Encoding Agent Trajectories as Representations with Sequence Transformers

Athanasios Tsiligkaridis, Zhongheng Li, Elizabeth Hou

Systems & Technology Research (STR)

Introduction

Problem motivation:

- Want to learn information about agents given their Pattern of Life (PoL) GPS data obtained over a large timeframe
- Deep learning-based models can be leveraged to learn informative embeddings of trajectories that can further be analyzed to learn relationships between agents

Contributions:

- Present a data discretization technique for the reduction of long and rich PoL data
- Propose a novel transformer-based method for obtaining informative embeddings that can be used to learn relationships between agents
- Showcase experimental results on real and simulated data

Data Preprocessing

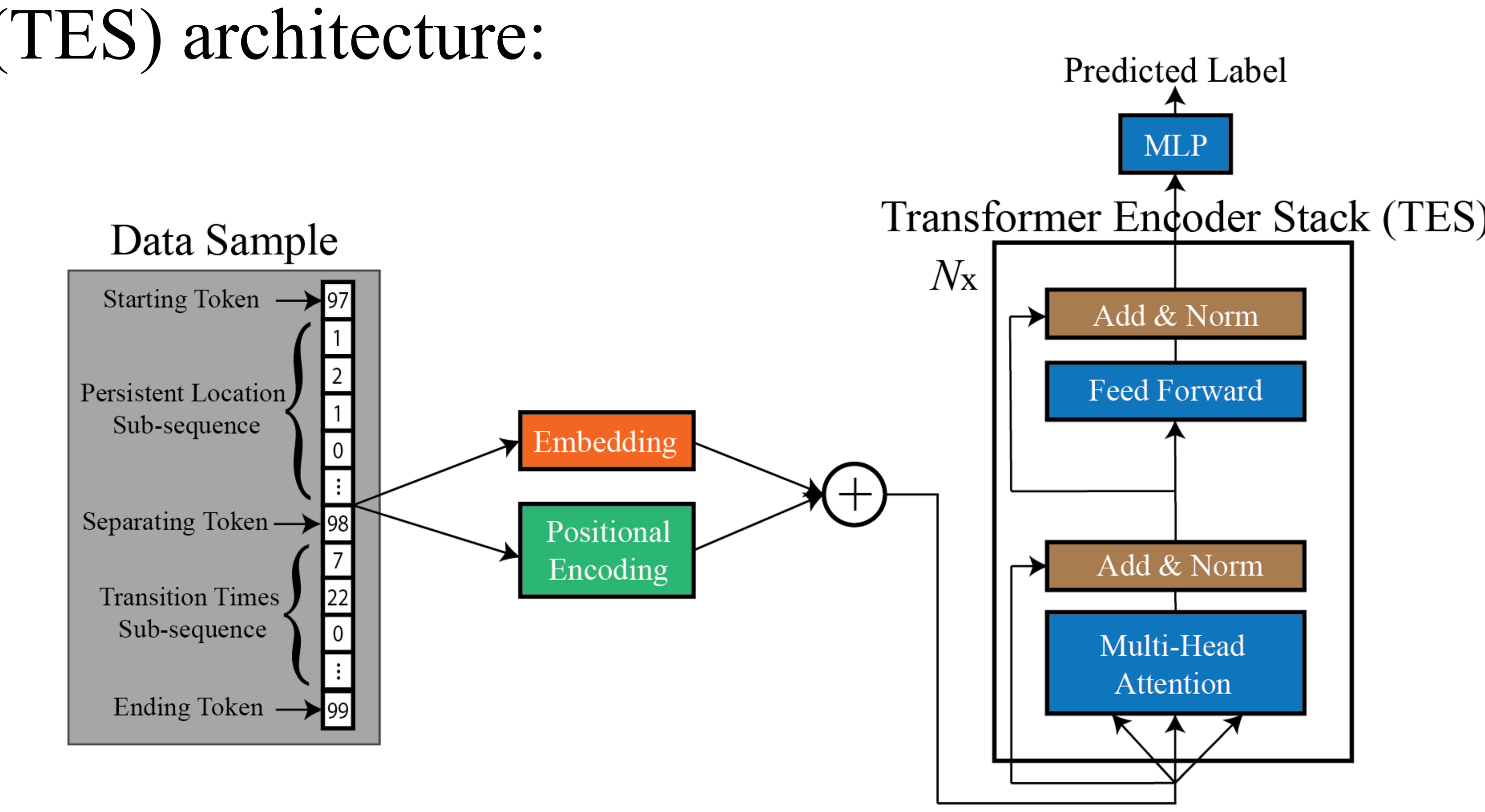
- Raw GPS data cannot directly be used due to:
  - Data scalability issues in high volume settings
  - Discretized input requirement of standard Transformers
- We partition an agent's data into days and discretize the GPS data by forming two daily sub-sequences:
  - 1) Visited Persistent Locations (PL)
  - 2) Transition Times (TT) between the visited PLs
- An agent data point is the concatenation of these two subsequences with *start*, *separating*, and *end* tokens:

$$x_{\text{point}} = [[\text{START}], [PL\ seq.], [\text{SEP}], [TT\ seq.], [\text{END}]]$$

Format of a Data Point

Proposed STARE Model

- We propose a Sequence Transformer for Agent Representation Encodings (*STARE*) model that aims to learn informative embeddings through a supervisory signal that is a classification task
- Classification task data:
  - Input: Daily agent data point sequences
  - Output: Agent labels or subpopulation labels (if existent)
- The STARE model is a Transformer Encoder Stack (TES) architecture:



Proposed STARE Architecture

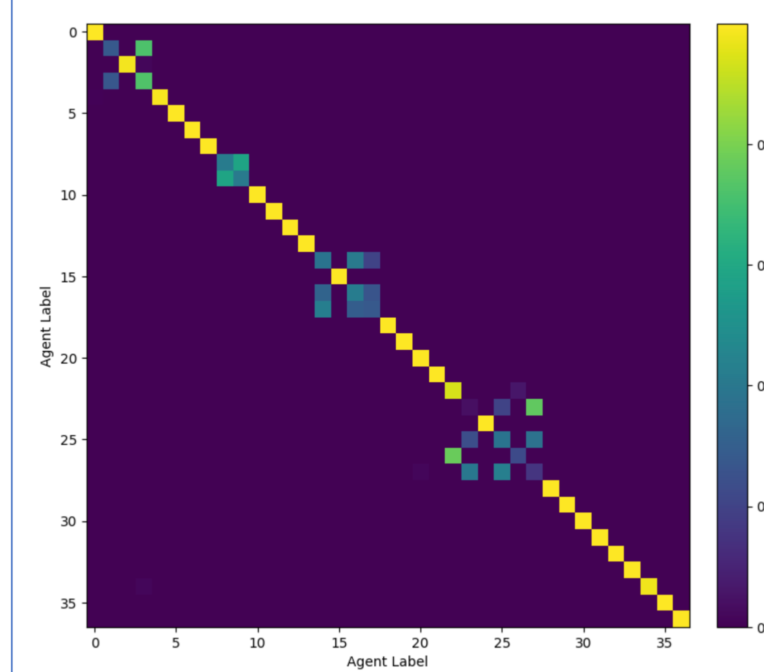
Experimental Results

- 1) Synthetic data:
  - PoL data created using a generative model parameterized by a series of PLs that are associated with each agent.
  - 3 different synthetic sets are created: (S), (M), (L)
  - We train STARE, LSTM, and BiLSTM models using both subpopulation and agent labels
  - Misclassifications can be used to learn subpopulations (if those labels did not exist)
- 2) Ravens data:
  - Real PoL data of 49 ravens from March-July 2018/2019
  - Train models using agent labels; misclassifications can be used to learn subpopulations

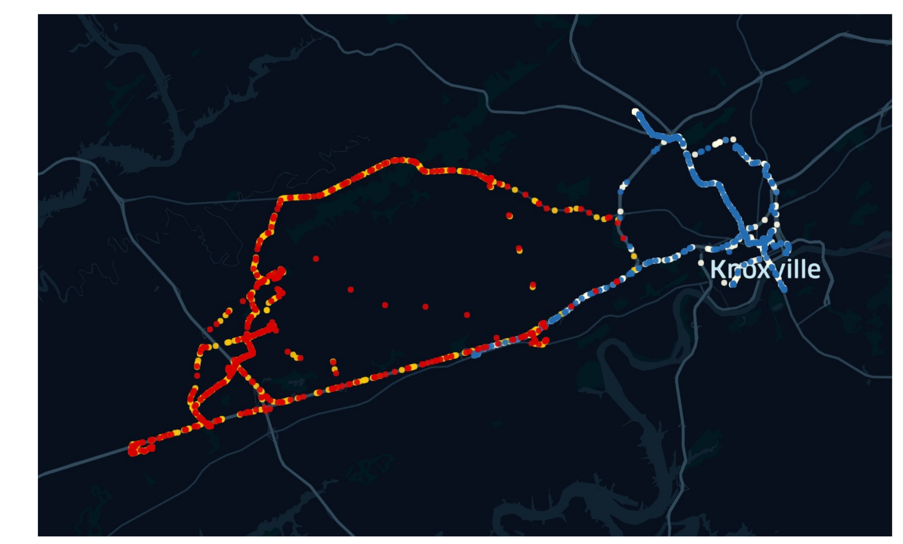
Experimental Results continued…

Dataset	STARE		LSTM		BiLSTM	
	Subpop	Agent	Subpop	Agent	Subpop	Agent
(S): 10 Subpops, 37 Agents	**100%**	**84.0%**	100%	82.6%	100%	80.6%
(M): 20 Subpops, 348 Agents	**100%**	**79.4%**	100%	77.6%	100%	79.4%
(L): 30 Subpops, 1288 Agents	**100%**	**74.2%**	99.7%	73.2%	99.9%	74.1%

Synthetic *- Model Classification Rates*



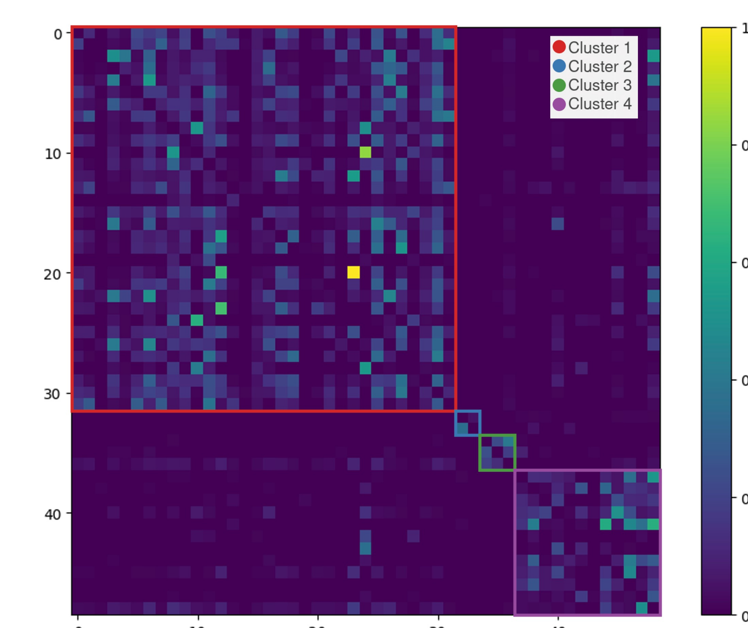
Synthetic *- STARE Predicted Probability Matrix for (S)*



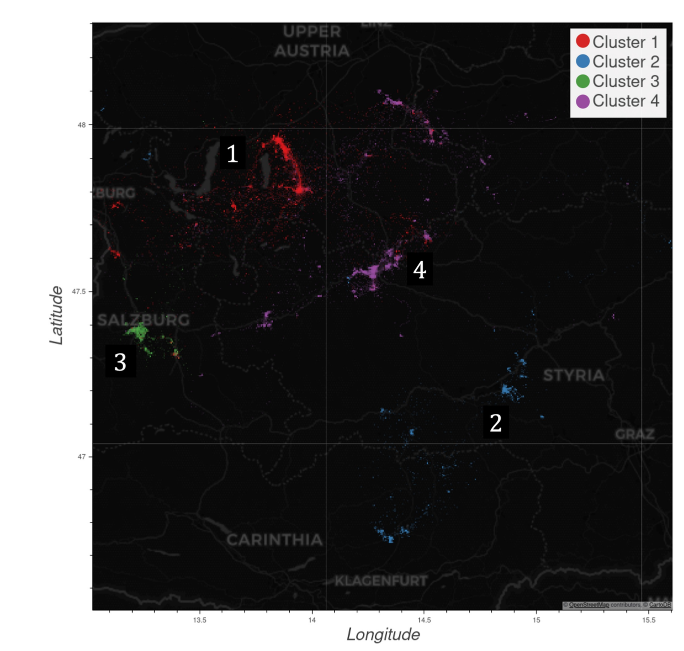
Synthetic *- Similar PoLs between Misclassified Agents*

Dataset	STARE	LSTM	BiLSTM
Ravens [Jain et al. (2022)]	**39.8%**	3.5%	3.9%

Ravens *- Model Classification Rates*



Ravens *– Clustered Predicted Probability Matrix*



Ravens *– Clustered PoL Trajectories*

Future Work

- Assess scalability of our approach by testing on high volume data
- Explore a masked-language training approach as an alternate means of learning latent info in trajectories