, the sentence is classified as informative.

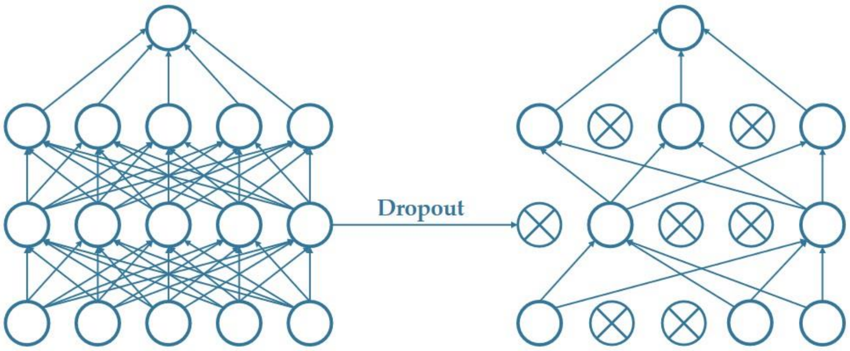


Figure 4. An example of a fully connected layer before and after dropout.

Parameters training

We have some of the sentences labeled. To train the model, we divided them into a training set and a validation set in a ratio of 8 : 2, and train the unpredetermined parameters (including the weight parameters w,$\forall w_t \in w$, and the bias term b, $\forall b_t \in b$ of the filters and the trainable variable θ in the fully connected layer) by minimizing the cross-entropy loss on the labeled sentences:

$$w^*,\ b^*,\ \theta^* = \underset{w,b,\theta}{\operatorname{argmax}} L(w, b, \theta) \tag{13}$$

$$L(w,b,\theta) = -\frac{1}{N}\sum_{n=1}^{N}\log p_n \tag{14}$$

where N is the number of sentences in either a batch of the training set or the validation set, p_n is the prediction of the CNN. The cross-entropy loss in the training set is used for backpropagation, and training is stopped when the cross-entropy loss no longer decreases in the validation set. We choose the Adam algorithm to optimize the parameters on a mini-batch size of 32. Adam optimizer combines the advantages of AdaGrad and RMSProp optimization algorithms. The step size is updated by considering the First Moment Estimation and Second Moment Estimation [69].

Finally, the trained CNN model (ReviewFilterTwo) is used to predict the category of the remaining sentences to obtain a set of informative sentences S_b^2.

3.6. Finding the Most Informative Collection of High-Quality Tip Sentences

The main function in the framework is SentenceSelector in lines 17 and 19, which filters sentences based on the matching of tokens and sentences, and finds higher quality sentences based on the presence of opinion words corresponding to the token. Therefore, we have two goals: fewer sentences and more tokens with opinion words. Formally, given the set of candidate sentences S and set of tokens T, let T_s^1 be the set of tokens with at least one opinion word pair in a sentence s, and T_s^0 be the set of tokens that do not have an opinion word in s. We want to find $S^* \subseteq S$ such that $T \subseteq \left\{t \middle| t \in T_s^0 \ or \ t \in T_s^1,\ s \in S^*\right\}$, $\sum_s \left|T_s^1, s \in S^*\right|$ is maximized and $|S^*|$ is minimized.

In each iteration, we first rearrange all of the candidate sentences and select the best sentence (i.e., the sentence that may contain the most information) and add it to the final tips set. The rules of ranking are as follows: the more tokens without repetition, the better the sentence. For example, there are two sentences: "Food was delicious, service was prompt and the atmosphere was very cozy and intimate," and "Staff were extremely nice and so lovely to chat to." The first sentence has three tokens (i.e., food, service, and atmosphere), while the second sentence has only one token (i.e., staff), so the former is better than the latter. For sentences with the same number of tokens, they are sorted according to the number of tokens with which there is an opinion word. Again, there are two sentences: "I recommend sitting by the window." "Sitting by the window is more comfortable and quiet." The latter contains the opinion words "comfortable" and "quiet" and is therefore preferable. Every time we select the first sentence, we count the tokens that it contains. When we rearrange the rest of the sentences with the number of tokens, if a sentence contains a previously covered token or its hypernym, and the opinion word expresses the same meaning, the token will not be counted for the sentence. Our goal is to give tips readers concise and clear information, and if future research wants the description of tips to be more detailed and colorful, it's better to add rules based on the number of opinion words for the same token.

3.7. Enhance the Readability of the Final Tips-Set

After TipScreener generates two collections of tips, we still need to do the final sorting. Since the sentences in the unique tips are likely to contain general tokens, to avoid duplicate opinions, we need to delete the sentences in the general tips whose token has been covered by the unique tips. To make it easier for readers to catch the focus of tips, we propose to highlight different categories of tokens with different colors.

4. The Application of TipScreener in the Catering Industry

Data

The data used in this section comes from TripAdvisor, a major travel website that is well-known and popular worldwide. We randomly crawled 550 URLs of the top 1000 restaurants located in London on TripAdvisor's recommended restaurants page, which are distributed in different ranges and categories in terms of stars, prices, and cuisines. Among these restaurants, we removed 28 more outlets that were open less than one year and had less than 200 reviews at the time we carried out the research. The locations of the 522 restaurants are shown in the map of Figure 5.

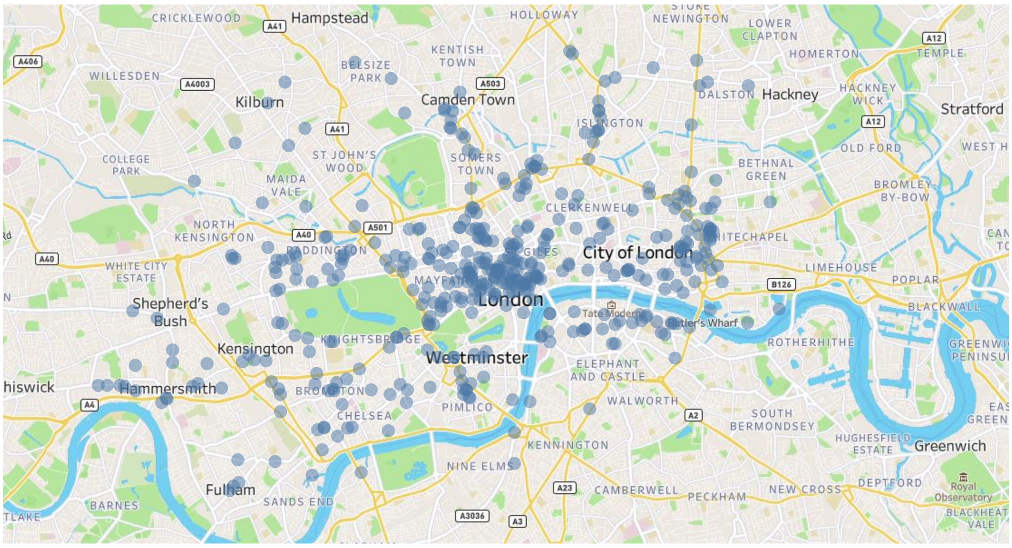


Figure 5. The locations of the restaurants on the map.

In practice, restaurant managers are able to subjectively determine their similar businesses (or competitors in other words) according to the actual situation and directly obtain customers' online reviews on these restaurants. In the experiment, in order to imitate the subjective judgment process of managers, we conducted similarity enterprise identification for the 522 restaurants according to Section 3.1. We draw a word-cloud figure to show the number of cuisines of them (Figure 6). It is shown that Mediterranean, British and Italian cuisines rank in the top three of the most common cuisines. We selected a restaurant named Mihbaj Cafe and Kitchen ("Mihbaj" in the following text) as the main object of analysis in this study and obtain all online reviews of its and its similar businesses. The main information about these restaurants is shown in Table 2 and their average prices per person are shown in Figure 7.

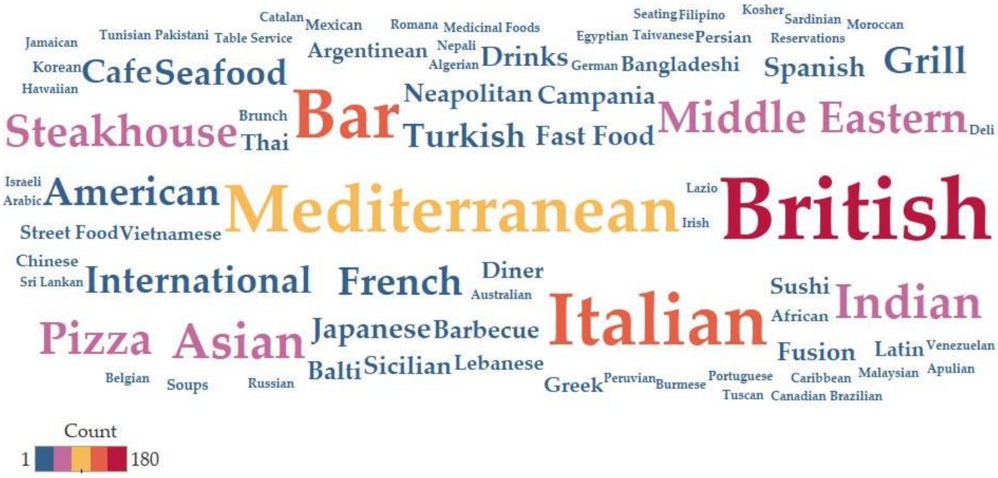


Figure 6. Word cloud of the cuisines.

Table 2. Basic information on Mihbaj and its similar entities.

Name	Location	Coordinates	The Distance from Mihbaj	Average Price per Person (£)	Cuisines
Mihbaj Cafe and Kitchen	153 Praed Street What3words, London	(51.5161602, −0.1744095)	0.0	9.72	Lebanese, Arabic
Cookhouse Joe	55 Berwick Street, London	(51.515617305, −0.1364719)	1.64	8.18	Lebanese, Mediterranean
Maroush Express	68 Edgware Road, London	(51.5151339, −0.1627560)	0.51	16.48	Lebanese, Mediterranean, Middle Eastern
Noura	16 Hobart Place, London	(51.4981821, −0.1477896)	1.69	25.65	Lebanese, Mediterranean, Grill, Middle Eastern

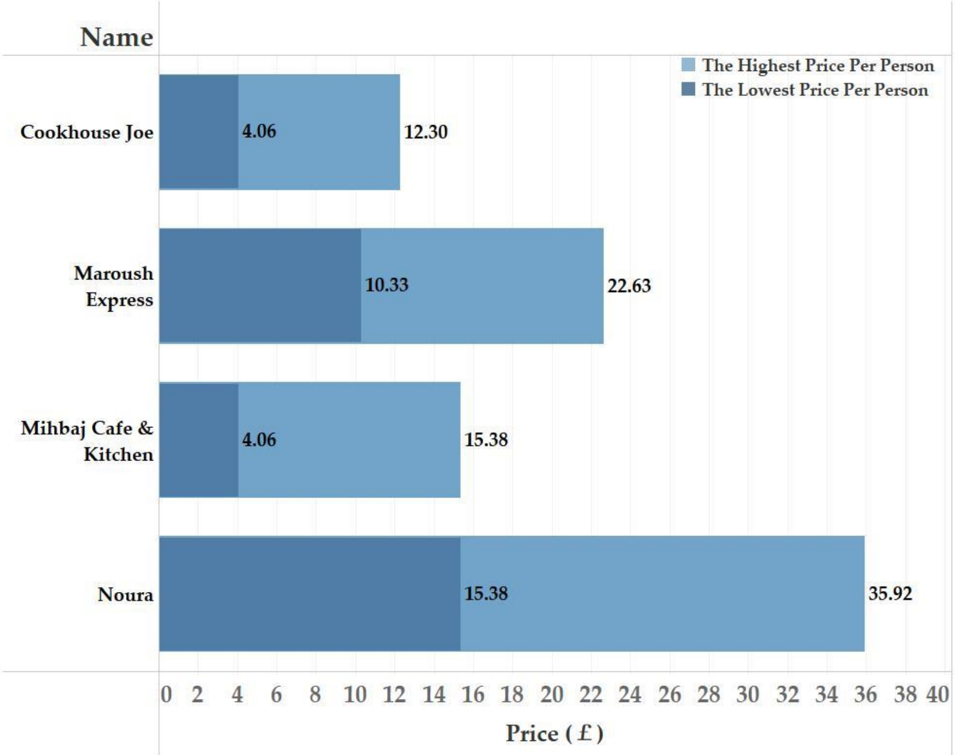


Figure 7. The highest and lowest price per person of the four restaurants.

Our work identifies general tokens and unique tokens for Mihbaj, as shown in Table 3. General tokens are key information carriers that are important for all of the four restaurants, while unique tokens contain some Mihbaj-specific dish names, as well as specific features. Mihbaj's all reviews are divided into sentences. We first filter out those without tokens and duplicates using ReviewFilterOne. Before the second filtering, 200 of the 20% of the sentences are randomly selected and sorted into informative class and uninformative class by two professionals. In the CNN training, when the accuracy of the validation set does not decrease anymore, we stop training. The remaining sentences are classified by the classifier and 730informative sentences are obtained for the candidate tips. We input tokens and the candidate sentences into SentenceSelector in the algorithm, and we get the general tip set consisting of 26 sentences and the unique tips set consisting of 38 sentences. By sorting out some general tips whose opinions have already been covered by the unique tips, the final tipset can be presented to readers. Part of the examples with tokens highlighted in bold type are as follows:

- **Avocado** on **toast**, many kinds of **coffee** preparations (**flat white**, Italian varieties), a little **patio** outside, **wifi**, friendly (if shy) **staff**.
- There's some less hectic **seating downstairs**, featuring a comfy **couch**, an old **fireplace**, a few **tables**, and the best part—a pair of **power outlets** anywhere you might sit, in addition to the free (secure) **WIFI**.
- Great **coffee** and **coconut porridge**, **nata tarts** and **avocado mash** on **bread**.
- Friendly **staff**, nice **cakes**, nice **coffee**, average **prices**, nice **decor**, lots of individual **seating**.
- **Espresso** was fine but the **panino** was really poor essentially while the **bread** was ok but the ingredients in this case **tomato** and **mozzarella** were minimal and totally inadequate.
- The **environment** is unique with a beautiful **art gallery downstairs**.

Table 3. Basic information on Mihbaj and its similar entities.

Unique tokens	sandwich, pastry, muffin, aubergine salad, soy milk, music, croissant, garden, snack, back yard, nata tart, pastel nata custard tart, art gallery, bean, cake, shakshuka, downstairs, coconut porridge, smoking area, toilet, chocolate, chocolate milkshake, mushroom, toast, eggs benedict, espresso, avocado, avocado mash, scone, guacamole, pancake, barista, gallery, flat white, feta cheese, almond, almond croissant, tart, pecan tart, panino, orange juice, sourdough, cappuccino, patio, paprika, prawn Kabseh, mozzarella, blueberry, aesthetic, couch, charging point, lactose, fireplace, power outlet
General tokens	food, service, coffee, atmosphere, price, drink, juice, salad, menu, bread, decor, decoration, table, tomato, serving, vibe, sauce, hummus, mint tea, chicken, sausage, manager, kitchen, cheese, wrap, WIFI, halloumi, staff, environment, waitress, falafel, chai latte, leaf, interior design, prawn, water, seat, pecan pie, chef, cookie, tea, milk, pie, latte, egg, seating, interior

5. Evaluation

5.1. Evaluation of the Usefulness and Novelty of Tips

To evaluate the usefulness and novelty of the TipScreener-generated tips, we designed two experiments and crawled reviews of three restaurants with different types of restaurant dishes from Section 4 to generate their sets of tips. The basic information of each restaurant is shown in Table 4.

5.1.1. Study A1

In this study, we hired four participants to read the top reviews of the four restaurants on TripAdvisor pages and asked them to select q review sentences they found useful. q equals the number of all tips generated by TipScreener of the corresponding restaurants. The sentences chosen by these participants are called participant-selected sentences. Usefulness is defined as whether it can help them make decisions quickly.

Experimental group: four participants were recruited, divided equally into four groups, each corresponding to a restaurant. They read participant-selected sentences and then tips that belong to their restaurant. For each tip, the participants were asked to assign a novelty rating on a Likert scale from 1 to 5, with higher values representing more favorable ratings. Novelty is defined as whether information that is not covered by the first batch of sentences is presented.

Control group: another four participants were recruited and equally divided into four groups, again, one restaurant for each group. The difference is that they read tips first, then participant-selected sentences. The novelty degree on a Likert scale from 1 to 5 was assigned for each of the participant-selected sentences.

Table 4. Basic information of the three additional restaurants.

Name	Cuisines	Coordinates	Similar Entities (Coordinates)
215 Hackney	middle eastern, cafe	(51.5633018, −0.0733957)	New London Cafe (51.5469438, −0.098291)
			Blighty Cafe (51.5639806, −0.1029146)
			Piebury Corner (51.5509858, −0.1104161)
Roxie Steak	803 Fulham Road Fulham, London	(51.4758293, −0.2052097)	Macellaio RC South Kensington (51.4926256, −0.1774369)
			Roxie Steak—Putney (51.4599947, −0.2125896)
			Gordon Ramsay Bar and Grill—Park Walk (51.4859695, −0.1799002)
Sinabro	28 Battersea Rise, London	(51.4610708, −0.1644541)	Augustine Kitchen (51.4781781, −0.1694937)
			Chez Bruce (51.4459466, −0.1655742)
			Restaurant Gordon Ramsay (51.4853968, −0.1620078)
			28–50 Wine Workshop and Kitchen Chelsea (51.4858753, −0.1732108)

The category of the sentence was hidden from the eight participants, meaning that the participants did not know whether the sentence they were reading was tips or participant-selected sentences.

Results: Average novelty scores for both groups in each hotel are shown in Figure 8. It can be seen that the experimental group achieves better results, indicating that tips have a higher novelty degree. We also conducted an ordinary least square regression to explore whether the type of a sentence can affect readers' perceived novelty, with the novelty value of the sentence as the dependent variable, and the independent variable as binary, which is 1 representing it is resourced from tips, and otherwise, 0 representing it is a participant-selected sentence. The regression result shows that the effect of being a tip on the higher novelty degree is significant (see Table 5, β_1 = 0.251, $p < 0.001$).

Table 5. Result of Study A1.

	Coefficient		Standardized Coefficient	t	Significance
	B	SE	Beta		
(constant)	2.397	0.108		22.211	0.000
type of sentence [a]	0.789	0.153	0.251	5.169	0.000

[a] Dependent variable: the novelty of sentences.

5.1.2. Study A2

In this experiment, we mixed and shuffled the participant-selected sentences and tips from each restaurant in Study A1 to form four new sentence sets. We hired four participants for each sentence set. For each sentence, the participants were asked to assign a usefulness rating on a Likert scale from 1 to 5. We work out the scores of tips and participant-selected sentences for each restaurant. As shown in Figure 9, tips obviously perform better than participant-selected sentences on usefulness. Another least ordinary square regression over the samples is conducted, with the dependent variable as the usefulness rating and the independent variable the same as in Study A2. The regression result shows that the usefulness degree of tips is significantly greater than participant-selected sentences (see Table 6, $\beta_2 = 0.209$, $p < 0.001$).

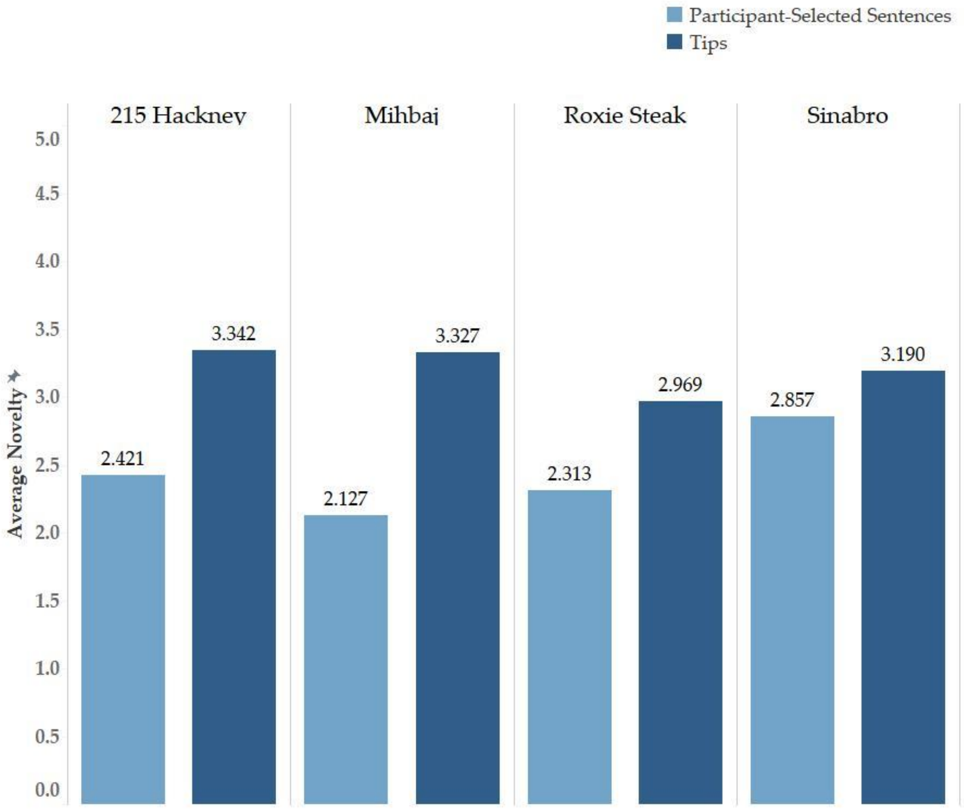


Figure 8. The novelty of tips and participant-selected sentences.

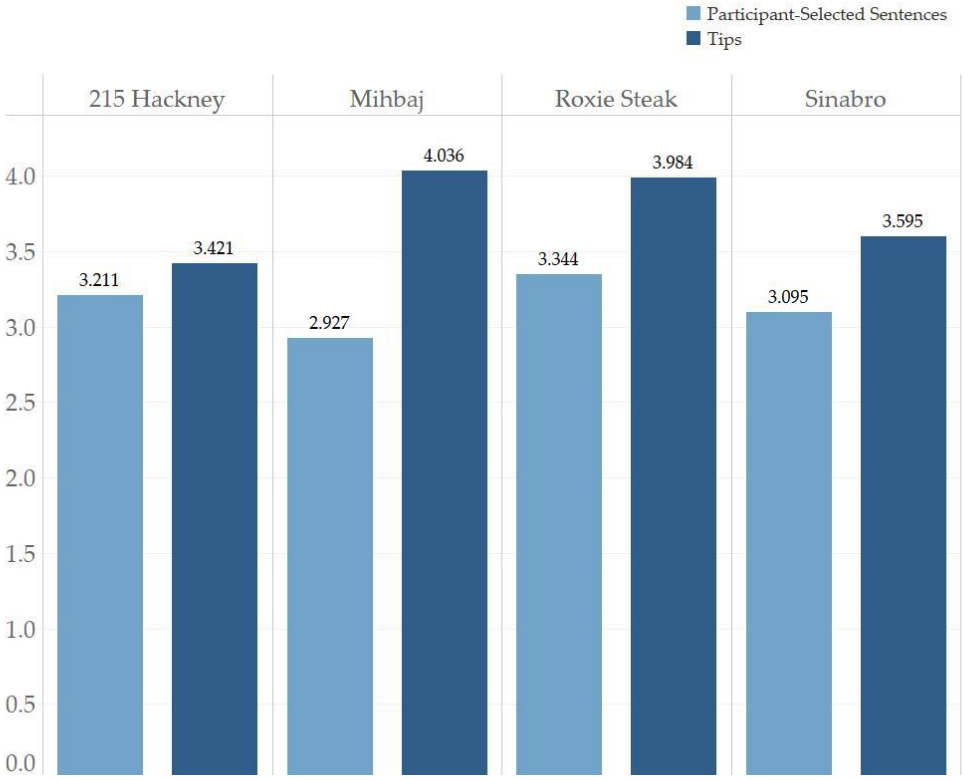


Figure 9. The usefulness of tips and participant-selected sentences.

Table 6. Result of Study A3.

	Coefficient		Standardized Coefficient	t	Significance
	B	SE	Beta		
(constant)	3.151	0.109		28.784	0.000
type of sentence [a]	0.658	0.155	0.209	4.252	0.000

[a] Dependent variable: the usefulness of sentences.

5.2. Evaluation of Matching Accuracy of Token-Opinion Words

In Section 3.2, we use a syntactic tree to match token and opinion words in each sentence. this allows the generated tips to have as few repetitive ideas as possible, allowing the reader to get more information in less time. In our framework, there are several possible errors in matching a token to an opinion word in a sentence: the token actually has no opinion word but incorrectly matches an opinion word; the token has opinion words but is shown to have no opinion word. The token incorrectly matches the opinion words of other tokens. For example, TipScreener will treat the word "coffee" in the sentence "Quality coffee at great prices" as a token without opinion words since "quality" rarely modifies nouns, but we intuitively think "quality" is its opinion word.

Let K be the number of tokens in a sentence, we first randomly selected30 sentences each with $K = \{1, 2, 3, 4, 5, 6\}$ tokens from the set of review sentences filtered by ReviewFilterOne for the above-mentioned four restaurants. Let TipScreener determine the token-opinion word pairs among them. An annotator is asked to match the actual token-opinion word for each sentence. For each value of K, we count the number of tokens that are incorrectly matched. No matter how many token-opinion word pairs a token has, as long as it is not matched exactly correctly, we treat it as an error. We did not evaluate the accuracy on a larger value of K since the vast majority of review sentences contain tokens less than seven. The results are shown in Table 7.

Table 7. Results of the matching accuracy of token-opinion words.

	Err = 0	Err = 1	Err = 2	Err = 3	Err = 4	Err = 5	Err = 6
$K = 1$	30	0	0	0	0	0	0
$K = 2$	27	2	1	0	0	0	0
$K = 3$	27	2	1	0	0	0	0
$K = 4$	26	3	1	0	0	0	0
$K = 5$	28	2	0	0	0	0	0
$K = 6$	29	1	0	0	0	0	0

The results show that for each K value, most sentences have zero errors and no sentence has more than two errors, indicating that they all correctly match opinion words for each token. We initially thought that a larger K value would lead to more errors, but observed that the number of perfectly matched sentences showed a U-shaped trend as the value of K increased. This may be due to the fact that sentences with four tokens are usually complex in structure, while those with five or six tokens are relatively simple, e.g., "Friendly staff, nice cakes, nice coffee, average prices, nice decor, lots of individual seating."

5.3. Comparison with The Baseline Algorithm

We compare TipScreener with TipSelector, the state-of-the-art for tip mining proposed by Zhu et al. [8]. Our implementation follows the process described in the original paper. The main difference between TipSelector and our algorithm is two-fold: (1) TipSelector only considers unique tokens that have a significantly higher frequency in the reviews of a business b than in any of its similar businesses. However, our work believes some general tokens are useful to review readers. (2) TipSelector results in an optimal set of tips with only two objectives: the fewer sentences in the set the better, and to cover all tokens. Our algorithm takes the role of opinion words into account. Comparing "I like the steak" and "the steak was flavorful and cooked as ordered", it is clear that the latter gives more information since it contains some opinion words. Therefore, we compared tokens identified by the two tip mining frameworks and the informativeness of the tips.

5.3.1. Evaluating Tokens Identification

Firstly, we use each algorithm to generate tokens for the four aforementioned restaurants: Mihbaj, 215 Hackney, Roxie Steak, and Sinabro. The similarities of these restaurants have been presented in Tables 2 and 4. Our results show that the unique tokens identified by TipScreener are highly consistent with all of the tokens identified by TipSelector. We evaluated tokens in the difference set (i.e., tokens identified by TipScreener but not by TipSelector), which are attributes that occur frequently in similar restaurants but are important. The general tokens in Table 3 are some examples. Tokens such as service, coffee, atmosphere, and price, which occur frequently in the catering industry are very important and should not be ignored. Table 8 shows some of the sentences with tokens in the difference set. As can be seen, tokens that are not recognized by TipSelector also convey a wealth of information.

Table 8. Examples of tokens in the difference sets and tips.

Name	Difference Set of Tokens	Example Tips
Mihbaj	staff, coffee, drink, etc.	The **staff** were very respectful and coordinated well. Good **coffee** and excellent healthy smoothie **drinks**.
215 Hackney	food, vegan, gluten free, etc.	The **food** was really fresh, well presented and delicious. If you're **vegan**, **gluten free** or looking for a healthy meal, this place is for you!
Roxie Steak	burger, pudding, etc.	Our nine-year-old girls took the adult sized **burger** which was fine for them. It was good except the **pudding** let it down.
Sinabro	food, take away service, etc.	The **food** is French in technique but very modern in interpretation. We have tested their new **take away service** doubting it would be easy to take away the chef and quality of the service.

5.3.2. Evaluating Tip Informativeness

First, we identified the tips of the four restaurants mentioned above using the two frameworks. We kept only unique tips when using the TipScreener method so that the tokens included in both were the same. We recruited four annotators, each responsible for one restaurant. They were given access to the corresponding restaurant's dedicated webpage on TripAdvisor, which included reviews of the restaurant and information about the cuisine. At each time, they were given two tips generated by the two algorithms for the same token. We hid the identity of the algorithms and randomized the order in which they were evaluated. For each token, they need to determine which tip has higher informativeness. We get the results shown in Figure 10, where it is obvious that TipScreener generates tips with higher informativeness at an overall level.

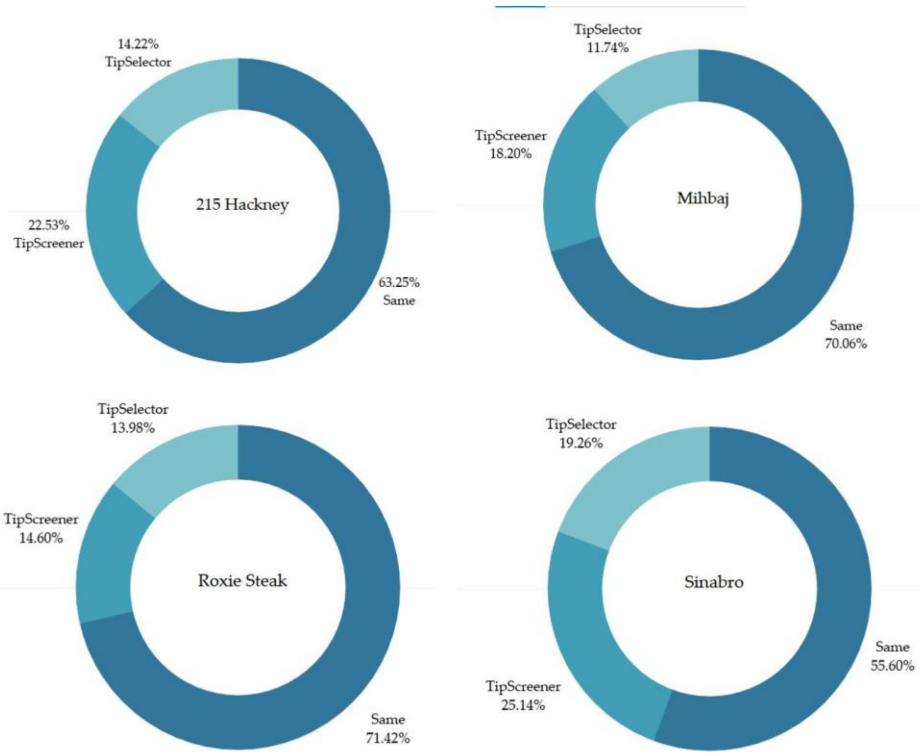


Figure 10. Informativeness of tips generated by the two algorithms.

6. Conclusions

With the explosion of information and data on e-commerce platforms, there is a dizzying array of advertisements, merchant promotions and user comments that are beyond what individuals can accept, process, or effectively use. Customers and even business managers reading user-generated content receive more information than they need and can process. Therefore, the question concerning how to manage repetitive and redundant user-generated content, especially user reviews in e-commerce platforms, has become a great challenge for information management. In order to help platforms organize reviews and comprehensively present them to readers in a more concise and clear manner, this study is designed to propose a framework to generate 'tips', which are useful and complete snippet sentences selected from online reviews of a product or a business. The tips can convey information that people are concerned about by matching review sentences and information-carried words (tokens) in this work. Called TipScreener, the framework has several functions:

(1) Tokens identification. The framework identifies unique tokens and general tokens by comparing the frequency of words that appear in the current reviews and reviews of other similar entities. The first one includes sentences delivering information about the unique features of the business. The second one includes sentences delivering information on features that are also shared by other similar businesses.

(2) Filtering uninformative sentences. The framework filters out sentences that fail to deliver useful information to readers through a CNN classifier.

(3) The output of the framework finds the sentence sets with the smallest cardinality, covering all of the unique tokens and general tokens, and uses opinion words to improve the quality of sentences. These sentences are unique tips and general tips. In unique tips, some general tokens may be covered. However, as the number of tips generated is generally small, some general tips with similar views can be manually removed.

In Section 4, we use TipScreener to find the similar entities of a Lebanese restaurant with three other restaurants, identify its unique tokens and general tokens, and mine its tips from its review data on TripAdvisor. In Section 5, we used four restaurants with different cuisines to conduct two experiments on tips generated by TipScreener, and found that tips are superior to the useful sentences selected by participants from comments in terms of novelty and usefulness. Since participants chose the sentences they considered useful from the default ranking of reviews on TripAdvisor's interface, a lot of repetitive information was generated. However, the generation of each tip takes into account whether the previous selection contains the current information. Since we improved the quality of tips through opinion words, we also verified the matching accuracy of token-opinion words, which showed a low error rate.

The study makes the following practical contributions. First, our framework provides e-commerce platform manipulators with a way to manage review information: sorting out the most useful and less repetitive information from a large number of user reviews and increasing the readability of the review interface. Second, based on the tips generated by the framework, business managers can understand customers' attitudes towards product or service features in detail and make enhancements. Third, if the platform can adopt TipScreener or adapt to the ideas of the framework, customers can quickly use Tips to compare the strengths and weaknesses of different businesses, identify their characteristics, and make effective decisions. Fourth, we used CNN to filter uninformative sentences in the framework, and although we do not elaborate on its efficiency in the evaluation section, we also verified the accuracy on the data of Section 4 beyond the paper, which enriches the study of CNN on text classification. Our study also makes academic contributions. First, our study improves on the algorithm designed by Zhu et al. [8]. Our algorithm generates tips that include both information unique to the subject of study and information shared with similar entities. We also consider the matching of opinion words with tokens to make the generated tips more informative, which provides a new research idea related to opinion mining based on user-generated content. Second, our research framework involves the application of deep learning to enrich the study of CNN for text classification on whether sentences are informative or not. Third, our evaluation method is innovative. We compare tips and participants-selected sentences in terms of novelty and usefulness, providing a new perspective for future related research to evaluate the results of tips mining.

Our research has several limitations and should be improved in the future. First, in the similar restaurant discrimination, due to the lack of online data, we only used location and cuisine as the discriminative features. However, other factors such as price and business description may also be used by customers as a basis for comparison. Future research can be improved with other methods. Secondly, we have filtered tokens that are smaller than a specific value, but sometimes these tokens may be very rare features that are difficult to be noticed by other customers but can also be very important for merchants to enhance their products. Investigations concerning how to ensure that these niche features are not excluded are also very valuable. Third, we tried to use opinion words to improve the quality of tips. If the tokens are already present and opinion words are synonyms, they are not counted in the number of tokens in the current sentence. However, non-synonyms may express the same meaning in some contexts, and later studies can refine this shortcoming.

Author Contributions: Supervision, H.L.; Conceptualization, H.L. and W.Z.; Funding acquisition, H.L.; Methodology, H.L., W.S. and W.Z.; Validation: writing—review and editing, H.L.; Software, W.S.; Validation, W.S.; Investigation, W.S. and W.Z.; Resources, W.Z.; Data curation, W.Z.; Writing—original draft preparation, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Discipline Co-construction Project for Philosophy and Social Science in Guangdong Province (No. GD20XGL03), the Universities Stability Support Program in Shenzhen (No. 20200813151607001), the Major Planned Project for Education Science in Shenzhen (No. zdfz20017), "Liyuan Challenge-Climbing Peak" Fund Project of Shenzhen University in 2022, and the Postgraduate Education Reform Project in Shenzhen University in 2019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Algorithm A1. TipScreener

```
1. Input: set of businesses $B$, set of customer reviews $R_B$ composed of subsets $R_b$,$\forall b \in B$, similarity function $f(\cdot)$.
2. Output: two sets of tips ($S_b^{unique}$ and $S_b^{general}$) for each business $b \in B$.
3. for $b \in B$ do
    4. $B_b^m = \underset{b' \in B \setminus b}{\text{argmax}_m} f(b, b')$#The set of similar businesses
    5. Find token set $T_b$
    6. Find opinion words set $O_b$
    7. $S_b^1$= ReviewFilterOne($R_b$, $T_b$) #To filter sentences containing no token and remove duplicates.
    8. $S_b^2$ = ReviewFilterTwo($S_b^1$) #To filter unhelpful sentences despitecontaining tokens.
    9. $D_b$ = { }#A"{token:frequency}" dictionary for $b$
    10. $X_b$ = { } #A"{sentence:tokens-list}" dictionary for $b$
    11. for $s$ in $S_b^2$ do
        12. for token $t$ in $T_b$ do
            13. $X_b[s]$.append($t$) #If $t$ in $s$, add t to the current tokenlist of $s$
            14. $D_b[t]$ = $D_b$.get($t$,0) +1 #Add 1 to the current count of $t$
15. for $b \in B$ do
    16. $T_b^{all} = \{t \in D_b : D_b[t] > k\}$,
    17. $S_b^{all}$ = SentenceSelector($T_b^{all}$, $X_b$, $O_B$)
    18. $T_b^{unique} = \{t \in D_b :\ D_b[t] \gg D_{b'}[t],\ \forall b' \in B_b^m\}$#A unique token set
    19. $S_b^{unique}$ = SentenceSelector($T_b^{unique}$, $X_b$, $O_B$) #Final unique tip-set
    20. $S_b^{general}$ = $S_b^{all} - S_b^{unique}$#Final general tip-set
21. Return $\mathrm{S}_{\mathrm{b}}^{\mathrm{unique}}$, $\mathrm{S}_{\mathrm{b}}^{\mathrm{general}}$, $\forall$b $\in$ B
```

Line 1: The inputs to the framework are a set of businesses B, a set of customers' reviews R_B composed of subsets R_b, $\forall b \in B$, a similarity function or a set of discrimination rules $f(\cdot)$ for identifying similar businesses b can be any product or service in any industry, but their reviews have to come from the same platform or from platforms with similar review specifications, otherwise, the results will be affected. This is due to the fact that different platforms will provide different dimensions or keyword templates when guiding customers to write opinions, which will result in different review concerns and systematic errors in subsequent token frequency statistics.

Line 3–6: For each b in set B, we find m similar b' according to $f(\cdot)$. In Section 3.1 we will show in detail the similar enterprise judgment methods we use in our examples. Tokens of similar businesses can help us separate unique tokens and general tokens of the business under study. The step can affect the two final tips-sets, one that shows how special b is in similar businesses and one that contains information about general features in the industry. $f(\cdot)$ can be a functional model or a set of reasonably custom rules. We also find tokens of b using a sentence parser and match their corresponding opinion words in every sentence. The definitions and acquisition of tokens will be discussed in Section 3.2.

Line 7–8: Our algorithm selects useful tips by mapping tokens (pieces of information carrier) with review sentences. If all sentences participate in the matching with the token set, it will require a lot of computation. Therefore, we first filter out all sentences without tokens using string search and duplicates (ReviewFilterOne), and then use supervised learning to initially exclude uninformative sentences that are unhelpful for readers (ReviewFilterTwo). In this study, ReviewFilterTwo is a CNN classifier.

Line 9–14: In TipScreener's token-sentence mapping process, we mainly use two dictionaries. D_b counts how many sentences a token appears in it in the business's comments. X_b stores the tokens each sentence contains.

Line 15–21: In these steps, TipScreener iterates over each b in B. In line 15, a set of token frequencies greater than k is generated. Generally speaking, the number of mentions of a word needs to exceed a threshold to be considered important or widely recognized by the public as a feature word. However, if the corpus is relatively small, this threshold cannot be set too high. In line 17, SentenceSelector maps T_b^{all} and X_b to find the set of sentences with the smallest cardinality covering all tokens of T_b^{all}. In human language, nuanced opinion words can convey ideas better than general positive and negative sentiment words. For example, when describing the meat quality of a chicken, the opinion word “succulent” is more appealing to the receiver than the positive sentiment word “like”. Therefore, we improve the information quality of the sentences in S_b^{all} by considering opinion words in O_B. Specific discussions of SentenceSelector will be presented in Section 3.6. Line 18 identifies the set T_b^{unique}, where the token appears much more frequently in R_b than in the reviews of any business similar to b. This step requires a two-by-two judgment of the Bbm set by setting the null hypothesis as “the token is equally likely to appear in the review sentences of both businesses”, and then conducting a Cochran–Mantel–Haenszel test. The same operation to line 17 is performed in line 19 to generate a set of tips S_b^{unique} that differs b from similar businesses (that is, the tips in S_b^{unique} contain information tokens that are significantly unique to b). Finally, in line 20, S_b^{unique} is subtracted from the total set of tips S_b^{all} to obtain set $S_b^{general}$, which delivers information on features common to all similar businesses. This set does not show the intersection of similar companies’ tips, but the token in it is shared by B_b^m. For example, the token “service” appears in many service industry’s reviews, so sentences that only contain this feature word are bound to not appear in S_b^{unique}. However, the description of service quality in $S_b^{general}$ differs from that of $S_{b'}^{general}$. Finally, the TipScreener returns $S_b^{general}$ and S_b^{unique} for each b of B.

References

1. Qi, J.; Zhang, Z.; Jeon, S.; Zhou, Y. Mining customer requirements from online reviews: A product improvement perspective. *Inf. Manag.* **2016**, *53*, 951–963. [CrossRef]
2. Hu, N.; Zhang, T.; Gao, B.; Bose, I. What do hotel customers complain about? Text analysis using structural topic model. *Tour. Manag.* **2019**, *72*, 417–426. [CrossRef]
3. Chevalier, J.A.; Mayzlin, D. The Effect of word of mouth on sales: Online book reviews. *J. Mark. Res.* **2006**, *43*, 345–354. [CrossRef]
4. Awad, N.F.; Ragowsky, A. Establishing trust in electronic commerce through online word of mouth: An examination across genders. *J. Manag. Inf. Syst.* **2008**, *24*, 101–121. [CrossRef]
5. Local Consumer Review Survey 2022. Available online: https://www.brightlocal.com/learn/local-sonsumer-review-survey/ (accessed on 17 October 2022).
6. Consumer Review Survey. Available online: https://www.brightlocal.com/research/local-consumer-review-survey/ (accessed on 17 October 2022).
7. Guy, I.; Mejer, A.; Nus, A.; Raiber, F. Extracting and ranking travel tips from user-generated reviews. In Proceedings of the 26th International Conference on World Wide Web, Perth, WA, Australia, 3–7 April 2017.
8. Zhu, D.; Lappas, T.; Zhang, J. Unsupervised tip-mining from customer reviews. *Decis. Support Syst.* **2018**, *107*, 116–124. [CrossRef]
9. Kumar, S.; Chowdary, C.R. Semantic Model to extract tips from hotel reviews. *Electron. Commer. Res.* **2020**, *22*, 1059–1077. [CrossRef]
10. Timoshenko, A.; Hauser, J.R. Identifying customer needs from user-generated content. *Mark. Sci.* **2019**, *38*, 1–20. [CrossRef]
11. Baccianella, S.; Esuli, A.; Sebastiani, F. Multi-facet rating of product reviews. In *Advances in Information Retrieval*; Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C., Eds.; Springer Berlin Heidelberg: Berlin, Germany, 2009; Volume 5478, pp. 461–472.
12. Chou, Y.-C.; Chen, H.-Y.; Liu, D.-R.; Chang, D.-S. Rating prediction based on merge-CNN and concise attention review mining. *IEEE Access* **2020**, *8*, 190934–190945. [CrossRef]
13. Wu, C.; Wu, F.; Liu, J.; Huang, Y.; Xie, X. ARP: Aspect-aware neural review rating prediction. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019.
14. Mahadevan, A.; Arock, M. Review rating prediction using combined latent topics and associated sentiments: An empirical review. *Serv. Oriented Comput. Appl.* **2020**, *14*, 19–34. [CrossRef]
15. Mahadevan, A.; Arock, M. Integrated topic modeling and sentiment analysis: A review rating prediction approach for recommender systems. *Turk. J. Electr. Eng. Comput. Sci.* **2020**, *28*, 107–123. [CrossRef]
16. McAuley, J.; Leskovec, J. Hidden factors and hidden topics: Understanding rating dimensions with review text. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12 October 2013.

17. Wang, B.; Chen, B.; Ma, L.; Zhou, G. User-personalized review rating prediction method based on review text content and user-item rating matrix. *Information* **2018**, *10*, 1. [CrossRef]
18. Lee, J.Y.-H.; Yang, C.-S.; Chen, S.-Y. Understanding customer opinions from online discussion forums: A design science framework. *Eng. Manag. J.* **2017**, *29*, 235–243. [CrossRef]
19. John, D.L.; Kim, E.; Kotian, K.; Ong, K.Y.; White, T.; Gloukhova, L.; Woodbridge, D.M.; Ross, N. Topic modeling to extract information from nutraceutical product reviews. In Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2019.
20. Kim, S.-M.; Pantel, P.; Chklovski, T.; Pennacchiotti, M. Automatically assessing review helpfulness. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, NSW, Australia, 22–23 July 2006.
21. Liu, J.; Cao, Y.; Lin, C.-Y.; Huang, Y.; Zhou, M. Low-quality product review detection in opinion summarization. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, Czech, 28–30 June 2007.
22. Zhang, Z.; Varadarajan, B. Utility scoring of product reviews. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, 6–11 November 2006.
23. Ghose, A.; Ipeirotis, P.G. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 1498–1512. [CrossRef]
24. Liu, Y.; Huang, X.; An, A.; Yu, X. Modeling and predicting the helpfulness of online reviews. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008.
25. Tsur, O.; Rappoport, A. Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In Proceedings of the International AAAI Conference on Web and Social Media, San Jose, CA, USA, 17–20 March 2009.
26. Lu, Y.; Tsaparas, P.; Ntoulas, A.; Polanyi, L. Exploiting social context for review quality Prediction. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010.
27. Cremonesi, P.; Facendola, R.; Garzotto, F.; Guarnerio, M.; Natali, M.; Pagano, R. Polarized review summarization as decision making tool. In Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces—AVI′ 14, Como, Italy, 27–29 May 2014.
28. Gong, Y.; Liu, X. Generic text summarization using relevance measure and latent semantic analysis. In Proceedings of the 24th annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, USA, 9–13 September 2001.
29. Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004.
30. Erkan, G.; Radev, D.R. LexRank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.* **2004**, *22*, 457–479. [CrossRef]
31. Luhn, H.P. The automatic creation of literature abstracts. *IBM J. Res. Dev.* **1958**, *2*, 159–165. [CrossRef]
32. Edmundson, H.P. New methods in automatic extracting. *J. ACM* **1969**, *16*, 264–285. [CrossRef]
33. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 22–25 August 2004.
34. Rapacz, S.; Chołda, P.; Natkaniec, M. A method for fast selection of machine-learning classifiers for spam filtering. *Electronics* **2021**, *10*, 2083. [CrossRef]
35. Michalopoulos, D.; Mavridis, I.; Jankovic, M. GARS: Real-time system for identification, assessment and control of cyber grooming attacks. *Comput. Secur.* **2014**, *42*, 177–190. [CrossRef]
36. Xiong, X.; Liu, Y. Application of quadratic dimension reduction method based on LSA in classification of the Chineselegal text. *Chin. Electron. Meas. Technol.* **2007**, *10*, 111–114.
37. Social-media-based public policy informatics: Sentiment and network analyses of U.S. immigration and border security. *J. Assoc. Inf. Sci. Technol.* **2017**, *68*, 2847. [CrossRef]
38. Figueira, O.; Hatori, Y.; Liang, L.; Chye, C.; Liu, Y. Understanding COVID-19 public sentiment towards public health policies using social media data. In Proceedings of the 2021 IEEE Global Humanitarian Technology Conference, Seattle, WA, USA, 19–23 October 2021.
39. Gallagher, C.; Furey, E.; Curran, K. The application of sentiment analysis and text analytics to customer experience reviews to understand what customers are really saying. *Int. J. Data Warehous. Min.* **2019**, *15*, 21–47. [CrossRef]
40. Luo, J.; Qiu, S.; Pan, X.; Yang, K.; Tian, Y. Exploration of spa leisure consumption sentiment towards different holidays and different cities through online reviews: Implications for customer segmentation. *Sustainability* **2022**, *14*, 664. [CrossRef]
41. Geetha, M.; Singha, P.; Sinha, S. Relationship between customer sentiment and online customer ratings for hotels—An empirical analysis. *Tour. Manag.* **2017**, *61*, 43–54. [CrossRef]
42. Jiang, S.; Pang, G.; Wu, M.; Kuang, L. An improved K-Nearest-Neighbor algorithm for text categorization. *Expert Syst. Appl.* **2012**, *39*, 1503–1509. [CrossRef]
43. Tago, K.; Takagi, K.; Kasuya, S.; Jin, Q. Analyzing influence of emotional tweets on user relationships using naive bayes and dependency parsing. *World Wide Web* **2019**, *22*, 1263–1278. [CrossRef]
44. Sánchez-Franco, M.J.; Navarro-García, A.; Rondán-Cataluña, F.J. A naive bayes strategy for classifying customer satisfaction: A study based on online reviews of hospitality services. *J. Bus. Res.* **2019**, *101*, 499–506. [CrossRef]

45. Moraes, R.; Valiati, J.F.; Gavião Neto, W.P. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Syst. Appl.* **2013**, *40*, 621–633. [CrossRef]
46. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text classification algorithms: A survey. *Information* **2019**, *10*, 150. [CrossRef]
47. Li, P.; Wang, D.; Wang, L.; Lu, H. Deep visual tracking: Review and experimental comparison. *Pattern Recognit.* **2018**, *76*, 323–338. [CrossRef]
48. Zhang, Y.; Wallace, B. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Taipei, Taiwan, 27 November–1 December; Asian Federation of Natural Language Processing: Taipei, Taiwan, 2015.
49. Nguyen, T.H.; Grishman, R. Relation extraction: Perspective from convolutional neural networks. In Proceedings of the 2015 Conference North American Chapter Association Computer Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015.
50. Chiu, J.P.; Nichols, E. Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 357–370. [CrossRef]
51. Dos Santos, C.N.; Gatti, M. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of the 25th International Conference Computer Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014.
52. Zhu, Q.; Jiang, X.; Ye, R. Sentiment analysis of review text based on BiGRU-attention and hybrid CNN. *IEEE Access* **2021**, *9*, 149077–149088. [CrossRef]
53. Mottaghinia, Z.; Feizi-Derakhshi, M.-R.; Farzinvash, L.; Salehpour, P. A review of approaches for topic detection in twitter. *J. Exp. Theor. Artif. Intell.* **2021**, *33*, 747–773. [CrossRef]
54. Jacob, M.S.; Selvi Rajendran, P. Fuzzy artificial bee colony-based CNN-LSTM and semantic feature for fake product review classification. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6539. [CrossRef]
55. He, Y.; Sun, S.; Niu, F.; Li, F. A deep learning model enhanced with emotion semantics for microblog sentiment analysis. *Chin. J. Comput.* **2017**, *40*, 773–790.
56. Zhou, Y.; Zhang, Y.; Cao, Y.; Huang, H. Sentiment analysis based on piecewise convolutional neural network combined with features. *Comput. Eng. Des.* **2019**, *40*, 3009–3013+3029.
57. Meyer, D.; Leisch, F.; Hornik, K. The support vector machine under test. *Neurocomputing* **2003**, *55*, 169–186. [CrossRef]
58. Hochreiter, S.; Schmidhuber, J. Long Short-Term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
59. Harris, Z.S. Distributional structure. *WORD* **1954**, *10*, 146–162. [CrossRef]
60. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*. [CrossRef]
61. Grishman, R.; Sundheim, B.M. Message understanding conference-6: A brief history. In Proceedings of the 16th International Conference on Computational Linguistics, Copenhagen, Denmark, 5–9 August 1996.
62. Pasca, M.; Lin, D.; Bigham, J.; Lifchits, A.; Jain, A. Organizing and searching the world wide web of facts-step one: The one-million fact extraction challenge. In Proceedings of the 21st National Conference on Artificial Intelligence, Boston, MA, USA, 16–20 July 2006.
63. Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *Lingvisticæ Investig.* **2007**, *30*, 3–26. [CrossRef]
64. Nothman, J.; Ringland, N.; Radford, W.; Murphy, T.; Curran, J.R. Learning multilingual named entity recognition from Wikipedia. *Artif. Intell.* **2013**, *194*, 151–175. [CrossRef]
65. Grimmer, J.; Stewart, B.M. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Anal.* **2013**, *21*, 267–297. [CrossRef]
66. Charniak, E. Statistical parsing with a context-free grammar and word statistics. In Proceedings of the Fourteenth NATIONAL Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence, Providence, RI, USA, 27–31 July 1997.
67. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 26–28 August 2014.
68. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
69. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.