Benjamin Song, Xiaohan Jia, Jingya Yu, Xinyu He, Jeremy Dou

Final Project Report - Final, EECS 486

Professor Mihalcea, Mr. Lee, and Mr. Wang

Apr 12, 2022

Final Project Report - Final

Abstract

The topic of this project is determining the best season to travel to a certain location. To make the determination, textual information about locations would be crawled from the internet, and predictions would be made based on the information, and output the best season to travel for several popular cities.

Project Description

During the project idea brainstorming sessions, we wanted our project to be as relevant to our daily lives as possible. For the past two years, the pandemic of COVID-19 has undoubtedly put a stop to world's travel, which has caused a great impact on the world's economy and travel industries such as airlines, hotels, cruises, etc.

As the vaccination rates steadily increase, more and more countries around the world have decided to fully reopen and to be ready to welcome the travelers and tourists from around the world for the first time since March 2020, and some of them, such as Singapore, South Korea, Australia, New Zealand, etc, were even the ones that used to impose the most strict travel bans and quarantine regulations back then.

As a matter of fact, one of our team members Ben proposed this idea, and he already took initiatives by starting traveling to Europe during the spring break. While we can't be more excited about all the new potential travel destinations that we can explore, Ben has convinced us, and we think it would be a great idea that we implement a project that can provide us with more ideal travel recommendations with automatic web crawling and language processing that we've been learning and practicing in this class.

The topic of this project is determining the best season to travel to a certain location. To make the determination, textual information about locations would be crawled from the internet, and predictions would be made based on the information. The system can output the best season to travel given a location as input.

- Data: Textual data about locations would be scraped from informational websites such as Wikipedia or travel review websites such as Tripadvisor.
- Method: Textual data scraped from the internet would be preprocessed for the analysis. A set of parameters would be constructed for each season, and the determination of best season would be computed through correlation between the data and season parameters.
- Evaluation: Widely-accepted opinions about the best season to travel to a location would be used as the correct answer. The location-season combinations would be collected by searches such as "best summer

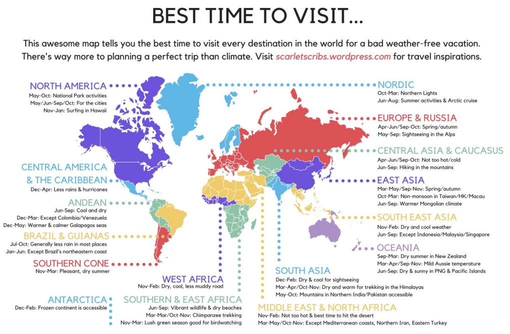
destinations" on the internet. The correct answers would be compared against model output of this project.

Related Work (References)

To help us evaluate our project's scope and plan our approach more holistically, our team has dived into researching existing works that are related to our study subject. Our research is based on two main aspects, including the current decision process of the best season to visit a destination used by travelers and travel recommendation publications, and the existing usage of textual data correlation in projects related to subject classification.

To understand the decision process, our team has chosen a summary article about the best season to visit all regions of the world with a map illustration [1]. Rather than using the generalized classification of seasons as a guideline for our own classification model, we used this map as a way to understand the key factors that contribute to seasons' suitability for destinations. An example would be having the term "Northern Lights" as a parameter for winter and summer destinations, as it is generally the best to visit Nordic destinations in the summer months or the winter months. A set of articles published by US News that discuss the best seasons to visit the world's most visited destinations is used to construct the correct answer key discussed above [2].

To review current availability of tools that could be used by the potential users of our model, our team has browsed through the internet for similar models. It has been found that most related models only take into account the average temperatures as the sole variable, mostly in an input-output representation [3]. No model has been found to utilize the information retrieval approach that we are taking.


BEST TIME TO VISIT...
This awesome map tells you the best time to visit every destination in the world for a bad weather-free vacation.
There's way more to planning a perfect trip than climate. Visit scarletscribs.wordpress.com for travel inspirations.
NORTH AMERICA
May-Oct: National Park activities
May/Jun-Sep/Oct: For the cities
Nov-Jan: Surfing in Hawaii
NORDIC
Oct-Mar: Northern Lights
Jun-Aug: Summer activities & Arctic cruise
EUROPE & RUSSIA
Apr-Jun/Sep-Oct: Spring/autumn
May-Sep: Sightseeing in the Alps
CENTRAL ASIA & CAUCASUS
Apr-Jun/Sep-Oct: Not too hot/cold
Jun-Sep: Hiking in the mountains
CENTRAL AMERICA & THE CARIBBEAN
Dec-Apr: Less rains & hurricanes
EAST ASIA
Mar-May/Sep-Nov: Spring/autumn
Oct-Mar: Non-monsoon in Taiwan/HK/Macau
Jun-Sep: Warmer Mongolian climate
ANDEAN
Jun-Sep: Cool and dry
Dec-Mar: Except Colombia/Venezuela
Dec-May: Warmer & calmer Galapagos seas
SOUTH EAST ASIA
Nov-Feb: Dry and cool weather
Jun-Sep: Except Indonesia/Malaysia/Singapore
BRAZIL & GUIANAS
Jul-Oct: Generally less rain in most places
Jan-Jun: Except Brazil's northeastern coast
OCEANIA
Sep-Mar: Dry summer in New Zealand
Mar-Apr/Sep-Nov: Mild Aussie temperature
Jun-Sep: Dry & sunny in PNG & Pacific Islands
SOUTHERN CONE
Nov-Mar: Pleasant, dry summer
SOUTH ASIA
Dec-Feb: Dry & cool for sightseeing
Mar-Apr/Oct-Nov: Dry and warm for trekking in the Himalayas
May-Oct: Mountains in Northern India/Pakistan accessible
WEST AFRICA
Nov-Feb: Dry, cool, less muddy road
ANTARCTICA
Dec-Feb: Frozen continent is accessible
SOUTHERN & EAST AFRICA
Jun-Sep: Vibrant wildlife & dry beaches
Mar-Mar/Oct-Nov: Chimpanzee trekking
Nov-Mar: Lush green season good for birdwatching
MIDDLE EAST & NORTH AFRICA
Nov-Feb: Not too hot & best time to hit the desert
Mar-May/Oct-Nov: Except Mediterranean coasts, Northern Iran, Eastern Turkey

[1]

[1] https://scarletscribs.wordpress.com/2018/01/07/this-awesome-map-tells-you-the-best-time-to-visit-every-destination-in-the-world/

[2] https://travel.usnews.com/Destinations/

[3] https://www.thebesttimetovisit.com/weather/afficherville.php

For the classification algorithm, basic searches about the ways to classify objects would frequently lead to the answer of text vectorization, which is the underlying implementation of our group's proposed approach [4]. A description article of text classification in Python provides an example of classifying the ethical characteristics of tweets [5]. The data preparation and feature engineering sections of the article have been reviewed by our group's members to inspire the implementation plan of our model. Another article about document classification suggests a similar approach that our team is planning on taking [6]. In the subsection of *defining keywords*, the practice of collecting a list of keywords, similar to the queries in our project proposal, for each subject category is introduced. An example would be "protein" and "calories" being the query for the category of nutrition.

[4] https://datascience.stackexchange.com/questions/98308/how-to-classify-objects-from-a-description-in-natural-language

[5] https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/
[6] https://www.altexsoft.com/blog/document-classification/

Data Collection, Annotation, and Data Samples

The datasets in this project include two parts: the reviews of traveling destinations and the passages associated with different seasons. For the reviews, we first chose a list of traveling destinations from the passage World’s Best Places to Visit ranked by travel.usnews.com, which include the best two seasons to visit for most of the cities and can be used for the evaluation part. Then we search for the reviews of these places respectively in Tripadvisor, a website for travelers to prepare for their trip. We used a web-crawler to access and download the travel guide in Tripadvisor, and finally get the original data of visitor’s review for 21 cities, with passages of the same places in the same folder. The web-crawler is written in python, and the packages we imported include requests, bs4, and gne. By replacing the link in the response function where the requests access through the corresponding websites, we integrated 21 pieces of python code and organized them in the coding folder named “Tripadvisor Code”.

For the second part, we aimed to retrieve related keywords of different seasons from the passages associated with them. But since these passages shown in Chrome do not have a common format of their website, this problem prevented us from continuing to use web-crawlers to access and download the related passages. Thus, we chose to copy and paste the related articles and sorted them into corresponding folders according to the seasons. The keywords we used in the searching process include “Is <season> a good time to travel?”, “Why travel in <season>?”, and “<season> characteristics” where <season> can be replaced by spring, summer, fall, and winter. In Particular, we chose “fall” rather than “autumn” since the former one is more common in American cities and also unified with the information we would use in evaluation.

Method Description (Methodology)

Our first step is to match the season-representative words with each season. Our assumption is that each season has representative features, and there are words that can represent those seasons. Those keywords are essential for our model to identify the best season for a specific destination to travel, and they also serve as criteria to evaluate how much a city that is related to a season. For example, when thinking about summer, we always think about ice-cream. Even though people might eat and talk about ice-cream in other seasons, the word “ice-cream” has more chances to appear in summer-related articles.

With this assumption in mind, we collect articles that focus on describing the seasons and start to extract the keywords. Our intuition is to use the Vector Space Model for our IR system. We utilize the season-related articles and create 3 vectors storing term frequency, inverse term frequency, and tf-idf model. We firstly read in all the articles that are for each season and regard each word as a string, then we preprocess the content to get rid of the punctuations and stop words to reduce the neutral words(words that are equally relevant to all of the seasons). After storing each word into each season’s vector, we count the term frequency for each word within each season’s vector and also store a copy of vocabulary. For the inverse term frequency, we use the formula . Finally, to find each season’s most relevant words, we multiply the target word ’s tf and idf within each season, and rank the result. The input number N will provide us with the top N words that are most relevant to each of the seasons, in terms of vectors. Those four vectors will be used as queries for further similarity score computation.

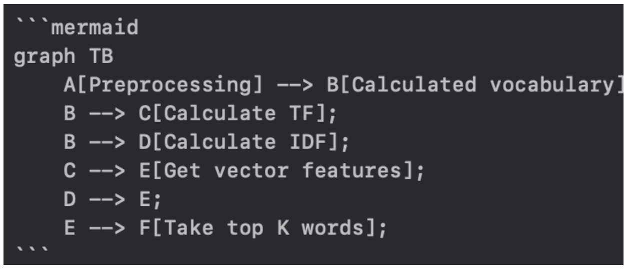
How to determine the most suitable season for a city to visit? In order to solve this problem, we classify the problem as: extract the features of city and season respectively, and then calculate the similarity between them. So, extracting features is a critical part of our model.

We used a crawler to crawl some questions from Google about ‘what season to go to a certain city’, and then divided them into four categories: spring, summer, fall, and winter. There are 5 articles in each category.

Then, in order to connect the learned knowledge, we decided to adopt the TF-IDF method as the method for feature extraction. Specifically, TF and IDF are counted separately, and then a word vector of length N is obtained for each season (N is the length of the vocabulary). Then, we used two methods to represent the seasonal information:

1. Use the word vector itself as a feature representation
2. Extract the first K words of the word vector as feature representations

At the specific implementation level, I reused some of the word preprocessing work in assignment1, and used the calculation method of TF-IDF in assignment2 to implement a new seasonal feature extractor. The basic flow of my implementation of the method is as follows:



~~~
```mermaid
graph TB
    A[Preprocessing] --> B[Calculated vocabulary]
    B --> C[Calculate TF];
    B --> D[Calculate IDF];
    C --> E[Get vector features];
    D --> E;
    E --> F[Take top K words];
```
~~~

After the data of keywords of four seasons and documents of the destination reviews has been collected and processed, a vector space model is used to match the travel destination and its best season of traveling. The vector space model is originally used for, given a large document collection and query string, finding the document relevance of the query. In our case, the query (season keywords) is defined as user’s expectation of traveling in each season and the documents is defined as others description of a given destination, it is also reasonable to use the vector space model to calculate the similarity between the query of season expectation and the document of destination review, and the result (best season) is the query with the highest similarity score.

The inverted index matrix is created to represent travel destination documents. The rows of the matrix represent documents and columns represent terms (excluding very high frequency stop words and stemmed). Matrix cells are weighting schemes, which help the model to exploit sparsity, and allow efficient storage and querying. A three-letter string was used in order to represent each term-weighting scheme with the particular local, global and normalization factor combinations. The following six local, global, and document length normalization weighting schemes were tested: txx, txc, tfx, tfc, tpx, tpc. The test performance suggests tfc performs better for documents. Also, other four matrices are used to represent season-keywords queries. By treating the query as a short pseudo-document, create other four matrices with tfx weighting scheme to represent the user expectations vector of each season.

The cosine similarity measure to compute the distance between two vectors was used, as it is the most commonly used measure of similarity in the literature and has been shown to produce good results. The vector space model is often used to rank documents by decreasing similarity with cosine and return to user the top k ranked documents. The measure methods are adjusted here because the season is treated as a query and the destination is treated as a document. Given a destination, calculate the similarity score of a certain season by summing up all the cosine similarity measures of each review document vector and the season query vector. Then, Compare the similarity score of this destination and each four seasons, and the best season is the query with the highest similarity score.

<u>Method Evaluation</u>

One of the evaluations of the project will be measured by comparing the season result we get from the model's output to our labeled data collection. The labeled data collection represents the actual best season, which is determined by the set of US News articles introduced in the Related Work section. At this stage we use labeled data containing the authoritative answers to test the model. These data are considered the gold standard when evaluating the accuracy of our model. With the data mentioned above, two metrics are computed by the team to evaluate the model's performance, which are model accuracy and adjusted model accuracy.

$$Accuracy = \sum_{i=0}^{n} \frac{x_i}{n}$$

x_i - 1 if correctly predicted, 0 otherwise

n - number of test locations

$$Accuracy = \sum_{i=0}^{n} \frac{a_i}{n}$$

a_i - adjusted correctness (see table)

n - number of test locations

It has been observed that for most travel destinations, traveling in spring and fall would have a similar experience, and most locations that are best to travel in spring are also best to travel in fall. It can also be inferred that traveling one season off (in summer for a spring destination) would be better than traveling in summer for a winter destination and vice versa. As a result, the adjusted correctness scale is constructed by the team based on these two facts, and the corresponding score is illustrated in the table below. This is an original evaluation approach that is proposed intuitively by the team.

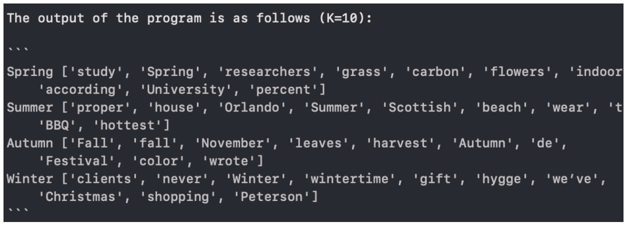
	Predicted Spring	Predicted Summer	Predicted Fall	Predicted Winter
Actual Spring	1	0.5	1	0.5
Actual Summer	0.5	1	0.5	0
Actual Fall	1	0.5	1	0.5
Actual Winter	0.5	0	0.5	1

The model predicts the best season to visit the 21 test locations with 39% accuracy, compared to random prediction with accuracy of 25%. Using the adjusted metrics based on the practical utilization in tourism, the model reaches 70% accuracy. Although the model has already obtained observed functionality, the team has proposed a few possible approaches for further improvement in future, including using a wide range of textual data for queries and exploring more extensive parameters, and expanding model input to include travelers' personal preferences.

Given the structure of the model, our team does not expect to have any significant concern for efficiency and scalability. As the approach computes the queries for seasons before a new location's best reason is computed, the model is eager in processing output therefore easily scalable without excessive time required.

Conclusion

For keywords identifying for each season, the output is the following (K = 10):



~~~
The output of the program is as follows (K=10):

```
Spring ['study', 'Spring', 'researchers', 'grass', 'carbon', 'flowers', 'indoor
    'according', 'University', 'percent']
Summer ['proper', 'house', 'Orlando', 'Summer', 'Scottish', 'beach', 'wear', 't
    'BBQ', 'hottest']
Autumn ['Fall', 'fall', 'November', 'leaves', 'harvest', 'Autumn', 'de',
    'Festival', 'color', 'wrote']
Winter ['clients', 'never', 'Winter', 'wintertime', 'gift', 'hygge', 'we’ve',
    'Christmas', 'shopping', 'Peterson']
```
~~~

According to the output, it’s safe to say that the experiment of extracting seasonal features has achieved quite successful results.

Utilizing the vector space model of information retrieval, a model is constructed successfully to predict the ideal seasons for traveling to given destinations. The model is proposed and implemented during a time when international travel is growing after an extensive period of pause.

The vector space model incorporates textual information about tourism and human reviews of travel destinations. It produces the predicted ideal season by comparing the features of seasons with the experiences described by tourists and identifying the closest match.

Functionality of the model has been observed, while improvement opportunities are still abundant on both general algorithmic logic and parameters selections.

Overall, the construction of this model has successfully provided intuition about a season recommendation platform for tourism that considers factors beyond geography and climate.

Individual Contributions

Benjamin Song has been mainly responsible for proposing the model and its implementation, researching related work, and evaluating the model.

Xiaohan Jia has been mainly responsible for proposing the data collection and model ideas and building the vector space model.

Jingya Yu has been mainly responsible for proposing the possible model evaluation strategies, research related work, and co-implement the training model.

Jeremy Dou has been mainly responsible for the implementation of the web crawling, text processing, keywords identifying for each season, and along with other overall project management tasks.

Xinyu He has been mainly responsible for the data collection, selection of city lists, implementation of the web crawling, and original data of the season-related passages.

References

Darwitan, Posted by Andrew. “This Awesome Map Summarizes the Best Time to Visit Every Country in the World.” *Scarlet Scribbles*, 9 Nov. 2021, https://scarletscribs.wordpress.com/2018/01/07/this-awesome-map-tells-you-the-best-time-to-visit-every-destination-in-the-world/.

Bragpacker. “A Destination for Every Season: Guide to Picking the Best Travel Destinations All Year Long.” *Bragpacker*, https://bragpacker.com/a-destination-for-every-season-a-quick-guide-to-picking-the-best-travel-destinations-all-year-long/.

“Best Times to Travel to the World's Most-Visited Cities.” *Far & Wide*, https://www.farandwide.com/s/best-time-to-travel-around-the-world-0df967e053fa4328.

BlackBeansBlackBeans 12333 bronze badges, and Nicolas MartinNicolas Martin 94011 silver badge1010 bronze badges. “How to Classify Objects from a Description in Natural Language.” *Data Science Stack Exchange*, 1 June 1969, https://datascience.stackexchange.com/questions/98308/how-to-classify-objects-from-a-description-in-natural-language.

“A Comprehensive Guide to Understand and Implement Text Classification in Python.” *Analytics Vidhya*, 26 July 2019, https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/.

Editor. “Document Classification with Machine Learning: Computer Vision, OCR, NLP, and Other Techniques.” *AltexSoft*, AltexSoft, 17 Nov. 2021, https://www.altexsoft.com/blog/document-classification/.