CSE 6242 Final Report

Toury

Team 06 - Jiaxing Yan, Xinyue Pan, Feng Xiong, Zeling Zhang

Introduction - Motivation

Tourism can be seen from two aspects, micro and macro.For macro aspect, people usually don’t want to read through all the statistics and for micro aspect seldom do people spend time reading a lot of comments from the commercial traveling website. Facing the huge amount of information online we care only about what’s the most important part of it. Our project generalize the macro and micro data about traveling, filter, analyze, re-organize and show you the most significant part of it.

Problem Definition

Advertisements are always unidirectional. People get all the information from the seller and messages provided in such way can be biased.

First of all, for the macro part, in this project we focus on a more general description of a country from different aspects. And conclude some important indicators which shows that for people stress more on different aspects of a country, they can see from the radar chart which country they should go for a vacation. We collected our data mostly from knoema, a detailed data and statistics website which provide the dataset we need to utilize.

For the micro part, in this project we focus on the comments from the customers of airbnb about the house they lived. The keywords are extracted from the comments of the airbnb housing of a city.We also build indicators from the macro data of the country to provide more objective reference of the city.

Survey

Indicators

In order to evaluate the tourism in macro scale, it is very important to use indicators to do this work. In our project, indicator, just as what the name directly represent, is some kind of index number which used to represent what level, or to what extent this country is strong at this part. Since we initially don’t know what values the indicators have, so the classification methods we choose to use is an unsupervised method, k-means classification method[1] , which is basically the most widely used unsupervised method, and to make the result more accurate, we further use lasson[6] to obtain the weight of different contributors of the indicators and make the indicator results more solid, then use naive bayes classification[2] to verify its correctness.

We will mention in the following paragraph that we also use principal component analysis[3] and correlation analysis[4] to generate visual plots. And for the design of the experiment, we use the crime rate[5] as example to achieve the indicator generation to see if this method be used more universal and generate indicators from other aspects.

Keyword Extraction

For the keyword extraction, we started from the simplest and well-known method in common use called Tf-idf[7] which based on the frequency of a term. However, this method works only for calculating the importance of single words in a document. In order to obtain meaningful phrases extraction, we considered the advanced methods based on tf-idf which works for n-grams. The first method is called N-gram IDF[8] which decompose a phrase and calculate the frequency of each part of the phrase based on a measure called information distance. The other one is called RAKE[9] which is also based on the word frequency counting but runs much faster.

Proposed Method

3.1 Tf-idf

In order to retrieve the keywords from the comments we apply the tf-idf method. This methods consists of two parts, term frequency(tf) and inverse document frequency(idf). Definition: Given a document collection D, a word w, and an individual document d ∈ D, calculate

$$w_d = f_{w,\,d} * \log\left(|D|/f_{w,\,D}\right)$$

Here $f_{w,d}$ equals the number of times w appears in d, $|D|$ is the size of the corpus and $f_{w,D}$ equals the number of documents in which w appears in D. A logarithm operation is then applied to the result to normalize the values.

The idea is that the more a word appears in an individual document the more important it is. However, when this word also appears frequently in many other documents it’s more likely to be a stop word like ‘you’,’I’,’a’ and ‘the’ which are much less important.

We have two approaches to calculate the tf-idf. First is to use a training set and model it into a vector space indicating the frequency:

$$\vec{v_{d_n}} = (\mathrm{tf}(t_1, d_n), \mathrm{tf}(t_2, d_n), \mathrm{tf}(t_3, d_n), \ldots, \mathrm{tf}(t_n, d_n))$$

Where t is the word or term and tf(t,d) returns the frequency of term t in document d. The vectors are then transformed into feature matrix:

$$M_{train} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

With this matrix we can calculate the idf vector:

$$\vec{idf_{train}} = (0.69314718, -0.40546511, -0.40546511, 0.0)$$

This approach is very fast but problem is that we fit the test set into the training matrix and thus ignore the unique terms in test set. Therefore we go for the second approach where we treat each document equally and count each words.

3.2 Ngram-idf

The tf-idf method introduced in 3.1 only works for finding the 1-gram words where the sequence of text consists of only 1 word. Actually tf-idf itself does not work well for N-grams since the definition of IDF totally contradicts the definition of important phrases in a document. Along with the logic of IDF, n-grams of unnatural collocation are more likely to be keywords. Since we want to get more accurate description of a city, we need a more reasonable way to retrieve the n-gram keywords out of the comments. The method we implemented is proposed in [10] where the new n-gram IDF is defined as:

$$\begin{aligned} IDF_{N\text{-}gram}(g) &= IDF(\theta(g)) - MED(g) \\ &= \log \frac{|D| \cdot df(g)}{df(\theta(g))^2} \end{aligned}$$

$\theta(g)$ indicates the all the possible 1-grams to n-grams that can form phrase g. For example,N=2 and g = [‘big red ball’] then $\theta(g)$ will be [‘big’,’red’,’ball’,’big red’,’red ball’]. The idea is that we consider not only the frequency of the n-gram g among all documents but also its components’ frequencies. If the components are too frequent then it reduces the importance.

3.3 Feature Selection

Due to the huge amount of variables in our dataset, feature selection is required for our project. There are three main categories of feature selection algorithm: wrappers, filters and embedded methods.

In our project, we use a traditional method in Statistics: stepwise regression, which is a wrapper method. The main idea is to add the best feature in each round and the process is controlled by the cross-validation.

In addition, Lasso is also one of our choice, which is an embedded method. Lasso regression will build a linear model penalized with L1 norm and the goal is to shrink the coefficient and hence reduce the dimension of data. Lasso will shrink many of the coefficient to zero and the feature selection can be done by selecting the non-zero coefficients.

Further, PCA is also applied in our project. Due to the absence of response variable in our indication data, the unsupervised methods is required. PCA satisfied our meets to generate weights and do feature selection for the multicollinear features.

3.4 Naive Bayes Classifier

We have different kind of indicators, include culture, history, scene, Education, etc. We think every indicator has their own ways to define themselves, and use the **Naïve Bayes Classification**, it is possible that we can narrow our complex dataset into simple version as 1, 2, 3 or so on to be easily seen. The basic principle is as below, and also include the example of the indicators.

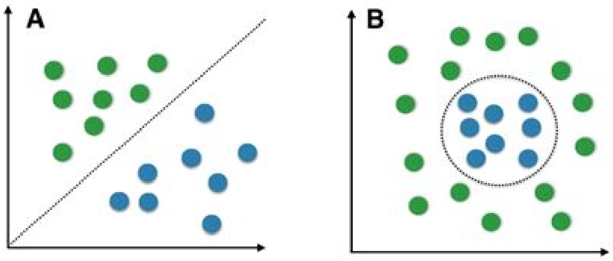
$$\underset{\text{Posterior Probability}}{P(c \mid x)} = \frac{\overset{\text{Likelihood}}{P(x \mid c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Naive Bayes Equation

3.5 K-means classification methods

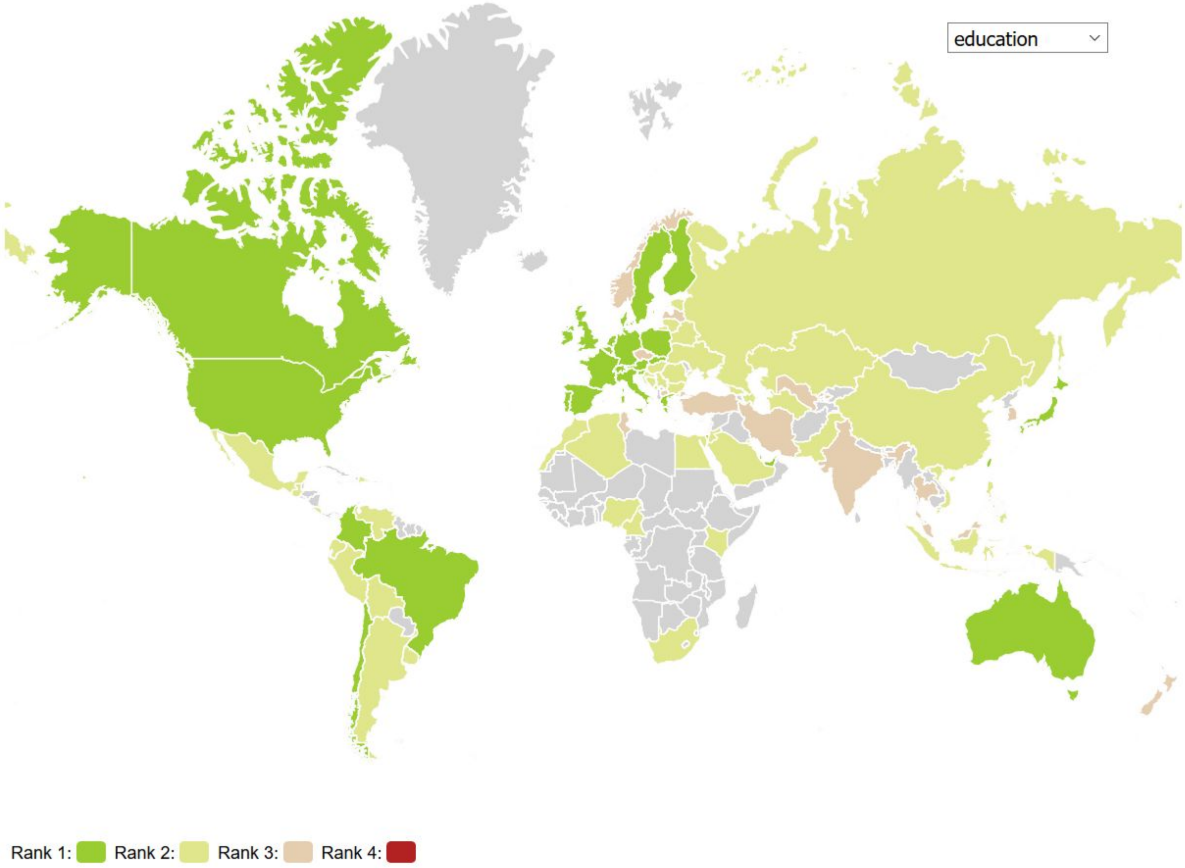
***k*-means clustering** is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *k*-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.



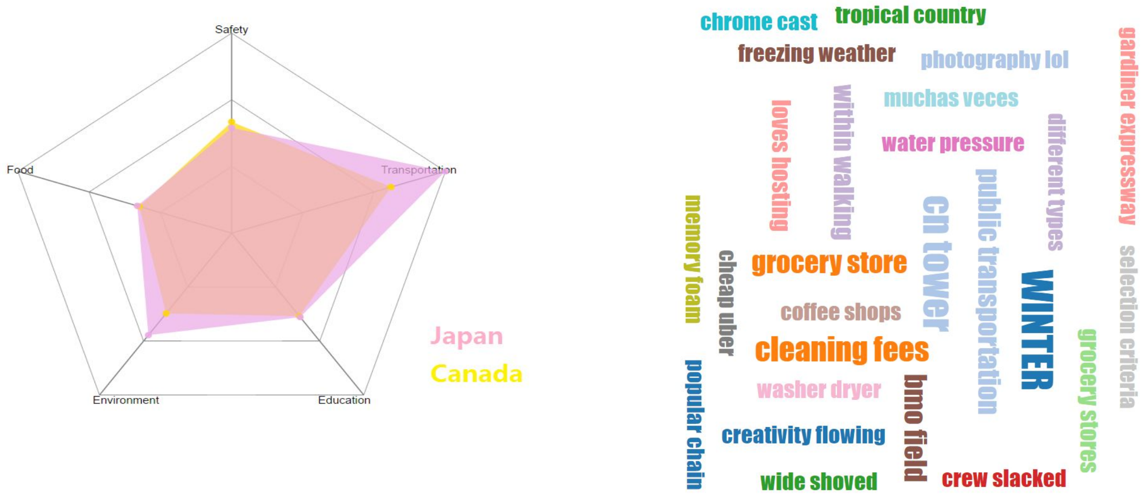
K-means classification methods

3.6 Data Visualization and User Interfaces

For data visualization we used D3. The index page shows the world map with the clustered indicators. The select box on the upper right side allows users to choose the indicator they want to see. Grey area means no data retrieved.



Heatmap for the indicators



Radar chart and key words presentation

The detailed pages show the radar charts of countries in terms of the five indicators we calculated and the keywords extracted from each of the five cities, animated and shown in the order of seasons. The sample pages are zipped and uploaded. They can be opened using browser and run locally. We upload everything on github page , https://github.com/rubbishbean/Toury and deploy interactive data visulizations on https://toury-10335.firebaseapp.com so that the content can be viewed online.

Experiments and Evaluation

Indicators

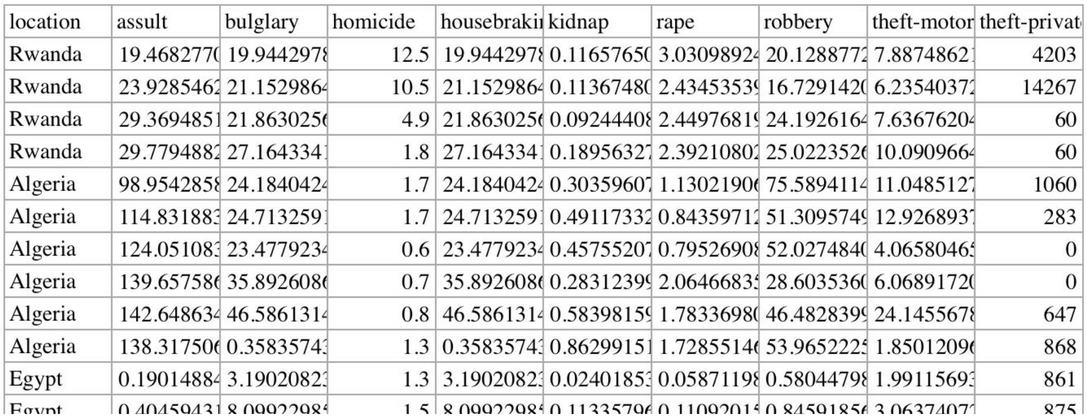
In order to decide how to calculate the values of all indicators, we must first use one indicator as example to do the test and come up with some useful conclusions, and then we can better understand and conduct the following steps to calculate the other indicators.

In our experiment, we select the safe index, which related to the crime situation as an indicator to describe if a country is safe or not. The following steps are how we design the test and obtain some meaningful results.

1. Collect the data from the website knoema about the crime situation in different countries. This rate may related to different kind of crimes. What we collected are the theft private car, theft motor, robbery, rape, kidnap, housebreaking, homicide,

burglary and assault, how many times those crimes are happened every 100,000 people.

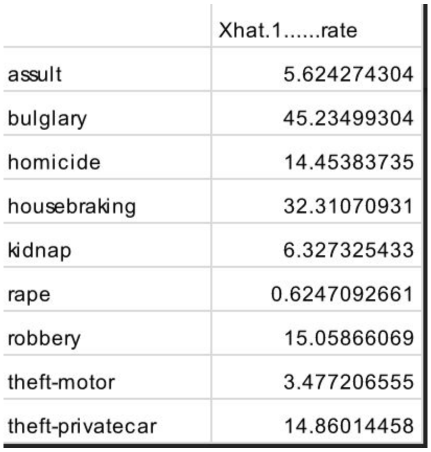
2. Then summarize those data in a table, list the name of the countries and different kind of crimes,Just as what the figure shows below, so we can use those data to generate our value of our crime indicator. (The same name of the location actually represent different years, which not include in this plot.



location	assult	bulglary	homicide	housebrakin	kidnap	rape	robbery	theft-motor	theft-privat
Rwanda	19.4682770	19.9442978	12.5	19.9442978	0.1165765	3.0309892	20.1288772	7.8874862	4203
Rwanda	23.9285462	21.1529864	10.5	21.1529864	0.1136748	2.4345353	16.7291420	6.2354037	14267
Rwanda	29.3694851	21.8630256	4.9	21.8630256	0.0924440	2.4497681	24.1926164	7.6367620	60
Rwanda	29.7794882	27.1643341	1.8	27.1643341	0.1895632	2.3921080	25.0223526	10.090966	60
Algeria	98.9542858	24.1840424	1.7	24.1840424	0.3035960	1.1302190	75.5894114	11.048512	1060
Algeria	114.831883	24.7132591	1.7	24.7132591	0.4911733	0.8435971	51.3095749	12.926893	283
Algeria	124.051083	23.4779234	0.6	23.4779234	0.4575520	0.7952690	52.0274840	4.0658046	0
Algeria	139.657586	35.8926086	0.7	35.8926086	0.2831239	2.0646683	28.6035360	6.0689172	0
Algeria	142.648634	46.5861314	0.8	46.5861314	0.5839815	1.7833698	46.4828399	24.145567	647
Algeria	138.317506	0.3583574	1.3	0.3583574	0.8629915	1.7285514	53.9652225	1.8501209	868
Egypt	0.1901488	3.1902082	1.3	3.1902082	0.0240185	0.0587119	0.5804479	1.9911569	861
Egypt	0.4045943	8.0992298	1.5	8.0992298	0.1133579	0.1109201	0.8459185	3.0637407	875

Crime situation

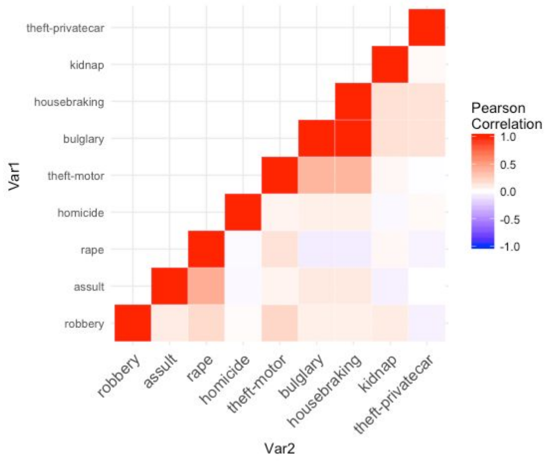
3. For the next step, we need to obtain the weight of different kind of crimes, here is to use the PCA to return the weight value of each crimes. Higher the scale factor, more important this kind of rate can be.

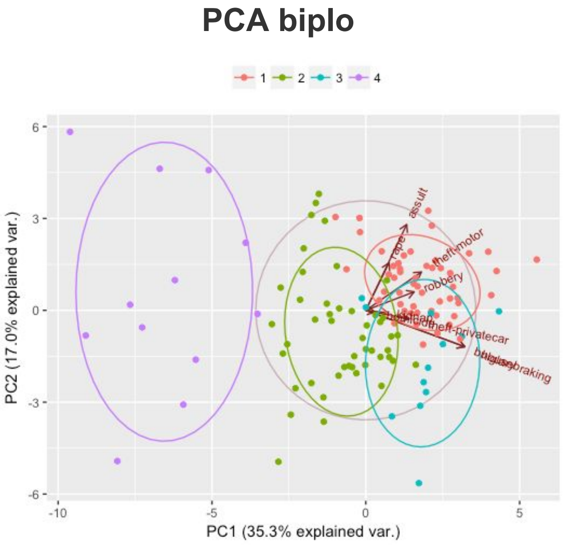


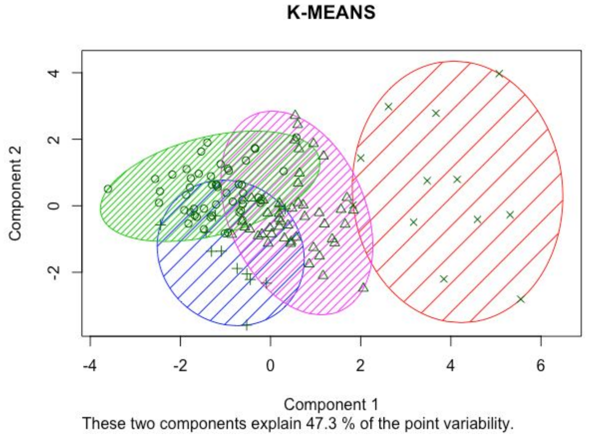
	Xhat.1......rate
assult	5.624274304
bulglary	45.23499304
homicide	14.45383735
housebraking	32.31070931
kidnap	6.327325433
rape	0.6247092661
robbery	15.05866069
theft-motor	3.477206555
theft-privatecar	14.86014458

Weight for different kind of crimes

4. Then it is the time to calculate the value of the crime indicator. We assume we have four level of safety, use numbers 1, 2, 3, 4 to represent them, larger the number is , more dangerous this country tend to, and safer the country can be.
5. To visualize our results, we need to further use different plots to do the analysis. Below are some plots which show the results more accurate and vivid.



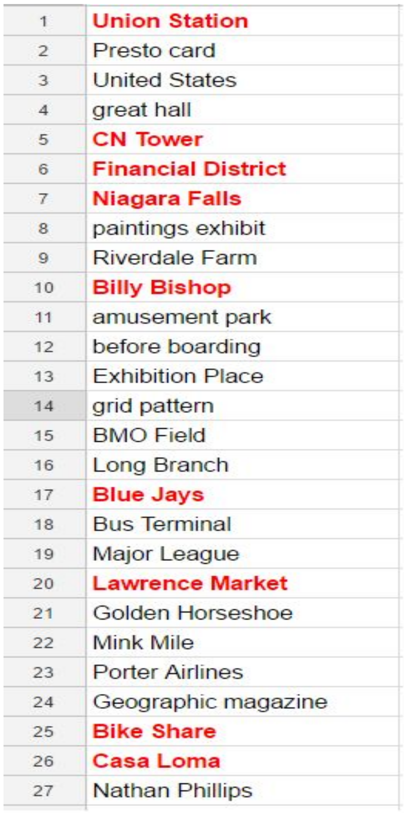




The first plot above is the correlation-heatmap, which represent the correlation between features and there are some multicollinearity inside, which means that PCA is applicaticable for this project. The second plot is the PCA biplot, which shows the contributions of each feature to the first two components. Finally, the last plot shows the result of k-means clustering that we divide the locations into four different groups.

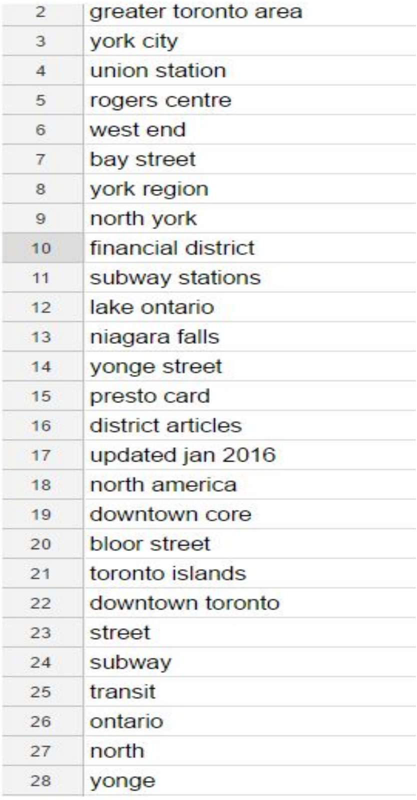
Word Extraction

We followed the experiments steps described in [9] and used the wikivoyage page of Toronto as a sample target. Intuitively we took the names of the tourism attractions and all the phrases shown in bold as important words. We ran both N-gram IDF and RAKE on the text content of Toronto's Wiki page. The results are shown below:



1	**Union Station**
2	Presto card
3	United States
4	great hall
5	**CN Tower**
6	**Financial District**
7	**Niagara Falls**
8	paintings exhibit
9	Riverdale Farm
10	**Billy Bishop**
11	amusement park
12	before boarding
13	Exhibition Place
14	grid pattern
15	BMO Field
16	Long Branch
17	**Blue Jays**
18	Bus Terminal
19	Major League
20	**Lawrence Market**
21	Golden Horseshoe
22	Mink Mile
23	Porter Airlines
24	Geographic magazine
25	**Bike Share**
26	**Casa Loma**
27	Nathan Phillips

Figure 1



2	greater toronto area
3	york city
4	union station
5	rogers centre
6	west end
7	bay street
8	york region
9	north york
10	financial district
11	subway stations
12	lake ontario
13	niagara falls
14	yonge street
15	presto card
16	district articles
17	updated jan 2016
18	north america
19	downtown core
20	bloor street
21	toronto islands
22	downtown toronto
23	street
24	subway
25	transit
26	ontario
27	north
28	yonge

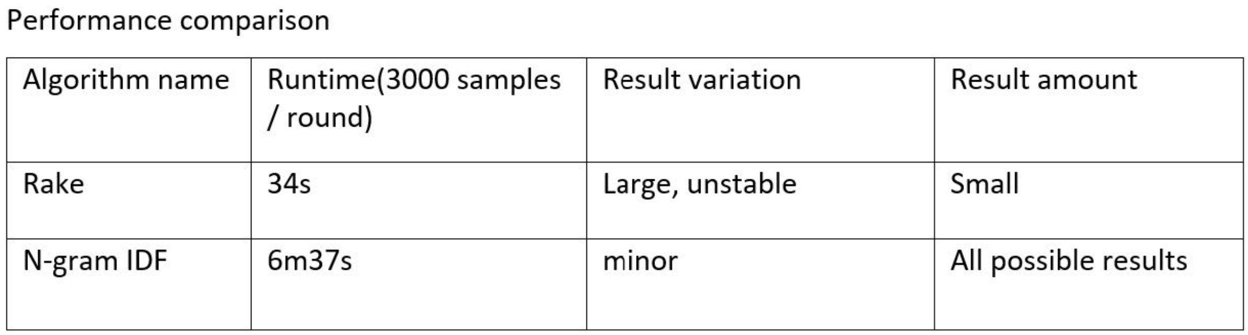
Figure 2

Figure 1 shows the result of N-gram IDF while Figure 2 shows the result of RAKE. We can see that both of them capture some of the keywords from the original text but N-gram IDF outperforms RAKE in terms of the capture amount and quality of the keywords.

We then ran the two methods on the airbnb comments of Toronto with sampling (number of samples = 3000) several times. The phrases extracted from both comments and the Wiki page using Ngram IDF are marked as red in Figure 1. RAKE did not perform very well in extracting the keyword and we failed to find any matches between the comments result top 50 and the Wiki keywords listed in Figure 2.

We concluded two reasons that may cause this result. The first is the nature of RAKE that it only counts the frequency of a phrase in its context which is the document that phrase appears in. Briefly speaking, this is a simplified version of tf-idf. Without the inverse document frequency step in normal tf-idf RAKE captured more trivial phrases. The second reason is that RAKE works fine with a single long file since it only counts the word frequency in the current document, however the comments are read in as separate documents. RAKE failed to take multiple documents into account at the same time and that lead to the poor performance.

We compared the two methods not only on the result quality but also on running time and other measures. The performance table is shown below:



Performance comparison

Algorithm name	Runtime(3000 samples / round)	Result variation	Result amount
Rake	34s	Large, unstable	Small
N-gram IDF	6m37s	minor	All possible results

The performance table shows that the Rake as stated by its name, works much faster but lacks of stability. N-gram IDF is extremely slow since we didn’t optimize the data structure but it gets more reliable results. Running with random sampling several times with N-gram IDF, we found that the results did not vary too much.

Conclusion

Summary of Innovation

1. Retrieve information from comments and improve algorithm from 1-gram to n-gram keywords extraction.
2. Generate indicators from international-wise travelling data with weight.
3. Display both world map for macro view as well as the detailed pages for cities.

Challenge and Improvement

For calculating the indicators, all of our data are unsupervised. One problem is that we can not do cross validation like in the supervised case where we can use the training data and testing data. To validate the clusters in the result, we can only judge by eyes to see if the results are reasonable or not. To somehow validate our results numerically, we applied Naive Bayes to the same dataset and found that the similarity is very high.

However, both K-means and Naive Bayes aim to reduce the variance in data. If the bias is too large both of the algorithms are not able to detect and adjust.

When we first decided to do the keyword extraction we prefered to use the library provided online and we found RAKE. After testing with huge amount of comments we found that RAKE could not satisfy our expectation so we did further research and implemented one of the most reasonable and doable method among all the papers we read. One limitation of our implementation is that we did not optimize N-gram IDF using the suffix tree structure as described in [9] which leads to an extremely slow running time. In this case we can only do sampling when processing the comments and this greatly affected the result of keyword extraction.

Distribution of Team Member Effort

Jiaxing Yan	Data collection,N-gram IDF implementation, Data visualization
Xinyue Pan	Data Cleaning, Indicator feature selection and classification
Feng Xiong	Indicator and experiment design, data collection, preprocessing.
Zeling Zhang	RAKE model code and test; Poster Design, Full-Stack Engineer for Website

Reference

[1]Hartigan, J. A., and M. A. Wong. “Algorithm AS 136: A K-Means Clustering Algorithm.” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, 1979, pp. 100–108., www.jstor.org/stable/2346830.

[2]Murphy K P. Naive bayes classifiers[J]. University of British Columbia, 2006.

[3]Jolliffe I. Principal component analysis[M]. John Wiley & Sons, Ltd, 2002.

[4]Cohen J, Cohen P, West S G, et al. Applied multiple regression/correlation analysis for the behavioral sciences[M]. Routledge, 2013.

[5]Huang W S W, Wellford C F. Assessing indicators of crime among international crime data series[J]. Criminal Justice Policy Review, 1989, 3(1): 028-047.

[6]Hans C. Bayesian lasso regression[J]. Biometrika, 2009: 835-845.

[7] Ramos, Juan. "Using tf-idf to determine word relevance in document queries." In Proceedings of the first instructional conference on machine learning. 2003.

[8] Masumi Shirakawa, Takahiro Hara, and Shojiro Nishio. N-gram IDF: A Global Term Weighting Scheme Based on Information Distance. In WWW ’15: Proceedings of International World Wide Web Conference, 2015

[9]Berry, Michael W., and Jacob Kogan, eds. *Text mining: applications and theory*. John Wiley & Sons, 2010.