LSTM을 이용한 사용자 맞춤 여행 경로 추천 모델 연구

김민선○ 조서영 이형준
이화여자대학교
minsk518@gmail.com, stellamore99@gmail.com, hyungjune.lee@ewha.ac.kr

A Study on the Customized Travel Route Recommendation Model Using LSTM

Minsun Kim○ Seoyoung Jo, HyungJune Lee
Ewha Womans University

요 약

본 논문에서는 시간 정보를 포함한 사용자 맞춤 여행 경로를 추천하기 위해 군집화와 LSTM을 이용한다. 사용자의 경로 특성을 명확히 알아내기 위해 방문 장소들의 특징을 나타낼 수 있는 알파벳을 사용하여 이동 경로를 문자열로 치환하였고, 이를 이용하여 여러 사용자의 이동 경로를 군집화하였다. 여러 머신러닝 군집화 방법 비교 결과 레벤슈타인 거리와 계층적 군집화를 사용하는 것이 효과적임을 알 수 있었으므로 해당 방법을 채택하였다. 군집화 후 사용자가 어느 군집에 속하는지 파악하기 위해 LCS를 이용했으며, 해당 군집의 데이터를 이용해 학습한 LSTM encoder-decoder 모델에 사용자의 초기 경로를 넣어 이후 경로를 예측받았다. 예측 결과가 실제 경로와 비슷하게 나타났으므로 모델이 합리적인 추천을 할 수 있음을 알 수 있었다.

1. 서 론

여행 산업은 상당한 규모를 지닌 산업으로, 한국관광공사에서 발행한 보고서에 따르면 기업의 매출액(Revenue)을 기준으로 전 세계 관광산업의 시장규모를 추정한 결과 2011년 약 1,437십억 달러(약 1,650조 원)이던 관광산업은 2019년 약 1,692십억 달러(약 1,940조 원)로 약 17.74% 증가했다(IBISWorld2), 2020). 하지만 코로나19가 발생한 2020년 약 1,541십억 달러(약 1,770조 원)로 전년대비 8.90% 감소하여 큰 폭의 감소세를 보였다. 2021년 예상 시장규모는 약 1,702십억 달러(약 1,950조 원)로 코로나19 팬데믹으로 관광산업이 큰 타격을 받았던 2020년에 비해 10.43% 성장할 것으로 기대하고 있다[1]. 코로나19로 인해 잠시 주춤했던 여행 산업이 다시 활기를 띨 것으로 예상되면서 여행을 준비하는 사람들 또한 늘고 있다. 여행을 준비할 때는 어떤 경로로 여행을 하느냐가 가장 중요한 고려사항 중 하나이므로 이는 여행 산업과 관련 연구에서 중요하게 다뤄지는 문제이다. 기존의 서비스나 연구에서는 사용자의 여행 경로 특성을 파악할 때 GPS 경로의 지리적 유사성만 고려하거나, 사용자의 취향 파악을 위해 GPS 데이터에 더불어 LBSN (Location-based Social Network) 데이터를 사용했다. 이는 추가적인 데이터 가공과 연산을 필요로 한다. 또한 사용자에게 장소들을 예측해줄 뿐 해당 장소에 얼마나 머무르면 좋을지를 포함해서 경로를 추천하지는 못하는 한계점을 가지고 있었다.

본 연구는 앞선 연구들과는 달리 GPS 데이터만을 이용하여 여행 경로의 특성을 추출하고, 지리적 경로뿐만 아니라 장소의 특성들까지 반영한 비교를 시도한다. GPS 데이터를 가공하여 방문 장소의 특성을 반영한 알파벳 시퀀스로 나타내고, 이 알파벳 시퀀스 간의 비교와 군집화를 통해 여행 경로 추출과 비교에서 그간 잘 다뤄지지 않았던 새로운 연구 방향의 가능성을 제시한다. 더불어 자연어처리 분야에서 많이 쓰이는 LSTM encoder-decoder 모델을 도입하여 시간 정보까지 포함한 이후 경로를 추천받는다.

2. 관련 연구

국내외에서 여행자의 여행 패턴을 파악하고 이를 여행 경로 추천에 이용하고자 하는 연구는 활발히 있어 왔다. 소셜 미디어를 이용하여 사용자의 선호를 고려해 여행 경로를 추천하는 연구 [2], 사용자의 여행 종료 시간과 선호를 고려하여 여행 경로를 추천하는 연구 [3], [4], [5]와 같이 다양한 방법을 통해 사용자에게 맞는 여행 경로를 추천하고자 하는 연구를 확인할 수 있었다. 그러나 앞선 연구들에서는 사용자의 선호도를 파악할 때 GPS 데이터가 아닌 외부 맥락의 데이터를 함께 사용했으며, 경로 비교 시 GPS 경로의 지리적 유사성만 고려했다는 한계를 갖는다. 또한, 사용자 개개인에 집중하였고 비슷한 특성을 갖는 사용자 집단에 대한 고려는 소홀했으며 시간을 고려한 경로 추천이 아닌 특정 장소들만 추천했다는 한계점을 지닌다. 따라서 본 연구에서는 이러한 한계를 보완하고자 장소 특성을 반영한 문자열 시퀀스를 생성하고, 머신 러닝 군집화 기법을 이용해 군집화함으로써 사용자 집단 경로 분석을 위한 초석을 마련하고자 한다. 더불어 시간 정보를 고려한 경로 추천을 통해 여행 경로를 짜는 데 쓰이는 시간을 줄임으로써 경로 생성의 효율성을 높이는 데 기여하고자 한다.

3. 데이터

3.1 데이터셋

본 연구에서는 다수의 사용자로부터 실제로 수집된 GPS 데이터를 사용하고자 하였기 때문에 2007년 4월부터 2012년 8월까지 Microsoft Research Asia에서 중국 베

이징 시를 중심으로 수집한 Geolife Dataset을 사용하였다. 수집된 데이터의 사용자 수는 182명, 경로 수는 약 15,000개이고 위도, 경도 좌표, 수집된 시간, 고도 등의 여러 정보가 포함되어 있다. 이 중 위도, 경도 좌표와 수집된 시간, 경로 고유번호와 사용자 고유번호를 본 연구에 이용하였다.

3.2 데이터 전처리

본 연구에서 사용하려는 데이터 형식은 장소와 머무른 시간을 나타내는 문자열 시퀀스이기 때문에 데이터 전처리 과정을 수행하였다. Scikit-mobility에서 제공하는 Stay Point Detection을 이용하여 사용자가 머물렀다고 판단할 수 있는 Stay point들을 선별했다. 이후, Google에서 제공하는 Places API, Maps Javascript API를 이용하여 Reverse-Geocoding으로 선별한 좌표들의 장소 정보를 추출하였다. 이때 장소의 성격을 나타내는 타입은 30개로 제한하였다. 30개의 선별된 타입을 가진 stay point와 raw data를 비교하여 알파벳 하나당 5분 머물렀음을 뜻하는 사용자의 이동 문자열 시퀀스를 생성하였다. 최종 생성한 문자열 시퀀스는 총 2781개이다.

[표 1] 시퀀스 형태와 그 예시

시퀀스 형태	시퀀스 = (장소$_1$, 장소$_2$, ⋯, 장소$_n$)			
시퀀스 예시	(Kx0003, Sx0012, Sx0012, Tx0001, Lx0004, Ag0001, Ux0004, Rx0020, Sx0198, Ux0006, Rx0021, -, -, -)			
장소 타입	Kx	공원	Sx	상점
	Tx	환승 터미널	Lx	숙박
	Ag	미술관	Ux	대학교
	Rx	음식점	-	이동 중

4. 군집화

본 연구에서 목표하고자 한 것은 가공을 거친 데이터를 여러 군집화 모델을 사용해 분석한 후, 최적의 군집화 모델을 찾아 분류하여 LSTM 모델을 제작하는 것이다. 코사인 유사도를 유사도 측정 방법으로 사용한 계층적 군집화, K-평균 군집화, DBSCAN 모델의 결과를 비교한 후, 문자열 군집화에 주로 사용되는 계층적 군집화 모델을 이용해 코사인 유사도와 레벤슈타인 거리를 사용했을 때의 결과를 비교하였다. 비교 후 최적의 군집화 방법을 찾아 실루엣 계수(Silhouette Coefficient)와 군집 내 거리(Intra-cluster Distance)를 군집 개수 별로 확인한 후 적절한 군집 개수를 지정해 분류하였다.

4.1 코사인 유사도

코사인 유사도[6]는 두 벡터 간의 코사인 각도를 이용하여 구할 수 있는 두 벡터의 유사도를 의미한다. TF-IDF 행렬을 통해서 문자열을 토큰화한 후 문자열 간의 유사도를 측정할 수 있다. 이는 [수식 1]과 같이 정의되며, A와 B는 비교하고자 하는 각 문자열을 벡터화한 것을 의미한다. -1 이상 1 이하의 값을 가지며 값이 1에 가까울수록 유사도가 높다고 판단할 수 있다.

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

[수식 1] 코사인 유사도

4.2 레벤슈타인 거리

본 연구에서 추가로 사용한 문자열 간의 유사도를 판별하는 도구는 레벤슈타인 거리 [7]이다. 삽입, 삭제, 변경을 몇 번 해야 두 문자열을 동일하게 만들 수 있는지를 측정하여 그 최솟값을 구해 유사도 판단의 척도로 다룬다.

4.3 계층적 군집화

계층적 군집화[7]는 문자열 군집화에 많이 사용되는 군집화 방법이다. 계층적 군집화는 유사도를 미리 계산해야 하기 때문에 두 시퀀스 사이의 코사인 유사도와 레벤슈타인 거리를 계산하는 함수를 제작해 유사도를 계산하였다. 두 점을 연결할 때 사용한 방식은 Ward linkage이다.

4.4 DBSCAN

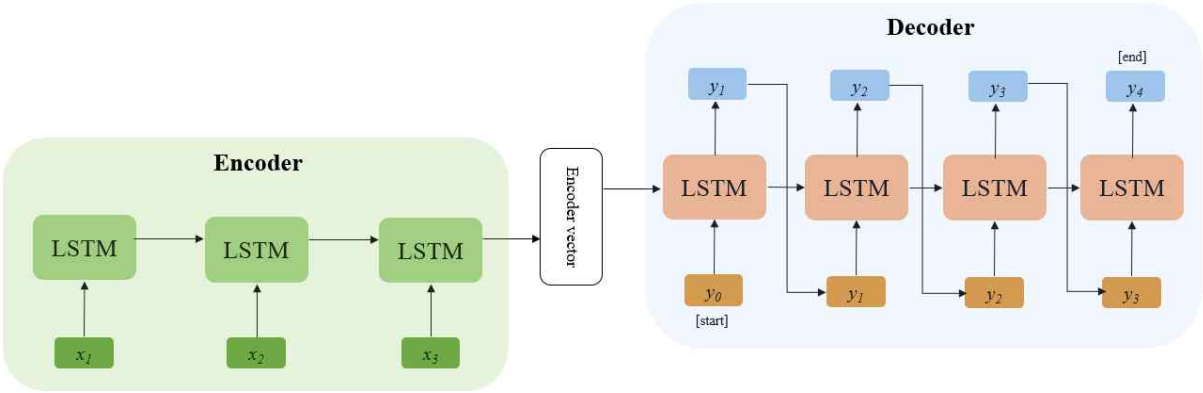
DBSCAN [8]은 밀도 기반의 군집화 방식으로 어느 특정 벡터부터 시작해 반경 내 기준치만큼의 점들이 존재한다면 군집화하는 방식으로 진행된다. DBSCAN은 중심점으로부터의 반경인 eps를 지정해야 하기 때문에 eps별 노이즈의 개수, 군집의 개수, 실루엣 계수를 확인하며 결과를 확인하였다.

4.5 K-평균 군집화

K-평균 군집화[9]는 주어진 데이터를 K개의 군집으로 분류하는 알고리즘으로, 각 군집과 거리 차이의 분산을 최소화하는 방식으로 동작한다. 해당 알고리즘은 일반적으로 유클리디안 거리를 사용해 유사도를 측정하지만, 본 연구에서는 DBSCAN에서 사용한 방식과 같이 문자열을 TF-IDF를 이용해 문자열 벡터로 변경한 후, 코사인 유사도를 계산해 측정하였다. K-평균 군집화는 군집 개수인 K를 지정해주어야 하기에 군집 개수와 실루엣 계수를 함께 확인하며 실험을 진행하였다.

4.6 RelaxedLCS

본 연구에서는 비슷한 경로를 가진 사람들의 경로에 기반한 추천은 더 사용자 맞춤형일 것이라는 가정을 가지고 사용자를 가장 유사한 군집에 할당하고자 한다. 따라서 사용자의 데이터를 대표하는 값과, 각 군집을 대표하는 값을 추출해 레벤슈타인 거리를 계산한 후 가장 유사한 군집에 할당했다. 사용자와 가장 유사한 군집을 찾기 위해서 최장 공통 부분수열(Longest Common Subsequence, LCS)의 개념을 이용하였다. 이는 [11]에서 사용한 방식과 같은데, 먼저 최장 공통 부분수열을 구한 후 더 긴 시퀀스의 가장 앞 항목과 가장 마지막 항목을 추가해 재귀적으로 계산하는 방식(relaxed LCS)이다.



[그림 1] LSTM encoder-decoder 모델

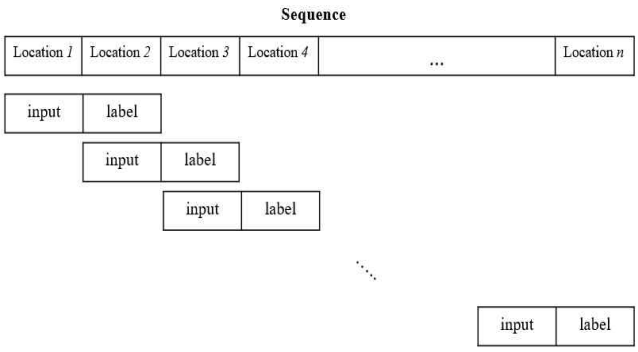
5. LSTM Encoder-Decoder

군집화 결과 나눠진 각 군집 별로 LSTM을 이용한 Encoder-decoder 모델에 넣어 학습시킨다. LSTM은 게이트를 활용하여 기존 RNN의 기울기 소실 문제를 해결하고자 한 RNN의 변형 모델이다[10]. Forget gate, input gate, output gate와 cell을 이용하여 어떤 값을 얼마만큼 반영할지 조절하고, cell이 memory 역할을 해서 long-term memory를 저장한다. 기울기 소실 문제를 완화하였으므로 보다 긴 길이의 입력값에 대해서도 비교적 좋은 학습 결과를 낼 수 있다.

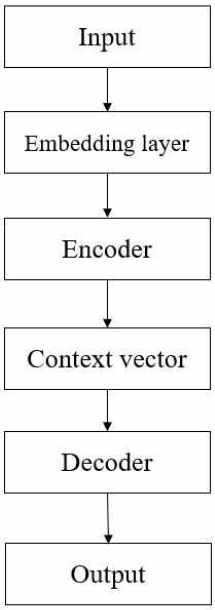
LSTM을 이용한 encoder-decoder 모델은 다양한 길이의 입력값과 출력값을 처리할 수 있으므로 기계 번역에서 많이 쓰이고 있는 모델이다. encoder에서 입력값을 하나의 벡터로 압축하고 이를 decoder에 넘겨주면, decoder는 이 정보를 바탕으로 최종 출력값을 생성한다.

5-1. LSTM encoder-decoder 모델 학습

앞서 생성한 장소로 이뤄진 문자열 시퀀스들을 '장소-정수 딕셔너리'를 통해 정수 시퀀스로 변환하고, 이 시퀀스를 Sliding window 방식으로 입력값, 출력값으로 나눈 후 이를 모델 학습에 이용했다.



[그림 2] Sliding window 방식으로 데이터 생성



[그림 3] 모델 구조

[그림 2]과 같은 방식으로 데이터를 생성한 후 이를 학습에 이용하는데, 모델에 넣기 전 Embedding layer를 거친다. 일반적인 딥러닝 모델에서는 'one-hot encoding'을 사용하기도 하나, 본 연구에서 다루는 데이터는 장소의 개수가 많아 one-hot encoding을 적용할 경우 벡터에 0이 너무 많아지는 희소성 문제가 나타날 수 있으며 메모리 부하가 심해지므로 Embedding layer를 사용하여 조밀한 벡터 표현을 할 수 있도록 했다. 간단하게 도식화한 모델 구조는 [그림 3]과 같다. 정수로 대응된 시퀀스의 각 장소가 Embedding layer를 거치며 각각 벡터로 변환되고, 이 벡터가 Encoder-decoder 모델에 들어가 학습에 사용된다.

6. 실험 및 결과 분석

6-1. 군집화 방법 비교

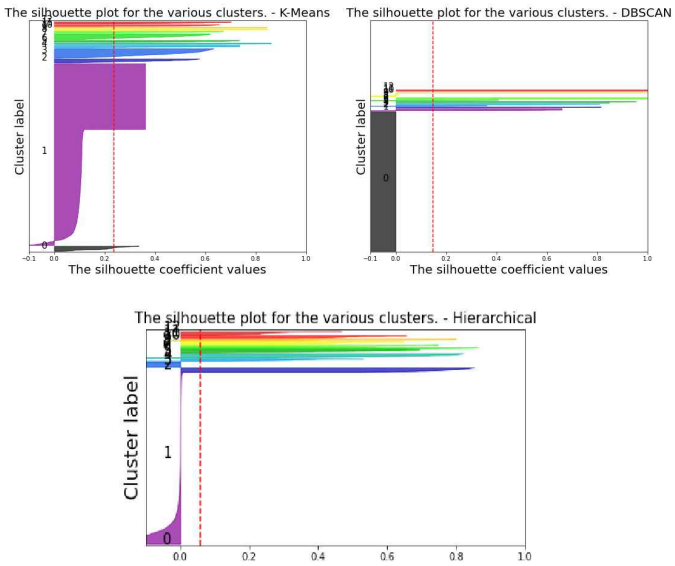
문자열 시퀀스로 치환한 GPS 데이터를 효과적으로 군집화할 수 있는 방법을 찾기 위해 다양한 실험을 수행하였다. 본 실험을 위해 사용한 연산 환경은 Intel(R) Core(TM) i5-8265U CPU, 16GB RAM, Intel(R) UHD Graphics 620 그래픽카드가 장착된 LG GRAM 15Z990-VA56K 이다. 먼저, 문자열을 치환한 벡터 사이의 각도를 이용해 유사도를 측정하는 코사인 유사도를 이용하였다. 계층적, K-평균, DBSCAN의 세 군집화 모델을 사용하였으며, 비교를 위해 군집화를 평가하는 척도로 활용되는 실루엣 계수(Silhouette Coefficient)와 클러스터별 시퀀스 개수를 확인하였다. 코사인 유사도를 사용한 군집화 성능 비교 결과는 [표 2]와 같다. 성능 비교에 사용한 군집 개수는 13개이다. 13개로 설정한 이유는

DBSCAN을 실행할 때 설정해야 하는 중요한 parameter인 eps를 조절하며 결과를 확인했을 때, 군집 개수가 110개 이상, 13개, 2개 이하였기 때문이다. 따라서 극단적이지 않은 값인 13개를 선정했으며 비교를 위해 다른 방식에도 군집 개수를 13개로 지정해 결과를 확인하였다.

[표 2] 세 군집화 모델의 실루엣 계수 비교

군집화 모델	실루엣 계수 (군집 13개)	소요시간
계층적 군집화	0.057	2시간
K-평균 군집화	0.243	5초
DBSCAN	0.148	5초

측정한 소요 시간은 전체 데이터의 유사도 계산과 군집화 과정에 사용된 시간의 합이다. K-평균과 DBSCAN과는 달리, 계층적 군집화를 진행하기 위해선 코사인 유사도를 미리 계산해야 하기 때문에 매 실행 시 약 2시간이 소요되었다. K-Means와 DBSCAN은 10초 이하의 시간이 소요되었다. 전체 실루엣 계수는 극단적으로 치우친 값에 의해 실제 군집화 결과를 정확하게 보여줄 수 없을 가능성이 존재하기에, 군집별 실루엣 계수를 그래프로 시각화하였다. 각 방식의 군집 별 실루엣 계수 그래프는 [그림 4]과 같다. 추가로 군집 별 포함된 시퀀스의 개수를 비교했을 때, 계층적 군집화 모델의 한 군집에 2312개의 데이터, K-평균 군집화 모델의 한 군집엔 2289개의 데이터, 그리고 DBSCAN의 한 군집에는 1762개의 데이터가 있는 것을 확인할 수 있었다. 세 결과 모두 데이터 쏠림 현상을 발견할 수 있었다.

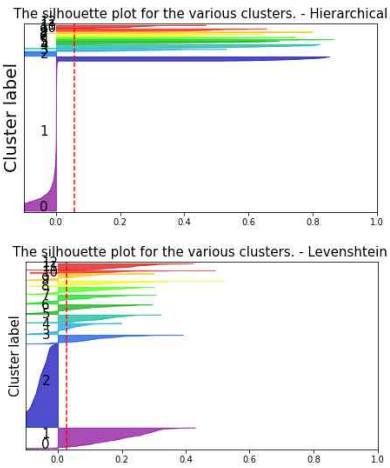


[그림 4] 코사인 유사도를 사용한 K-Means, DBSCAN, 계층적 군집화 모델의 실루엣 계수 시각화

세 방식 모두 균일하지 않은 시퀀스 분포와 실루엣 계수를 보였기 때문에 문자열을 군집화할 때 가장 많이 사용되는 레벤슈타인 거리를 이용한 계층적 군집화 결과를 비교하였다. 앞의 방식과 동일하게 군집 개수는 13개로 진행하였다. 레벤슈타인 거리를 측정해 군집화를 진행하는 과정엔 평균적으로 20분이 소요되었다. 실루엣 계수를 확인한 결과, [표 3]과 [그림 5]에서 확인할 수 있듯 코사인 유사도를 사용한 계층적 군집화 뿐만 아니라 K-평균, DBSCAN 보다 안정적이고 균일한 분포를 확인할 수 있었다.

[표 3] 유사도 측정 방식을 달리한 계층적 군집화 모델의 실루엣 계수 비교

유사도 측정 방식	실루엣 계수 (군집 13개)	소요시간
코사인 유사도	0.057	2시간
레벤슈타인 거리	0.049	20분



[그림 5] 코사인 유사도와 레벤슈타인 거리를 이용한 계층적 군집화 모델 실루엣 계수 시각화

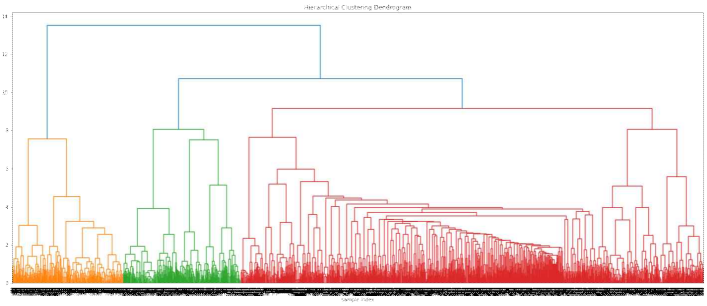
6-2. 군집화 결과 및 분석

레벤슈타인 거리를 유사도 측정 기준으로 사용한 계층적 군집화를 본 연구의 군집화에 사용하였다. 이때 모든 두 쌍의 문자열 시퀀스에 대해 유사도를 측정하고 이를 레벤슈타인 거리로 변환하여 사용했다. 유사도를 측정하는 과정에 사용된 시퀀스는 중복 요소가 제거된 trimmed sequence이다. [11]에서 차용한 방식을 이용해, 시퀀스의 거리가 시퀀스 사이의 유사도에 영향을 주지 않도록 삽입(insertion)과 삭제(deletion)에는 가중치를 부여하지 않았고, 변경(substitution)에는 실제 두 장소의 거리를 계산해 500m 이내일 때만 가중치를 부여하는 방식으로 유사도를 계산하였다. 이는 [수식 2]와 같이 정의된다. 짧은 시퀀스보다 긴 시퀀스 사이의 유사도가 더 크게 나올 수 밖에 없기에 유사도를 측정한 두 시퀀스 중 더 짧은 시퀀스의 길이로 나누어 정규화를 진행했다.

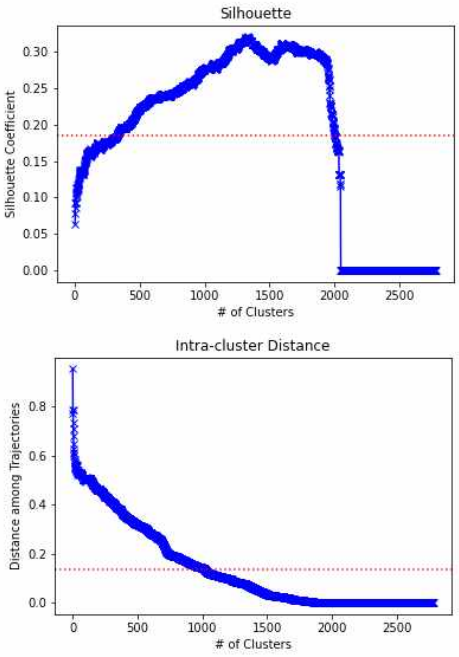
$$S_{X,Y}(i,j) = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ \max \begin{cases} 0 \\ S_{X,Y}(i-1,j) \\ S_{X,Y}(i,j-1) \\ S_{X,Y}(i-1,j-1)+1 \text{ if } d<500 \end{cases} & otherwise \end{cases}$$

[수식 2] 시퀀스 유사도

측정한 유사도로 군집화를 진행한 결과, 덴드로그램은 [그림 6]과 같이 나온다. 최적의 군집 개수를 찾기 위해 군집 개수를 1개부터 2781개까지 늘려가며 실루엣 계수, 군집 내 거리를 계산한 결과는 [그림 7]과 같다. 실루엣 계수와 군집 내 거리를 고려한 최적의 군집 개수는 100개, 629개 또는 1383개이다.

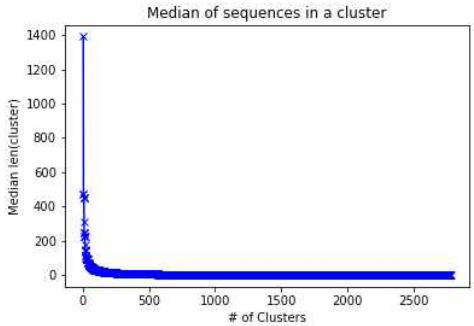


[그림 6] 군집화 결과 덴드로그램



[그림 7] 군집 개수 별 실루엣 계수, 군집 내 거리 그래프

하지만 문자열 시퀀스를 군집으로 나누는 것이 본 연구의 최종 목표가 아니고 군집 별 LSTM 모델을 제작하는 것이 연구의 목적인데, 군집 수가 많아지면 한 군집에 속한 시퀀스 개수가 30개 이하로 적어져 학습에 어려움이 생긴다. 따라서 군집 개수를 늘려가며 군집 개수 별 군집에 속한 시퀀스 개수의 중간값을 계산해 보았고 이는 [그림 8]와 같다. LSTM 모델 학습에 필요한 군집 별 시퀀스 개수의 최소 한계점이 200개라고 판단하여 확인했을 때, 10개의 군집 개수를 갖는 것이 이상적이라고 판단하여 최종적으로 총 10개의 군집으로 나누었다. 10개의 군집에 대한 시퀀스 정보, 실루엣 계수, 군집 내 거리 정보는 [표 4]에서 확인할 수 있다. 이 때 시퀀스 길이는 중복을 제거한 시퀀스의 길이이다.



[그림 8] 군집 개수 별 시퀀스 개수 중간값

[표 4] 군집 10개 분석 정보

군집	시퀀스 개수	시퀀스 길이 (평균/최대/최소)	실루엣 계수	군집 내 거리
1	310	6.32 / 25 / 1	0.311874	0.49558
2	1211	6.73 / 36 / 1	-0.081802	0.97947
3	139	6.47 / 22 / 2	0.218992	0.53899
4	226	7.63 / 29 / 2	0.338558	0.50366
5	180	7.04 / 23 / 2	0.195998	0.68743
6	152	6.99 / 43 / 3	0.161593	0.64728
7	93	8.19 / 45 / 2	0.430315	0.42835
8	126	7.21 / 20 / 2	0.180205	0.77987
9	251	7.96 / 25 / 2	0.109043	0.71952
10	94	6.27 / 32 / 2	0.456459	0.50505
평균 실루엣 계수			0.10691	
평균 군집 내 거리			0.62852	

6-3. Encoder-decoder 모델 학습 결과 및 분석

군집화 결과 생성된 각 군집 별로 Encoder-decoder 모델 학습을 진행했다. 연산 환경은 Intel core i9-10900 CPU, 64G RAM, Intel(R) UHD Graphics 630 그래픽카드가 장착된 LG Desktop이다. 각 군집 내의 시퀀스를 Sliding window 방식으로 input과 label로 나눈 후 랜덤하게 섞었다. 이때 input 부분의 장소 개수는 15개, label 부분의 장소 개수는 10개가 되도록 하였다. 그 후 Trainig data : Validation data : Test Data = 8 : 1 : 1 이 되도록 분할하였다. 각 군집 별 데이터 개수는 [표 5]와 같다.

Encoder-decoder 모델에 training data를 넣어 학습하고 과적합 문제가 발생하지 않는지 확인화기 위해 validation data를 이용했다. 학습 결과 [표 6]에서 볼 수 있는 것처럼 loss가 0에 가깝게 나온 것을 확인할 수 있었다. 학습 과정에서의 loss 변화 추이를 관찰하기 위한 그래프는 [그림 9]과 같다. Cluster 3의 loss 그래프를 대

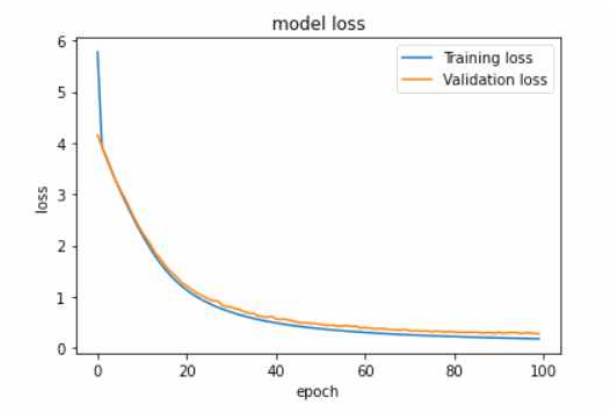
표로 나타낸 것이며 다른 군집에서의 학습 결과도 [그림 9]과 같이 꾸준히 줄어드는 모습을 보였다.

[표 5] 군집 별 데이터 개수

데이터 종류 / 군집	Training data	Validation / test data
Cluster 1	19405	2425
Cluster 2	63823	7977
Cluster 3	6727	840
Cluster 4	14535	1816
Cluster 5	8379	1047
Cluster 6	14997	1874
Cluster 7	6176	772
Cluster 8	8958	1119
Cluster 9	14969	1871
Cluster 10	4401	549

[표 6] 군집 별 loss

loss / 군집	loss
Cluster 1	0.2119
Cluster 2	0.6644
Cluster 3	0.1762
Cluster 4	0.1798
Cluster 5	0.1415
Cluster 6	0.1434
Cluster 7	0.2095
Cluster 8	0.1885
Cluster 9	0.1575
Cluster 10	0.2847



[그림 9] Cluster 3의 loss 그래프

학습된 모델이 실제로 의미 있는 결과를 출력할 수 있는지 확인하기 위해 test data를 이용해 결과를 출력하고 해당 결과를 MA, MRE를 이용해 평가했다. MA는 순서는 고려하지 않고 장소만 고려한 정확도로, 출력 시퀀스가 정답 시퀀스에 있는 장소 중 몇 개를 맞췄느냐를 나타낸다. MRE는 시퀀스 순서도 고려한 정확도로, 출력 시퀀스와 정답 시퀀스가 얼마나 비슷한지를 나타낸다. MA는 값이 1에 가까울수록, MRE는 값이 0에 가까울수록 모델이 정확하게 맞혔음을 의미한다.

$$MA = \frac{N_{correct}}{N_{locations-in-target}}$$

[수식 3] MA

$$MRE = \frac{Edit(Seq_{predicted}, Seq_{target})}{\max(length(Seq_{predicted}, Seq_{target}))}$$

[수식 4] MRE

군집 별 MA, MRE 평균을 보면 [표 7]에서 볼 수 있듯 MA 값이 대체로 0.5를 넘는 값을 가졌는데, 이는 장소 예측에 대해 절반 정도 혹은 넘게 맞혔다는 의미이다. 또한 MRE의 경우 0.5 부근의 값을 가졌는데, 순서를 고려한 시퀀스에 대해서도 절반 정도는 맞혔다는 뜻이다. 순서를 고려했을 때의 결과값이 단순한 장소 예측에 비해 훨씬 어렵다는 점을 고려하면 고무적인 결과라고 할 수 있다.

[표 7] 군집 별 MA, MRE 평균

Metric / 군집	MA	MRE
Cluster 1	0.803	0.462
Cluster 2	0.660	0.581
Cluster 3	0.841	0.471
Cluster 4	0.483	0.682
Cluster 5	0.616	0.576
Cluster 6	0.496	0.662
Cluster 7	0.821	0.497
Cluster 8	0.643	0.577
Cluster 9	0.474	0.691
Cluster 10	0.652	0.558

해당 모델이 실제로 의미 있는 결과를 출력하는지 확인하기 위해 test data의 시퀀스를 이용하여 정답 시퀀스와 출력 시퀀스를 비교한 예시는 [표 8], [표 9]과 같다.

[표 8] 예측 결과 1

input	Rx0031 Rx0031 Rx0031 Rx0031 Sx0156 Sx0156 Sx0156 Sx0156 Sx0156 - - Sx2128 Sx2128 Sx2128 Sx2128
target	Sx2128 Sx2128 Sx2128 Sx2128 Sx2128 Sx2128 - - - -
predict	Sx2128 Sx2128 Sx2128 Sx2128 Sx2128 - - - - -

[표 9] 예측 결과 2

input	Rx0445 Rx0445 Rx0445 - - - - - - Rx0258 Rx0258 - Rx0258 Rx0258 -
target	- Rx0258 Rx0258- - - Rx0258 - - Rx0258
predict	Rx0258 Rx0258 Rx0258 Rx0258 - - - - - -

예측 결과 정답 시퀀스와 예측 시퀀스에 큰 차이가 없음을 확인할 수 있었다. 정확도가 높다는 것은 새로운 경로가 들어왔을 때도 타당한 경로 예측이 가능하다는 것이므로 해당 모델이 유용한 추천을 할 수 있다고 추론 가능하다.

7. 결론 및 향후 연구

이 논문에서는 여행자의 경로 분석을 위해 GPS 데이터로 이루어진 경로를 방문 장소들의 특성을 반영한 문자열로 치환하였으며, 문자열 간의 유사도를 계산한 결과를 이용해 머신러닝 군집화를 진행하였다. 여러 군집화 방법을 이용하여 실험한 결과, 문자열 군집화에서는 레벤슈타인 거리를 유사도 측정 기준으로 사용한 계층적 군집화가 가장 좋은 결과를 보였다. 레벤슈타인 거리와 계층적 군집화를 이용해 군집화를 진행하였고, 모델 학습에 필요한 최소 시퀀스 개수를 기준으로 군집을 분리하였다. 분리한 군집 별로 학습할 딥러닝 모델에는 시계열 데이터 예측에 효과적이며 입력과 출력을 시퀀스로 받을 수 있는 LSTM Encoder-Decoder을 사용하였다. 비슷한 방문 장소를 갖는 여행 경로들이 군집으로 묶였으므로, relaxed LCS를 사용해 사용자가 속하는 군집을 찾은 후 해당 군집의 데이터를 기반으로 학습시킨 딥러닝 모델을 사용해 사용자에게 여행 경로를 추천하였다. 실제 추천에는 정답이 존재하지 않지만, 모델의 정확도를 판단하기 위한 정답과 비교해보았을 때, 모델이 비교적 정확하게 추천하는 것을 알 수 있었다.

본 연구는 여행이 아닌 일상생활에서 수집된 데이터를 사용하였기 때문에 여행 경로 추천이라는 목적에 다소 적합하지 않은 장소도 추천한다는 한계를 가진다. 따라서 실제 여행 데이터를 찾아 변경하면 더욱 여행 경로 추천이라는 목적에 부합하는 모델을 개발할 수 있을 것으로 보인다. 그리고 모델의 예측 결과를 분석했을 때 실제 장소 사이의 거리 관계는 파악하지 못하는 것으로 밝혀졌다. 그러므로 장소 사이의 거리 관계를 파악하는 레이어를 모델에 추가한다면 더욱 정확한 추천을 할 수 있을 것으로 보인다. 하지만 해당 연구의 결과는 실생활에서 사용할 수 있는 서비스로 발전할 가능성이 있으며, 위치 정보를 문자열로 치환해 여행 경로를 추천하는 연구의 초석이 될 수 있다는 점에서 발전 가능성과 의미가 있다고 판단된다.

참고 문헌

[1] 한국관광공사, “여행업의 넥스트레벨”, 「데이터앤 투어리즘」, 3호, p. 16, 2021.

[2] Xing, Lye Guang, et al. “A personalized recommendation framework with user trajectory analysis applied in Location-Based Social Network (LBSN).“ 2017 IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS). IEEE, 2017.

[3] H. Liu, J. Li and J. Peng, “A novel recommendation system for the personalized smart tourism route: Design and implementation,“ 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), 2015, pp. 291-296, doi: 10.1109/ICCI-CC.2015.7259400.

[4] Y. Xu, T. Hu and Y. Li, “A travel route recommendation algorithm with personal preference,“ 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016, pp. 390-396, doi: 10.1109/FSKD.2016.7603205.

[5] Yoon, Hyoseok, et al. “Smart itinerary recommendation based on user-generated GPS trajectories.“ International Conference on Ubiquitous Intelligence and Computing. Springer, Berlin, Heidelberg, 2010.

[6] P.P Gokul, et al. “Sentence similarity detection in Malayalam language using cosine similarity,” 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, doi: 10.1109/RTEICT.2017.8256590

[7] Jilei Tian and Jani Nurminen, “Optimization of text database using hierarchical clustering,” 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, doi: 10.1109/ICASSP.2009.4960572

[8] Chen Qi, Lu Jianfeng and Zhang Hao, “A text mining model based on improved density clustering algorithm,” 2013 IEEE 4th International Conference on Electronics Information and Emergency Communication, 2013, doi: 10.1109/ICEIEC.2013.6835520

[9] Pramod Bide and Rajashree Shedge, “Improved Document Clustering using k-means algorithm,” 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2015, doi: 10.1109/ICECCT.2015.7226065

[10] Graves, Alex. “Long short-term memory.“ Supervised sequence labelling with recurrent neural networks. Springer, Berlin, Heidelberg, 2012. 37-45.

[11] J. Yoon, A. Lee and H. Lee, “Rendezvous: Opportunistic Data Delivery to Mobile Users by UAVs Through Target Trajectory Prediction,“ in IEEE Transactions on Vehicular Technology, vol. 69, no. 2, pp. 2230-2245, Feb. 2020, doi: 10.1109/TVT.2019.2962391.