CSC 791: Project Report R3

Itinerary Summarization from Travelers' Blog

Submitted by:

Tasmia Shahriar **(tshahri)**

Anurata Hridi **(aphridi)**

November 29, 2021

Introduction

The rapid progression of information technology and the advancement of the internet have revolutionized tourism [3]. Tourism websites serve as a valuable tool for both the end-users and the providers for marketing, cost and time-effective travel planning, and online communication. Travelers nowadays seek to utilize their travel time at its best and look back to these websites for planning their tours. Travelers' blogs contain adequate information on the locations they have covered during their tour and, in some cases, the order of these travel points. The experience shared by the travelers in their blogs can be processed to help new travelers plan their tours and maximize their travel experience in a short time. We want to extract information about the places they mentioned in their blogs, and in which order they covered all those destinations, which we call *itinerary summarization*. Instead of scouting around the internet for all sorts of information available before we can make a plan for a trip, we propose to deliver an accurate summary of the itineraries. This will save time and efforts of the people in need.

People who want to go through reviews and experiences narrated by other travelers before they plan for vacations will be the potential users of this application, among others. Instead of reading lengthy blogs about travel information, schedule, and whatnot, the readers will be able to get an idea of the possible itinerary they can follow based on other people's experiences and feedback. They can remain less distracted and more focused on the information they want. Since we propose analyzing travel blogs and coming up with probable itineraries based on them, travelers will instantly know how they need to organize their tour. Such a system will save their time and effort as they will not have to study multiple blogs; rather, they can compare the summarized outputs from multiple similarly written travel blogs and put together their own plan.

We propose to work with time-sensitive orders of the location alongside the proper extraction of the locations, which means our summary will maintain the exact order of any location by which they were traversed regardless of how arbitrarily the traveler mentioned those locations in their blogs. To that aim, we mine travel blogs focusing on different localities and periods and use those as ground truth. Based on our annotations, we will use them for evaluating our model using BLEU score. We use written blogs only, preferably published in Medium and similar blogging sites. Compared to the baseline model that suggests itinerary based on verb and its related noun phrases, our proposed model suggests itinerary based on verb and its refined named entity. The model outperformed the baseline model with 60% BLEU score.

Involved NLP task

Let’s assume that we want to visit Yosemite National Park and while surfing for relevant information, we stumble upon this blog excerpt of a traveler in this regard,

“Your 3 day Yosemite itinerary begins with a long drive into the park. Starting from either San Francisco, San Jose or Oakland, you’re looking at a 3.5 – 4.5 hour drive, depending on your exact starting point and traffic conditions.
It’s in your best interest to arrive to the park as early as possible, but as long as you arrive by late afternoon you’ll still be able to follow this itinerary. Stop by Tunnel View on your way into the park; it will be directly on your route if you’re taking the south entrance into the park, or is just a short detour away if you’re taking the north entrance.
After snapping a few photos and soaking in the sweeping scenery at Tunnel View, we continued on to Yosemite Valley Lodge and immediately checked into our room. We were eager to get the car parked and unpacked so we could start enjoying our long weekend in Yosemite!
From the lodge, it was a short walk down the road to one of the valley’s easiest and most picturesque “hikes” – Cook’s Meadow. This flat, scenic 1-mile trail yields amazing views of Yosemite Falls and Half Dome right from the valley floor, and is the perfect way to get acquainted with the park. On our way back to the lodge, we also did a quick stroll along the Lower Yosemite Falls trail.
Today is the big day! You’ve got a full day to spend in Yosemite National Park, so let’s make the most of it. One of the very best ways to truly explore Yosemite is on your own two feet. We opted for the Upper Yosemite Falls Trail, mostly because the trailhead was a short walk down the road from our lodge. We were on the trail by 6:30am. By 1pm, we were back in the Yosemite Valley Lodge parking lot.
After our strenuous hike the day before, it’s safe to say Derek and I weren’t up for any more physical activity. So instead, we opted for a scenic drive up to Glacier Point, home to some of the most dramatic views in all of Yosemite National Park.
From way up here above the valley, you can gaze upon iconic sights like Half Dome and Yosemite Falls, and see all the way across to Yosemite’s high country. From Glacier Point, the drive home to San Jose took us around 4.5 hours through the park’s south entrance.”

Expected summarized itinerary outcome could be,

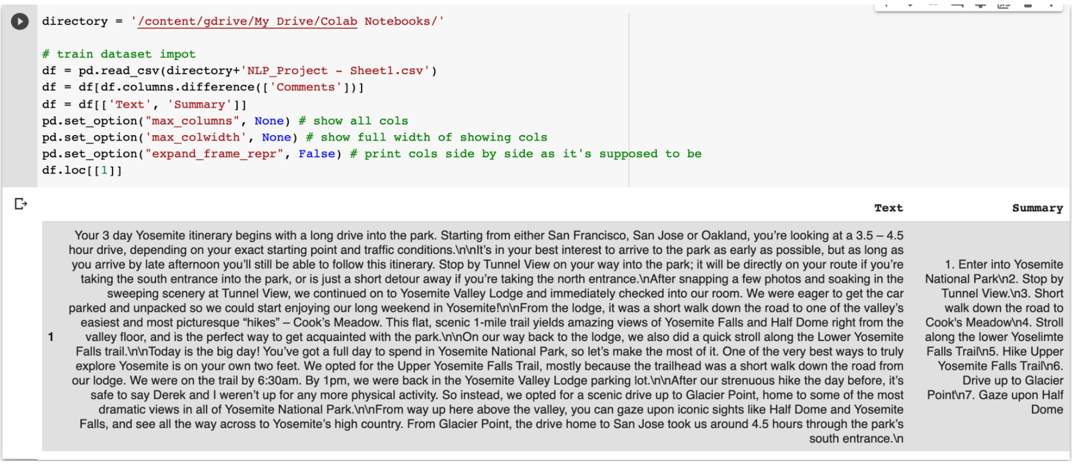
1. Enter into Yosemite National Park
2. Stop by Tunnel View
3. Short walk down the road to Cook's Meadow

4. Stroll along the lower Yosemite Falls Trail
5. Hike Upper Yosemite Falls Trail
6. Drive up to Glacier Point
7. Gaze upon Half Dome

The set of events described above can be considered the ground truth if we want to find a summary from the descriptive itinerary found in the form of textual blogs. We will show below how NLP can help us in finding the events, the places and the activities done by the bloggers to make a summary of their travel itinerary shared.

Dataset

We created a dataset based on the travel blogs available online [1]. Below is a snippet of our dataset, where we have texts and the corresponding summary as the ground truth.



```
directory = '/content/gdrive/My Drive/Colab Notebooks/'

# train dataset impot
df = pd.read_csv(directory+'NLP_Project - Sheet1.csv')
df = df[df.columns.difference(['Comments'])]
df = df[['Text', 'Summary']]
pd.set_option("max_columns", None) # show all cols
pd.set_option('max_colwidth', None) # show full width of showing cols
pd.set_option("expand_frame_repr", False) # print cols side by side as it's supposed to be
df.loc[[1]]
```

	Text	Summary
1	Your 3 day Yosemite itinerary begins with a long drive into the park. Starting from either San Francisco, San Jose or Oakland, you're looking at a 3.5 – 4.5 hour drive, depending on your exact starting point and traffic conditions.\n\nIt's in your best interest to arrive to the park as early as possible, but as long as you arrive by late afternoon you'll still be able to follow this itinerary. Stop by Tunnel View on your way into the park; it will be directly on your route if you're taking the south entrance into the park, or is just a short detour away if you're taking the north entrance.\nAfter snapping a few photos and soaking in the sweeping scenery at Tunnel View, we continued on to Yosemite Valley Lodge and immediately checked into our room. We were eager to get the car parked and unpacked so we could start enjoying our long weekend in Yosemite!\n\nFrom the lodge, it was a short walk down the road to one of the valley's easiest and most picturesque "hikes" – Cook's Meadow. This flat, scenic 1-mile trail yields amazing views of Yosemite Falls and Half Dome right from the valley floor, and is the perfect way to get acquainted with the park.\n\nOn our way back to the lodge, we also did a quick stroll along the Lower Yosemite Falls trail.\n\nToday is the big day! You've got a full day to spend in Yosemite National Park, so let's make the most of it. One of the very best ways to truly explore Yosemite is on your own two feet. We opted for the Upper Yosemite Falls Trail, mostly because the trailhead was a short walk down the road from our lodge. We were on the trail by 6:30am. By 1pm, we were back in the Yosemite Valley Lodge parking lot.\n\nAfter our strenuous hike the day before, it's safe to say Derek and I weren't up for any more physical activity. So instead, we opted for a scenic drive up to Glacier Point, home to some of the most dramatic views in all of Yosemite National Park.\n\nFrom way up here above the valley, you can gaze upon iconic sights like Half Dome and Yosemite Falls, and see all the way across to Yosemite's high country. From Glacier Point, the drive home to San Jose took us around 4.5 hours through the park's south entrance.\n	1. Enter into Yosemite National Park\n2. Stop by Tunnel View.\n3. Short walk down the road to Cook's Meadow\n4. Stroll along the lower Yoselimte Falls Trail\n5. Hike Upper Yosemite Falls Trail\n6. Drive up to Glacier Point\n7. Gaze upon Half Dome

We have the original texts from a travel blog in the column called 'Text' and 'Summary' contains the summarized itineraries developed by humans as our gold labeled data.

Baseline Approach

We implemented the baseline method by finding the names of the places and the corresponding action verbs in the blogs to build up the baseline itinerary using a dependency parser. We deduced a sentence into verb phrases and noun phrases. We picked Proper Nouns referring to location or organization as our points of interest. Then, based on the relation with the associated action verbs, we sequentially wrote down the place names and the corresponding activities. In short, our baseline approach incorporates the provision to extract proper nouns and verb phrases and concatenate them to form a sequence.

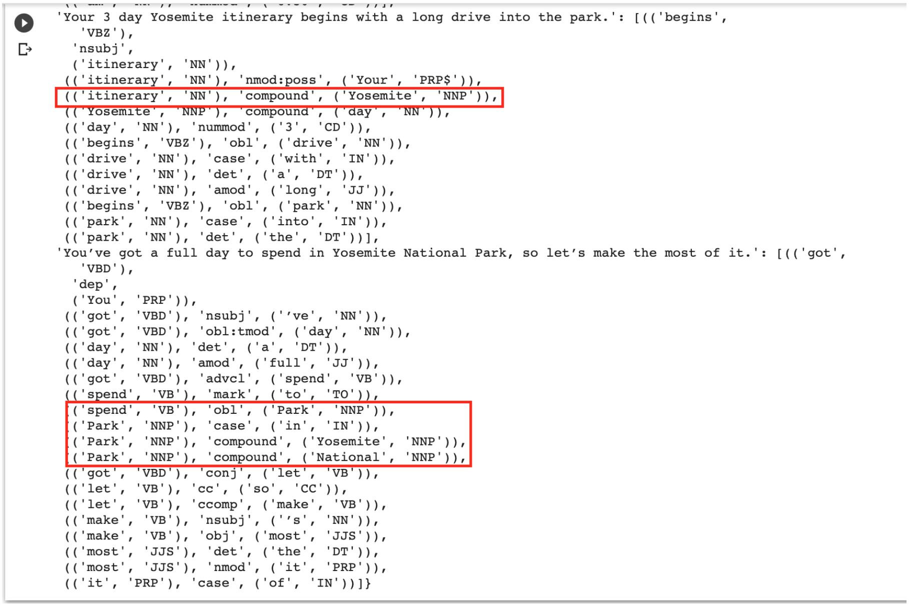
It should be noted here that we cannot rely on the periods or semicolons in the corpus to understand the inclusion of a clause or the completion of a sentence. Therefore, we did our own processing on the data before we applied NLP techniques from the NLTK package. The preprocessing steps that we have followed are:

1. Sentence Tokenization of the input texts
2. Lemmatize the tokens

After the preprocessing phase, we built our baseline model using nltk dependency parsing [2] and extracted the noun phrases along with its related verb phrase to formulate the itinerary tuples.

Preprocessing step 1 was done to fragment our corpus into smaller chunks, i.e. sentence tokens and step 2 helped us get the base forms of the verbs and brought ease and simplicity in our subsequent data processing. Finally, we applied Stanford Dependency Parser and among other pairs, we identified NNP/NNPS (Proper Noun- Singular and Plural forms) and VBP (Verb Phrases). This step helped us establish the link between verbs and nouns in a sentence. At this stage, our outcome was about which nouns are related to which verbs that emerged in the form of an imperative sentence; e.g., visit X, stop by Y, etc.

However, "Lower Falls Yosemite Trail" was considered separate proper nouns, which was undesirable. It also could not differentiate between the person noun phrase vs the location noun phrases. Moreover, if there was a sentence where the main verb was connected with more than one Proper Noun, e.g., “We visited X, Y and Z.”, the dependency parser could not identify one verb related to three of the names separately. Finally, travelers may include a place in their itinerary that they did not cover in that particular tour, instead a place in that tour reminded them of another place. Verb and noun tuple extracted from the dependency parsing contains such reminded places in the itinerary. We mitigated these challenges in our proposed method.



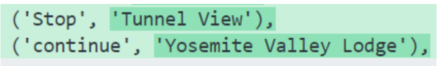
```
'Your 3 day Yosemite itinerary begins with a long drive into the park.': [(('begins',
   'VBZ'),
  'nsubj',
  ('itinerary', 'NN')),
 (('itinerary', 'NN'), 'nmod:poss', ('Your', 'PRP$')),
 (('itinerary', 'NN'), 'compound', ('Yosemite', 'NNP')),
 (('Yosemite', 'NNP'), 'compound', ('day', 'NN')),
 (('day', 'NN'), 'nummod', ('3', 'CD')),
 (('begins', 'VBZ'), 'obl', ('drive', 'NN')),
 (('drive', 'NN'), 'case', ('with', 'IN')),
 (('drive', 'NN'), 'det', ('a', 'DT')),
 (('drive', 'NN'), 'amod', ('long', 'JJ')),
 (('begins', 'VBZ'), 'obl', ('park', 'NN')),
 (('park', 'NN'), 'case', ('into', 'IN')),
 (('park', 'NN'), 'det', ('the', 'DT'))],
'You’ve got a full day to spend in Yosemite National Park, so let’s make the most of it.': [(('got',
   'VBD'),
  'dep',
  ('You', 'PRP')),
 (('got', 'VBD'), 'nsubj', ('’ve', 'NN')),
 (('got', 'VBD'), 'obl:tmod', ('day', 'NN')),
 (('day', 'NN'), 'det', ('a', 'DT')),
 (('day', 'NN'), 'amod', ('full', 'JJ')),
 (('got', 'VBD'), 'advcl', ('spend', 'VB')),
 (('spend', 'VB'), 'mark', ('to', 'TO')),
 (('spend', 'VB'), 'obl', ('Park', 'NNP')),
 (('Park', 'NNP'), 'case', ('in', 'IN')),
 (('Park', 'NNP'), 'compound', ('Yosemite', 'NNP')),
 (('Park', 'NNP'), 'compound', ('National', 'NNP')),
 (('got', 'VBD'), 'conj', ('let', 'VB')),
 (('let', 'VB'), 'cc', ('so', 'CC')),
 (('let', 'VB'), 'ccomp', ('make', 'VB')),
 (('make', 'VB'), 'nsubj', ('’s', 'NN')),
 (('make', 'VB'), 'obj', ('most', 'JJS')),
 (('most', 'JJS'), 'det', ('the', 'DT')),
 (('most', 'JJS'), 'nmod', ('it', 'PRP')),
 (('it', 'PRP'), 'case', ('of', 'IN'))]}
```

Output (verb,noun) tuple from baseline dependency parser

The figure shows the link between '*spend*' and '*Yosemite*' for the sentence "*You've got a full day to spend in Yosemite National Park*"

Proposed Method

Our proposed method hopes to resolve a couple of issues that made the baseline approach a not-so-practical method. The first challenge posed by the baseline dependency parser is separated noun phrases which together makes a location valid. Therefore, we needed to tokenize our words in a way so that it considers the entire name of the location together instead of identifying them as separate noun phrases. To mitigate this challenge, we used SpaCy to identify the Named Entity from every corpus. It helped us to include "Lower Falls Yosemite Trail" as one entity.



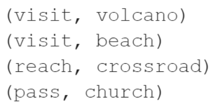
```
('Stop', 'Tunnel View'),
('continue', 'Yosemite Valley Lodge'),
```

Output with dependency parsing after tokenization with named entity

This figure shows that 'Yosemite Valley Lodge' has been considered as a single entity for the sentence "*After snapping a few photos and soaking in the sweeping scenery at Tunnel View, we continued on to Yosemite Valley Lodge.*"

To discard the places that were actually not covered in the itinerary like "reminded Alaska", we added a verb filter to discard those tuples. However, this method was not as effective and we aim to further improve that using a geolocation based token method.

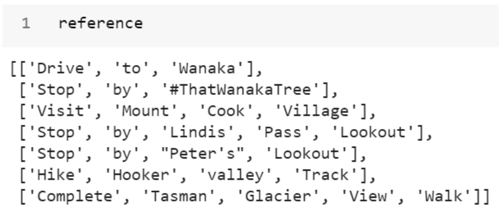
We also wanted to do event extraction based on the relationship between the ROOT, nsubj and dobj. By doing so, we could identify the participating action verbs and the objects that are connected to them. This way, we found a sequence like below, whereas the original text was: "*I visited the world-famous active volcano Gunung Api and went for a swim in Natsepa beach from there. You will reach a crossroads where you can turn right onto the Pineapple Route. Be sure to pass by the Saint Joseph Catholic Church, which is as cute as can be.*"



```
(visit, volcano)
(visit, beach)
(reach, crossroad)
(pass, church)
```

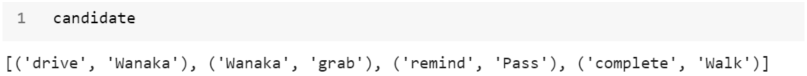
BLEU Score

Here, we see a part of the reference (ground truth) and candidate (our hypothesis).



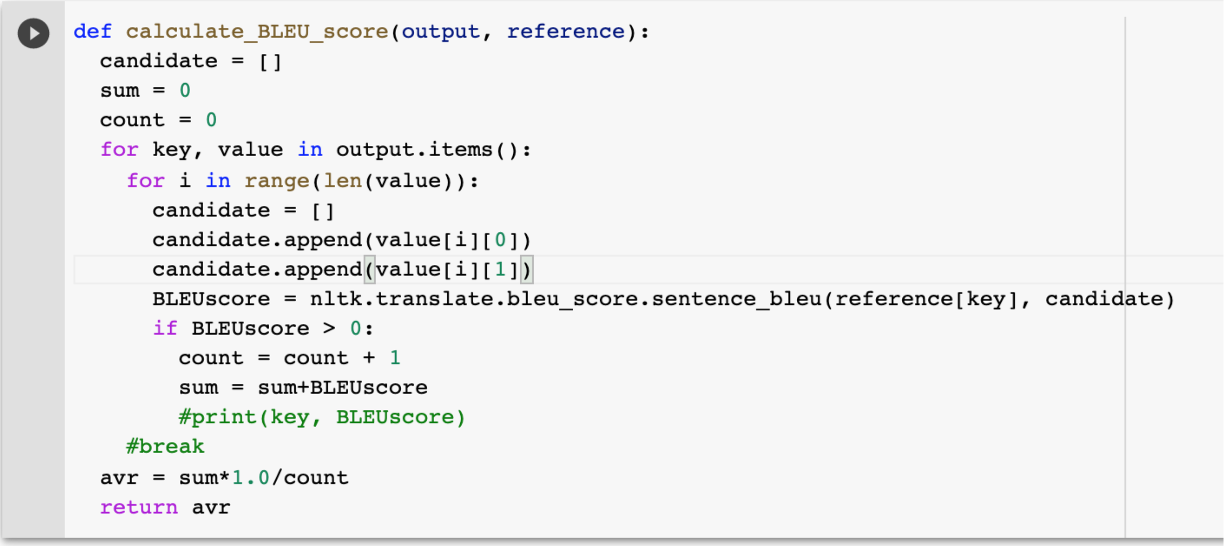
```
reference
```

```
[['Drive', 'to', 'Wanaka'],
 ['Stop', 'by', '#ThatWanakaTree'],
 ['Visit', 'Mount', 'Cook', 'Village'],
 ['Stop', 'by', 'Lindis', 'Pass', 'Lookout'],
 ['Stop', 'by', "Peter's", 'Lookout'],
 ['Hike', 'Hooker', 'valley', 'Track'],
 ['Complete', 'Tasman', 'Glacier', 'View', 'Walk']]
```



```
candidate
```

```
[('drive', 'Wanaka'), ('Wanaka', 'grab'), ('remind', 'Pass'), ('complete', 'Walk')]
```



```
def calculate_BLEU_score(output, reference):
  candidate = []
  sum = 0
  count = 0
  for key, value in output.items():
    for i in range(len(value)):
      candidate = []
      candidate.append(value[i][0])
      candidate.append(value[i][1])
      BLEUscore = nltk.translate.bleu_score.sentence_bleu(reference[key], candidate)
      if BLEUscore > 0:
        count = count + 1
        sum = sum+BLEUscore
        #print(key, BLEUscore)
    #break
  avr = sum*1.0/count
  return avr
```

Method	**BLEU score**
Baseline	**48.5**
Proposed	**60.4**

We achieved a 60.4% BLEU score using our proposed method compared to 48.5% achieved by our baseline method.

REFERENCES

1. http://thewanderingblonde.com/category/itineraries/
2. https://www.analyticsvidhya.com/blog/2019/02/stanfordnlp-nlp-library-python/
3. Ho, C.-I. and Y.-L. Lee, The development of an e-travel service quality scale. Tourism management, 2007. 28(6): p. 1434-1449.
4. Yuan, H., et al., Make your travel smarter: Summarizing urban tourism information from massive blog data. International Journal of Information Management, 2016. 36(6): p. 1306-1319.