DOI: 10.15439/2016F594
ACSIS, Vol. 9. ISSN 2300-5963

DAILY TOURISTIC PLAN RECOMMENDATION USING TEXT MINING

Kerem Turgutlu
Istanbul Technical University
Email: keremturgutlu@gmail.com

Erkan Isikli, PhD
Istanbul Technical University
Email: isiklie@itu.edu.tr

"This study focuses on the proposal of a recommender system for daily touristic plans. In order to construct such a system it is further examined that there is a need of text mining applications. Moreover, Sentiment Analysis and Keyword Extraction techniques are evaluated by developing and testing different approaches. Sentiment Analysis approaches are examined step-by-step in order to pick the best among them to score restaurant data. Similarly, Keyword Extraction is evaluated from various perspectives of statistics, visualization and machine learning. By the end of the paper the structure and the flow of the proposed system is illustrated upon the chosen approaches which were tested throughout this paper."

I. Introduction

THE main aim of this study is to create a dynamic environment that enables users to find the best activities and dining options on real time around a desired travel destination. The project tackles certain problems that many travelers face in a day-to-day basis such as wasting unnecessary amount of time planning what to do in a touristic area and struggling with the difficulty of finding activities or dining places which suits them best. Inadequacy of conventional travel websites, blogs, reviews and their lack of simplicity due to large amount of data makes such decisions time consuming.

In order to create such a system which allows users to pick their interests from a text cloud and later allows them to choose from the recommendations of auto-generated day plans considering their needs, desires and budgets, one can make use of two major text mining techniques: *Sentiment Analysis* and *Keyword Extraction*.

The scope of this idea of creating a smart online travel day recommendation system is considerably wide as there are thousands of travel destinations worldwide. For simplicity and testing matters, only the restaurants in Amsterdam area are considered at the modeling stage, but it should be noted that each and every step of this study can be applied to any touristic attraction in every travel destination as long as there is available data online. All datasets are generated by collecting user reviews from TripAdvisor due its simplicity of API integration and cost efficiency. Additionally, later during the product development stage of the proposed recommender system, many other review sources will be combined to elaborate the findings [1].

II. TEXT MINING

The "Text Mining" is often generalized as processing structured or unstructured but information holding data for the sake of generating patterns of information [12]. Natural Language Text is the major study and analysis source of Text Mining, thus exponentially growing web-based textual information makes it more attractive every passing year. There are various ways of analyzing textual documents via text mining, as well as many types of information gathered by using its techniques.

For both data and text mining it is primarily important to analyze or process a data which potentially holds information. In other words the actions taken by data and text mining tools should be in a way that it illustrates or generates an information from a given data. Aside from this mutual need of having and analyzing potentially useful data; text and data mining differ substantially when it comes to type of data that is used. Data mining deals with incomprehensible data with binary, nominal, ordinal and interval features which only deploy a meaning when certain algorithms are used or statistical interpretations are made. On the other hand, text mining data is already comprehensible and gives a textual information even without processing it. This uniqueness of explicit information bearing puts text mining one step ahead [12]. Nonetheless, for both cases detecting an informative pattern is equally tricky since both data mining features and text mining data are incomprehensible to computers or machines which tries to interpret them.

This uniqueness of text mining is the reasons that it is adopted at the core of this project. Many traveling sites and other sources offer direct information about restaurants or touristic attractions but this information generally does not go beyond cost, address, opening-closing hours and other type of strict data. Most of the time what real travelers seek during their explorations are fast recommendations which suits them best usually from their close network of friends and families. Our aim goes beyond the limited circle of friends and families, considers millions of available reviews.

III. SENTIMENT ANALYSIS

Sentiment analysis or opinion mining is the task of identifying the subjectivity of a document and later determining its class as being; neutral, positive or negative.

Sentiment analysis is widely used in business related domains such as; marketing, customer satisfaction and benchmarking, as well as in political science, law, sociology and psychology [18].

Statistics or machine learning algorithms are used to classify the documents' sentiment. Even though state-of-art methods and algorithms tend to give satisfying results, sentiment analysis is a difficult task when the complexity of people expression, unrelated lexical content, negations and rhetorical devices as irony and sarcasm is considered [18]. Deciding positivity and negativity of a text may come out differently with errors even when it is done manually by 2 different human candidates. This shows the complexity of human expressions and perceptions. As the complexity of the domain and opinion increases the more difficult the task becomes. Relatively sentiment analysis on a product is easier to a political opinion [18]. Moreover in this paper sentiment analysis on restaurants is studied, using machine learning methods and probabilistic approaches.

Particularly in this paper, the main objective had been to help tourists to be able to have the optimal day plan with significant amount of time saving. In order to guarantee them to have the best possible experience: each restaurant and touristic attraction should be scored. Different modeling approaches are studied and evaluated step-by-step. Moreover, tourist reviews are collected from *TripAdvisor* and processed by applying sentiment analysis using different methods and later picking the dominating method to score each restaurant and activity.

IV. Sentiment Analysis Application

A. *TF-IDF Approach I*

First part of the sentiment analysis is data collection. As mentioned earlier, the whole data (training and test together) is collected from TripAdvisor website under Amsterdam search. In the first model, only the 50 top-ranked restaurant reviews are used. The whole data consists of 11688 reviews in total, which are scraped by using Kimono API creator [11]. The features that are collected to be used in modeling are: *title of the review*, *review text*, *number of stars out of 5*, *total number of previous reviews made by the same user* and *the total number of helpful reviews that are made by the same user so far*. The last two features are not directly related with the sentiment analysis; in fact, they are not used in application, but they might be helpful to assign weights on each review once they are tagged as a positive or a negative review and may have an effect on the overall score of a restaurant. In total, there are 11027 positive and 661 negative reviews combined.

In this part, supervised learning techniques are employed and in order to apply these techniques effectively, labels are assigned as 1 for a positive review and 0 for a negative review. Many studies in the related literature suggest the use of an additional and a priori labeling which conditions on subjectivity or objectivity of the text. However, since we are dealing with customer reviews, we are almost certain that all the reviews are subjective at some point and does not state facts as in news articles or product descriptions. Hence, it would be convenient to bypass subjectivity analysis part assuming all the reviews are subjective and only use a binary labeling for the supervised learning. To do so, a simple code was written to automatically separate the reviews with less than 4 stars as negative reviews, from the positive reviews (those with 4 or higher stars). The title and review text were then concatenated as a single string. Finalizing the data set into two features as *full text* which has the text and as *sentiment* which contains the class labels 1 for a positive review and 0 for a negative review. Tail of the structured data is shown in Table 1.

TABLE I.
TAIL OF THE STRUCTURED DATA WITH FULL TEXT AND SENTIMENT FEATURES

	Full Text	**Sentiment**
11683	Worth every cent!. Went a little out of the ce...	1
11684	Very good experience.. We had a amazing night …	1
11685	EXCELLENT MEAL. Have been here before two year…	1
11686	Amazing. Went to this restaurant for boyfriend…	1
11687	Absolutely stunning throughout. Restaurant was..	1

In the next step before training the data set the text is preprocessed in order to get rid of the punctuation marks, to get rid of html markups, to deal with emoticons and with lowercase letters [7]. Even though punctuation marks address significance about the sentiment class identification it may lead the classifier into an unwanted direction, in which case "!" may be a negative or positive claimer [7]. In almost every NLP (Natural Language Processing) tasks tokenization is necessary and for this study each individual sentence is broken into words for further processing [6]. In the next step the reviews are converted into a feature matrix consisting rows for reviews and columns for each tokenized words. In this study features are single words - unigrams, but different n-grams could be chosen for different processing purposes [6].

After tokenization step, term frequencies of each single unigram-feature under each document is denoted by tf (t, d). An illustrative example 3 different document sentences is given in Figure 1.

{This: 1, restaurant: 2, is: 3, good: 4, bad: 5, ok: 6}

D1: [1, 1, 1, 1, 0, 0]
D2: [1, 1, 1, 0, 1, 0]
D3: [1, 1, 1, 0, 0, 1]

Fig 1. Each document is given with its feature matrix *tf (t, d)*.

The sentences in Figure 1 are read in this order: “This restaurant is good”, “This restaurant is bad” and “This restaurant is ok”.

Unimportant English words are overly weighted when dealing with term frequencies. In order the tackle this problem and give each feature a weight corresponding to its importance , a special form of feature vectorizer called *term frequency – inverse document frequency tf-idf (t, d)* is used [7, 8].

$$tf-idf(t,d) = tf(t,d) \times idf(t,d)$$

The inverse document frequency is calculated as follows:

$$idf(t,d) = \log \frac{n_d}{1 + df(d,t)}$$

In the last equation, n_d is the total number of documents and $df(d,t)$ is the number of documents that contain term *t*. Addition of term 1 in the denominator allows smoothing and deals with log expression [9].

Before further processing, the data is split into training and test sets with 0.67 to 0.33 ratio. Later, in Python a pipeline is constructed including a tf-idf vectorizer and a Logistic Regression classifier with L2 regularization and parameter C = 10 [7]. It should be noted that once the documents are converted into a feature matrix, any classifier might have been used. SVM, MaxEnt, Random Forests are commonly used classifiers for this purpose [2].

There are various machine learning metrics used to analyze how well a model performs: accuracy, precision, recall, and F measures [10]. Since the test set is pre-labeled as in every supervised machine learning model, a comparison between true classification labels and predictions could be made. To provide a better understanding, a confusion matrix is given in table II and related metric functions are provided.

TABLE II.
CONFUSION MATRIX

	Predicted NO	**Predicted YES**
Actual NO	True Negative	False Positive
Actual YES	False Negative	True Positive

Accuracy = TN + TP/ (TN + TP + FP + FN)
Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
F1 = 2 * (Precision * Recall) / (Precision + Recall)

The predictive value negative, which can be given by:

$$\frac{\textbf{True Negative}}{\textbf{True Negative + False Negative}}$$

This metric addresses the power of predicting negative reviews. Due to lack of negative reviews, under fitting occurred for this specific case. In the proceeding, more negative reviews are collected to improve predicting the negative reviews as well.

B. TF-IDF Approach II

After additional data collection, the number of negative sentiment reviews increased by 2100 and the model trained with the same parameters but with a larger data set.

Predictive value negative metric increased by more than 0.40 points as we anticipated and obtained as 0.7452. Nonetheless, this increment is not a pure improvement since the newly generated test data is different from the previous dataset both in scale and observations. Thus, in order to evaluate the performance in an unbiased way, the same splits from the old data were created and tested with the new model.

The performance of tf-idf model is increased by collecting more negative reviews, especially significantly in the power of predicting negative reviews with a point difference more than 0.30.

C. Boolean Multinomial Naive Bayes Approach I

According to many studies in the related literature, when it comes to sentiment analysis, Naive Bayes tends to give promising results. This approach has Bayes Theorem in its core and naive term comes from its simplicity due to the avoidance of the dependency of occurrence of each word. It is assumed that each word is independent [3 4 5]. The theorem basically argues that the class or the sentiment of a document is the maximum probability it gets, given such a document: sequence of words. The documents we are referring throughout this part are individual tourist reviews. To briefly illustrate steps of Naïve Bayes mathematically [3 4 6]:

C: Class, D: Document

1) Objective Function: $argmax[P(C|D)] \ \forall \ C$
2) Expand $P(C|D)$:

$$\frac{[P(D|C) * P(C)]}{P(D)}$$

$P(D)$ is a global constant. So, the numerator part is enough.

3)

$$[P(D|C) * P(C)]$$

$P(C)$: Proportion of class C upon all documents.

4) Expand $P(D|C)$:

D: documents composed of unigram tokens.

Represented as:

$$P(w_1, w_2, .., w_n|C)$$

$$w_n: nth\ word\ in\ document$$

Word occurrences are considered independent:

$$P(w_1|C) * P(w_2|C) *..* P(w_n|C)$$

5) Final estimator:

$$\mathrm{P(C)} * \prod_{\mathrm{i}} \mathrm{P}(w_i|\mathrm{C})$$

Log scaling is used in order to prevent floating points and to prevent excessive weights on frequently used words.

$$\mathrm{argmax}[\log(\mathrm{P(C)}) + \sum_{\mathrm{i}} \log(\mathrm{P}(w_i|\mathrm{C}))]$$

Boolean Multinomial Naive Bayes is a special case of Naive Bayes with steps:

1) Preprocess text.
2) Remove all duplicate words in each document.
3) Do Naive Bayes.

$$P(w_i, C) = (count(w_i, C) + 1)/ (count(C) + |V|)$$

|V|: vocabulary size

+1 and +|V| is for Laplace smoothing in order to avoid none observations.

Another problem that this method faces is the biased weighting due to count differences of each word in each class. To get over this mighty problem, likelihood approach is defined [6]:

$$\mathrm{P(w|C) = f(w, C)} / \sum_{\mathrm{C}} \mathrm{f(w, C)}$$

f: frequency of word w in class C.

Again the same dataset from TFIDF Part 2 is used but with a different sample size. In order to balance weights of each class and their effects on prediction, negative and positive review sample sizes are selected equally.

As a new data set; 2100 negative and 2100 positive reviews were merged. First the “stop words” in English language are removed. Then it is split into train and test data by 80% to 20% [7].

In the preprocessing step, regular expressions and stop words are removed and all the words are re-written in lower case. Next, sentences are broken into unigram word tokens. Finally in order to begin Boolean Multinomial Naive Bayes each duplicate of words in all sentences are removed.

An Out-of-Vocabulary (OOV) word is considered to appear equally likely on both class such as words like “restaurant”, “food”, etc. Thus, assigning a 50% likelihood to those unseen words when dealing with smoothing is important in order to avoid undefined operations like 0/0, log (0) or to avoid smashing all the chain probabilities to 0. This modification allows the algorithm to deal with unfamiliar and unrelated words equally [6].

Smoothing may be changed and tested accordingly to evaluations of the study. It is decided to assign 0.5 to likelihood for OOV words and to assign a very small number close to 0 such as 1e-15 for log (0) cases. Basically; given a word if that word is not in the given class but in the other class: assign 1e-15. If the given word is not appearing on both classes then assign 0.5.

Collective frequencies are simply the sum of the frequencies in both classes. Taking a word token into account. If it is not found in the negative class then it is set and labeled as “no found”, in smoothing step that expression is replaced with the minimum value 1e-15.

Naive Bayes has improved the previous model’s power of predicting sentiment for class 0 (negative) reviews. It used be around 0.6460 and after Naive Bayes Classifier it increased to 0.7845 boosting it up almost 0.15 points.

In general, Naive Bayes gives high hopes on both predicting positive and negative sentiment equally. Performance evaluations can be improved by collecting more data, improving preprocessing steps and applying fine tuning by using stratified k-fold [7].

D. Boolean Multinomial Naive Bayes Approach II

In this part of the Boolean Multinomial Naive Bayes, rather than using equal sized samples for both classes all of the data is used for better fitting. Bad/Good Ratio for all data is around 0.1788. Train and Test data is split by 80% to 20%. Further for better validation of the model, stratified k-fold cross validation is used where k = 5. By using cross validation training data is split into 5 different train and validation parts for best selection, while the bad/good ratio is preserved [7].

Using a data set composed of 1341 negative reviews and 7517 positive reviews was not enough explanatory when it comes to predict negative reviews. Previous part of this model outperformed this case. Equal sized samples with good/bad = 1 ratios tend to fit negative reviews better where this case with large positive dataset and low negative dataset performed only as good as fair coin toss. Here, the major problem is the convergence of the predicted sentiment of the reviews to a single class due to its higher weight and feature scale compared to the other class. As positive training samples dominate in size and negative samples lack to describe an unseen test sample; the minority class tend to converge into the dominating class. Since it does not represents all the features or word tokens.

In literature and from the previous parts of study more data collection can be considered as a better approach. But, at the same time enough descriptive information for each class should be collected in order to perform equally good in each case.

One might argue that Naive Bayes Algorithm can deal with unequal sized class samples by having P(C) class probabilities inside the formula [3, 4, 6]. Even though this is the case sometimes P(C) ratios can weight more than word frequency ratios.

For example, a sample review is examined:

Word "great" is a feature which should be considered as a positive identifier according to human logic and English language. But the following frequency values pulled from the model indicates the effect of "great" on prediction is not great as the class frequency itself, hence not making the impact it should have made.

Word "great":

Negative frequency: 0.004
Positive frequency: 0.014

"great positive frequency"/ "great negative frequency" = 3.5

"positive class frequency"/ "negative class frequency" = 5.6

Having 5.6 > 3.5 and also seen from this example, in some cases class ratios have more weights on the predictive algorithm than the actual descriptive words. So to finalize, for cases like restaurant review sentiment analysis with 2 classes it can be considered as a better approach to have balanced class samples. Additionally even if equal size samples are not favorable in some cases, there should be a threshold value to prevent overweight of a class frequency rather than the actual words.

Table below represents the model evaluations for each approach used in sentiment analysis case.

TABLE III
PERFORMANCE METRICS

	Accuracy	Precision	Recall	F1 Score	Predictive Value Negative
TF-IDF I	0.9567	0.9799	0.9953	0.9774	0.3363
TF-IDF II (More Data)	0.945	0.9766	0.9802	0.9681	0.7452
TF-IDF II (Same Old Test Data)	**0.9653**	**0.9886**	**0.9851**	**0.9816**	0.646
Bool. Multi. Naïve Bayes I	0.8155	0.8308	0.7845	0.807	**0.7845**
Bool. Multi. Naïve Bayes II	0.8985	0.9188	0.9655	0.9465	0.5283

V. RESTAURANT SENTIMENT SCORING

In overall, Multinomial Naïve Bayes Part 1 results outperformed other methods and in this section each individual restaurant review is predicted by that same classifier. Later positive and negative reviews of each restaurant are used in order to score them to create a ranked list of restaurants. Percentage of positive reviews are given to assign a score to restaurants. A short sample list of scored restaurants are given in Table IV as an illustration. The scores will allow the proposed recommender system to recommend top restaurants which are related with the user's interests.

TABLE IIV
RESTAURANT SCORES

Restaurant	Score
Arendsnest Dutch Beer Bar	0.8419
Bakers & Roasters	0.7663
Biercafe Gollem	0.8424
Bird Thai Snackbar	0.8242
Bord'Eau	0.9427
Brasserie Ambassad	0.8533
Brasserie SenT	0.7642
Brasserie Vlaming - Amsterdam	0.84
Braziliaans Grill Restaurant Rodizio.nl	0.2241
Broodje Bert	0.7895

VI. KEYWORD EXTRACTION

Information Extraction (IE) is widely used in order to get a smaller structured information from a document by using statistical analysis, machine learning and NLP (Natural Language Processing) techniques. IE also contains sub-tasks as NER (Named Entity Recognition), Semi-Structured IE, Terminology Extraction, Keyword Extraction and Audio Extraction. Interestingly, IE is not only used to extract from textual data but also involves studies in multimedia extraction. There are mainly three widely accepted methods to extract information from a document, which are Hand Written Regular Expressions, Classifiers as Naive Bayes and MaxEnt and finally Sequence Models as Markov Models or Conditional Random Fields. [23].

In this paper, keyword extraction plays a vital role on determining the characteristics of a particular restaurant and in the future implementations to determine the characteristics of a touristic activity. The main objective of extracting keywords or information from a particular domain is to find out which attributes describes that entity best in an optimal and efficient manner. Later these extracted keywords will be clustered into groups according their similarity in order to generate a word cloud. This word cloud or network of descriptive words will be represented to the users allowing them to pick words according to their interests. Word selection phase will later lead to generate an optimal day plan by co-working with the scores we obtained in the sentiment analysis phase.

Three different approaches are tested during keyword extraction task. These approaches are Rake Algorithm, KeyGraph Algorithm and machine learning approach by Random Forests respectively. Again, for this part the same restaurant data set from *TripAdvisor* is used for test and evaluation purposes. Additionaly, following applications are modeled and tested by datasets generated from *Arendsnest Dutch Beer Bar* only.

VII. Keyword Extraction Application

A. RAKE Algorithm

Rapid Automatic Keyword Extraction (RAKE) Algorithm is the first approach adopted. Its good statistical interpretation and computational effectiveness due to its fast nature makes it a desirable candidate.

RAKE involves the following steps:

1. Data Preparation and Processing
2. Candidate Keywords Generation
3. Keywords Selection

In data preparation step, tourist reviews of Arendsnest Dutch Beer Bar are concatenated in order to form a single document. This document holds the latest 215 reviews that are made recently. Later, in the preprocessing step regular expressions, stop words, html markups and emoticons are removed from the document, also all words are lowered.

Candidate keywords are the tokens we would like algorithm to statistically evaluate and later assign a desirable amount of them as true keywords. So, a smart interpretation should be made while generating candidate keywords from a document. Also, one must have a good prior knowledge about the domain which of the document which will be processed. Here, we are dealing with restaurants and in contrast to a news article or a scientific paper keywords would not be longer than three words. Most of the time a restaurant would be described by its cuisine or atmosphere. Hence, candidate keywords are appended to a list by generating 1, 2 and 3 – gram tokens. Each n-gram group later evaluated separately in order to avoid cross dominations across groups.

In the final step, two different methods are employed in order to select keywords from n-grams. Each method use a statistical interpretation in order to score candidates and later outputs the desired amount of keywords or the ones that are above a given threshold.

First method is Best Match (BM):

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i)[\frac{f(q_i,D)*(k_1+1)}{f(q_i,D)+ k_1*(1-b + b *\frac{|D|}{avgdl})}]$$

|D| = Document length in tokens
avgdl = Average document length in tokens
f (qi, D) = Frequency of candidate q in document D
k1, b = Free default parameters

Due to lack of an error function and thus lack of an optimization, default values are given as k1 = [1.2, 2] and b = 0.75. IDF part is ignored in calculation since the stop-list the common English words in the preprocess step. Also having a single document gave identical scores for every k value in the range. Top 5 keywords obtained by BM for each n-gram group is given in table V.

Table V
Top 5 keywords obtained by best match

Top 5		
1-gram	2-gram	3-gram
"beers"	"dutch beers"	"great beer selection"
"selection"	"dutch beer"	"dutch craft beer"
"place"	"beer selection"	"dutch beer bar"
"staff"	"great beer"	"great beer bar"
"friendly"	"beer bar"	"best beer bar"

Second method is TF-IDF:

$$score = \left(0.5 + 0.5 * \frac{f(t,d)}{max_t f(t,d)}\right) * \log(\frac{N}{n_t})$$

f (t, d) = frequency of term t in document d
N = number of documents
n_t = number of documents containing term t

Logarithmic part of the score equation represents the IDF and it is neglected since we are processing a single document. As a result after applying TF-IDF method, the same top 5 keywords obtained for every n-gram group as in the BM method. Both TF-IDF and BM gave identical results with one documents case. Our conclusion here is that term frequencies of keyword tokens are highly correlated with the fact of being a keyword. Term frequency is later adopted as an important feature in machine learning model approach. RAKE is an incomplex and fast algorithm which yields satisfying results.

B. Key-Graph Algorithm

Key-Graph Algorithm is a visual indexing tool that is used to represent the characteristics of a single document. The algorithm creates a visual map containing clusters of words according their frequencies and co-occurrences [24].

Key-Graph involves the steps below:

1. Data Preparation and Preprocessing.
2. Extracting Foundations.
3. Extracting Columns.
4. Extracting Roofs.

In the first step, same preparation and preprocessing is applied and additionally each word is stemmed by Porter Stemmer algorithm. Different from grouping keyword candidates as n-grams in Key-Graph the aim is to find long keywords. So a candidate keyword list is created by deriving 2 and 1- gram tokens from 3-grams. Later, candidate phrase list is sorted by their frequencies in decreasing order. As it is mentioned above a relatively longer keyword is more favorable to others in this algorithm. So, if 1 or 2-gram phrases have the same frequency as their parent 3-gram phrase, the algorithm automatically eliminates lower gram phrases from the candidate list. After this elimination step the sorted candidate phrase list is finalized. Top 50 candidates is chosen empirically for further steps.

Next, the association or co-occurrence scores of word pairs from the top 50 list is computed in order to cluster them by their scores.

$$assoc(w_i, w_j) = \sum_{s \in D} \min(|w_i|, |w_j|)$$

|w| = word count.

s = Sentence, a single tourist review in our case.

D = Document, collection of reviews of a restaurant.

Below is the pair association scores of pairs from our top 50 list. The pairs which are not included here have 0 association scores.

Words are clustered according to their scores as shown in figure 1 below. This is a multiply connected graph with two clusters, top group has association score of 2 and the other has score of 1. Dashed line connects two groups. Since each group has equal level of connection within themselves, 2!*3! = 12 different combinations can be maintained.

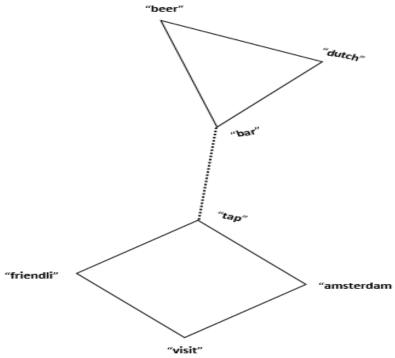

Fig 1. Extracted Foundations

Next, in order to extract columns to be added to foundation two functions are defined:

$$based(w, g) = \sum_{s \in D} |w|_s * |g - w|_s$$

$$neighbors(w, g) = \sum_{s \in D} \sum_{w \in s} |w|_s * |g - w|_s$$

w = words in top 50 list excluding the words in clusters.

g_i = graph including words in ith cluster

By based and neighbor scores key values of each word w is to be calculated as follows:

$$key(w) = 1 - \prod_{g \in G} \left(1 - \frac{based(w, g)}{neighbors(w, g)}\right)$$

Later top 5 ranking words w selected empirically according to their key scores. This key score represents the closeness to a cluster and to a specific word in that cluster. Words selected to be added to clusters as columns: [u'great', u'select', u'staff', u'place', u'tri']

These words are paired with words in graph clusters and scored according to column scoring function:

$$column(w_i, w_j) = \sum_{s \in D} \min(|w_i|_{s,} |w_j|_{s,})$$

Each column pair score resulted in 0 indicating no new term to be added next words in any of the clusters. Our final graph is the one we obtained in previous steps shown in fig 1.

C. *Machine Learning Approach by Random Forests*

The final approach used for keyword extraction is the machine learning classification task which intuitively modeled with the findings in the previous steps. For this classification task we define features which illustrates a keyword candidate [25]. Again, keyword candidates are constructed by forming 1, 2 and 3-gram word tokens. Features are described in table VI. Here each tourist review of a restaurant is considered as a single document and again *Arendsnest Dutch Beer Bar* dataset is used to make computations. As it is discussed in the beginning of the paper, each tourist review (document) has a title and a review text.

Since this is a supervised classification task the keywords for this sample is picked by volunteers and later used in the labeling stage. Categorical data is dealt by hot-encoding in order to be ready for training model. After necessary manipulation in data frame, it is decided that the keyword occurrence is a very rare event with a class ratio of 7:10000. Over sampling and under sampling may be applied in order to overcome the imbalance. Besides, decision tree classifiers tend to give promising results by generating rule based algorithms. Next, the data is split having 2000 observations of test data and around 9000 training data. They have 5 and 3 keywords in their samples respectively.

CART Decision Tree Algorithm is used in training. Testing the model gave 100% results in all the following metrics: accuracy, precision, recall and F1 score. All the keywords are predicted correctly.

TABLE VI
FEATURES OF A KEYWORD CANDIDATE

	Explanation	Type
Name	Name of the candidate	String
TF (Term Frequency)	Total count of the candidate / Max count in Document	Numerical
IDF (Inverse Document Freq.)	# of documents having the candidate/ total # of dociments	Numerical
TOR (Title Occurrence Ratio)	# of titles having the candidate/ total # of titles	Numerical
ROR (Review Occurrence Ratio)	# of reviews having the candidate/ total # of reviews	Numerical
POSS (Part-Of-Speech Sequence)	Part-of Speech Sequence Tag ex. {NN}	Categorical
Ngram (N-gram Tag)	Unigram, Bigram or Trigram	Categorical
Keyword (Target Value 1-0)	if the candidate is a keyword 1; else 0	Binary

VIII. RESULTS & DISCUSSIONS

All of the approaches in keyword extraction part has their own advantages and disadvantages. RAKE is very fast and easy but does not go beyond picking frequent words and does not take other features into account. Key-Graph is a strong visualizer which allowed us to see relationships between words but its expertise is not a primary concern for case since restaurants are described by relatively shorter independent sequence of words. Our final approach, machine learning by decision trees is the most promising one due to its high performance scores. In contrast, it is cost expensive in the terms of computing features of large candidate set and labeling data.

The flow of the proposed system:

1. Online data is collected from multiple sources.
2. Each entity; restaurants and touristic events are scored based on sentiment scoring.
3. Each entity's keywords are generated by the machine leaning approach. Later all of these keywords are gathered to form a text cloud.
4. Users will be asked to pick n desired keywords from that text cloud. These keywords will be the core inputs of the system, additional inputs such as desired money to be spent or the hourly time range that the user would like to be spending can also be added.
5. By taking primarily the keywords input and additionally other extra inputs, the system will generate an optimal automated day plan by using the sentiment scores that are stored and constraints that are defined by the user.
6. The system will also output the overall satisfaction score and the average estimated cost of that plan.
7. Users may discard an entity on the recommended day plan with an option of with or without replacement. They can even discard the whole optimally recommended day plan to go with the next optimal one.

IX. CONCLUSION

Among used methods Naïve Bayes gave satisfied results with a balanced training data set, whereas TF-IDF approach failed to perform well at predicting negative reviews. Later three different approaches are tested on a sample restaurant data for keyword extraction. The extracted keywords will be the descriptive tags of each restaurant and touristic event, hence it plays an important role in recommender system development. It should be noted that each and every step of this study can be applied to any touristic destination as long as there is available data online. Additionally, the proposed recommender system will be developed to combine a variety of review sources.

X. REFERENCES

[1] TripAdvisor, retrieved from https://www.tripadvisor.com.tr/Tourism-g188590- Amsterdam_North_Holland_Province-Vacations.html (Last access: 22.05.2016)

[2] Pang, B., Lee, L., Vaithyanathan, S. Thumbs up?: Sentiment Classification Using Machine Learning Techniques, Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, 79-86, July 2002.

[3] Metsis, V., Androutsopoulos, I., & Paliouras, G. Spam filtering with naive Bayes – Which naive Bayes? Third Conference on Email and Anti-Spam (CEAS), 2006.

[4] Schneider, K.M. On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification, in Proceedings of the 4th International Conference on Advances in Natural Language Processing, Alicante, Spain, October 2004, 474-485.

[5] Rennie, J.D.M., Shih, L., Teevan, J., Karger, D.R. Tackling the Poor Assumptions of Naive Bayes Text Classifiers, Proceedings of the 20th ThInternational Conference on Machine Learning (ICML-2003), Washington DC, 2003.

[6] Stanford Natural Language Processing on Coursera: https://www.coursera.org/course/nlp (Last access: 22.05.2016)

[7] Raschka, Sebastian. (2015) Pyhton Machine Learning. Birmingham, UK: Packt Publishing

[8] Paltoglou , G., Thelwall, M. A study of information retrieval weighting schemes for sentiment analysis, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 1386-1395, July 2010, Uppsala, Sweden

[9] Ghag, K. and Shah, K. (2014). SentiTFIDF – Sentiment Classification using Relative Term Frequency Inverse Document Frequency. (IJACSA) International Journal of Advanced Computer Science and Applications

[10] Performance Measures for Machine Learning, retrieved from http://www.cs.cornell.edu/courses/cs578/2003fa/performance_measures.pdf (Last access: 22.05.2016)

[11] Kimono + MonkeyLearn: sentiment analysis with machine learning and web scraped data, retrieved from https://blog.monkeylearn.com/kimono-monkeylearn-sentiment-analysis-with-machine-learning-and-web-scraped-data/ (Last access: 22.05.2016)

[12] Witten, H., Ian. Text Mining. Computer Science, University of Waikato, Hamilton, New Zealand.

[13] Bansod, R., Mangrulkar, R. & Bhujade,, G. Text and Image based Spam Email Classification using an ANN Model- an Approach. International Journal on Recent and Innovation Trends in Computing and Communication.

[14] Part-of-Speech Tagging [PowerPoint Slides]. Retrieved from https://www.cs.umd.edu/~nau/cmsc421/part-of-speech-tagging.pdf (Last access: 22.05.2016)

[15] Part-of-Speech Tagging [PowerPoint Slides]. Retrieved from http://www.computational-logic.org/iccl/master/lectures/summer06/nlp/part-of-speech-tagging.pdf (Last access: 22.05.2016)

[16] Brants, Thorsten. TnT A Statistical Part-of-Speech Tagger. In Proceedings of the Sixth Applied Natural Language Processing Conference ANLP -2000, April 29 – May 3, 2000, Seattle, WA.

[17] Ritter, A and et al. Named Entity Recognition in Tweets: An Experimental Study. Computer Science and Engineering University of Washington Seattle, WA 98125, USA.

[18] Introduction to Sentiment Analysis [PowerPoint Slides]. Retrieved from http://lct-master.org/files/MullenSentimentCourseSlides.pdf (Last access: 22.05.2016)

[19] Clark., J., H. and Gonzales-Brenes, J., P. Coreference Resolution: Current Trends and Future Directions. November 24, 2008.

[20] Searle, J., R. (2010) Making The Social World: The Structure of Human Civilization. New York, NY: Oxford University Press.

[21] Word Sense Disambiguation. Retrieved from http://www.scholarpedia.org/article/Word_sense_disambiguation (Last access: 22.05.2016)

[22] Tripathi, S. and Sarkhel, J., K. Approaches to Machine Translation. Annals of Library and Information Studies Vol. 57, December 2010, pp388-393.

[23] Information Extraction. Retrieved from https://en.wikipedia.org/wiki/Information_extraction (Last access: 22.05.2016)

[24] Ohsawa, Y., Benson, N., E. & Yachida, M. KeyGraph: Automatic Indexing by Co-occurrence Graph based on Building Construction Metaphor. Graduate School of Engineering Science Osaka University, Toyonaka, Osaka 560-8531, Japan.

[25] Chen, C. Using Random Forest to Learn Imbalanced Data. Department of Statistics, UC Berkeley.