Let's Speak Trajectories

Mashaal Musleh*
University of Minnesota, USA

Mohamed F. Mokbel*
University of Minnesota, USA

Sofiane Abbar
Meta Platforms, Inc., UK

ABSTRACT

Trajectory-based applications have acquired significant attention over the past decade with the rising size of trajectory data generated by users. However, building trajectory-based applications is still cumbersome due to the lack of unified frameworks to tackle the underlying trajectory analysis challenges. Inspired by the tremendous success of the BERT deep learning model in solving various NLP tasks, our vision is to have a BERT-like system for a myriad of trajectory analysis operations. We envision that in a few years, we will have such system, where no one needs to worry again about each specific trajectory analysis operation. Whether it is trajectory imputation, similarity, clustering, or whatever, it would be one system that researchers, developers, and practitioners can deploy to get high accuracy for their trajectory operations.

ACM Reference Format:
Mashaal Musleh, Mohamed F. Mokbel, and Sofiane Abbar. 2022. Let's Speak Trajectories. In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3557915.3560972

1 INTRODUCTION

The vast amount of user trajectories being collected nowadays have enabled numerous applications, including transportation (e.g., mapping and routing [6, 16, 39, 59], traffic monitoring and forecasting [18, 21, 36, 48]), location-based service (e.g., recommendations [3, 15, 67]), health (e.g., contact tracing [2, 37, 55]), and urban planning [17, 28, 29], which all have a significant impact on people lives. All such applications have to tackle a wide range of trajectory problems, including trajectory similarity search [5, 10, 26, 31, 41, 56, 58], trajectory imputation [24, 27, 35, 51, 54, 63], classification [38, 44, 45, 49, 65, 66], prediction [13, 19, 32, 50, 57], and simplification [20, 23, 33, 34, 53, 61]. Such problems have been a research focus of the spatial community for years, which has led to numerous and completely diverse solutions for each problem (e.g., see [52, 64] for surveys). Despite the fact that all of these problems deal with the same trajectory data, each of the proposed solutions is entirely designed to solve one problem of interest. This makes it hard to have a unified efficient and practical framework that is capable of supporting most (if not all) trajectory problems.

*The work of these authors is partially supported by the National Science Foundation (NSF), USA, under grants IIS-1907855 and IIS-2203553.

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11...$15.00
https://doi.org/10.1145/3557915.3560972

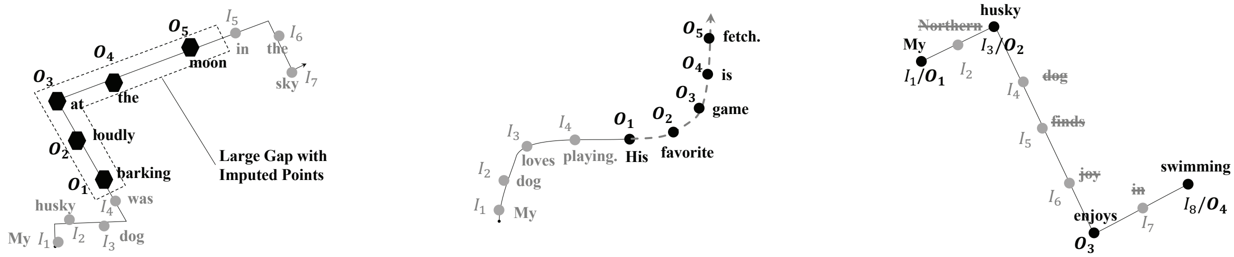
Meanwhile, the research landscape of Natural Language Processing (NLP) was in a similar situation with decades of research in pushing the accuracy and efficiency of various NLP tasks, e.g., text similarity and sentiment analysis. This has led to a myriad of different solutions for each of these problems, even though all tasks are for the same textual data. Most recently, in 2018, the BERT deep learning model [11] (Bidirectional Encoder Representations from Transformers), is proposed by Google to act as a unified solution infrastructure for a wide variety of NLP tasks. BERT, at its core, is equipped with the necessary NLP infrastructure to solve various NLP tasks, which only needs to be externally tuned with minimal overhead for each task. Examples of NLP tasks that used the BERT model include sentiment analysis [12], question answering [8], spell checking [60], text classification [9], text generation[7], text summarization [42], among others [25, 40]. BERT has also been used for similar problems with respect to speech processing, where the words are spoken instead of written [22, 46]. As a testimony to the importance and ubiquity of BERT to NLP research, the main BERT paper [11] has been cited 40+K times within four years.

Our vision is to have a BERT-like model that will magically deal with almost all trajectory analysis techniques. Once we have such model, various trajectory analysis ideas will be just about how to tune that model one way or another to support the required analysis. Such vision will lead to a long-waited-for full-fledged trajectory data management system that does not only store and index trajectory data, but natively support all its data analysis needs.

Our vision is grounded by the fact that we can actually think of trajectories as statements. A statement is composed of a set of words drawn from a set of limited words (language), while a trajectory is represented by a set of GPS points, which are also limited. Section 2 elaborates more on our vision's ground. Section 3 shows how to apply BERT directly to trajectory analysis. Customizing trajectory data and BERT to be used together towards higher accuracy is discussed in Section 4. Finally, Section 5 outlines the full vision.

2 TRAJECTORIES ARE STATEMENTS

The main idea behind our vision is to deal with trajectories as statements because of their similarities, hence a BERT-like model can be applied to trajectories. Such similarities include: (a) A statement is composed of an ordered set of words drawn from a finite pool of words per the underlying language. Similarly, a trajectory is composed of an ordered set of GPS points drawn from a finite pool of possible points per the underlying space, (b) Words in a statement are semantically related where random words cannot make a statement. Similarly, points in a trajectory are spatially and temporally related where random points cannot make a trajectory, (c) Statements are constrained by rules imposed by the language grammar. Similarly, trajectories are constrained by rules imposed by the road network and physical constraints, and (d) The choice of words in a statement depends on the user or topic writing style, where some



(a) Trajectory Imputation (b) Trajectory Prediction (c) Trajectory Simplification

Figure 1: Examples of Trajectory Data Analysis Problems Supported By BERT Model

users/topics may use stronger words than others. Similarly, the choice of points metadata (e.g., speed) in a trajectory depends on the user or driving modality (e.g., bus or motorbikes) style, where some users/modalities may use different speed patterns than others.

Few earlier works have recognized some of such similarities between trajectories and word sequences, and have used this either for trajectory representation [14, 68] or for solving one specific trajectory problem [26, 30]. Most of such works were even before the idea of the BERT model was proposed. Our vision is to go beyond this, and have a BERT-like model that is not specific to one trajectory problem. Instead, it will act as a Swiss army knife that supports a myriad of diverse trajectory operations.

3 BERT FOR TRAJECTORY ANALYSIS

This section discusses the first step towards our vision, which is using BERT for various trajectory analyses, namely, imputation, prediction, classification, simplification, and similarity.

Trajectory Imputation. As trajectory data is usually sparse with some significant spatial gaps between consecutive points, trajectory imputation is the process of densifying such sparse trajectories by inferring additional points that would fill the gaps [24, 27, 35, 51, 54, 63]. This is an important and crucial preprocessing step for a myriad of trajectory applications that need dense trajectories. Trajectory imputation can be seen as analogous to the *"finding the missing word"* problem in NLP, which is usually solved using a BERT model. Given a statement like, "My husky dog was — loudly at the moon", where "—" represents a missing word (due to speech recognition, translation, or typo), BERT finds out that the missing word is "barking". To do so, a BERT model is first trained by hundreds of thousands of true statements that will make it understand the context and accurately find out the missing word. Hence, one way to solve the trajectory imputation problem is to train a BERT model by a large number of trajectories, and then use it to find out a missing point between two consecutive points. Figure 1(a) gives an example of an English statement with some missing words, laid out on a trajectory (I_1 to I_7) with missing points. Both the statement and trajectory are depicted in gray. Then, the words/points in black are the imputed ones when applying BERT.

Trajectory Prediction. Trajectory prediction is the task of predicting the next few points of the current trajectory. Due to its importance in trajectory and traffic analysis, over the last decade, significant efforts have been dedicated to both short-term (next few minutes) [13] and long-term (next 20-30 minutes) [19, 57] trajectory predictions. Trajectory prediction can be seen as analogous to the *"next sentence prediction"* problem in NLP, which is usually solved using a BERT model. Given a sentence, a BERT model can be used to find the most likely sentence that naturally follows the given one. To do so, BERT is trained and fine-tuned using pairs of <*input*, *target*> sentences that will make it understand a target statement given an input one. Hence, one way to solve the trajectory prediction problem is to train a BERT model with pairs of sequence trajectories that can be obtained by splitting real trajectories into two parts, *input* and *target*. Then, we can use it to predict the next trajectory (next few points) for a given one. Figure 1(b) gives an example of input sentence and trajectory (shown in gray), where BERT predicted their next few words and points (depicted in black).

Trajectory Classification. Trajectory classification is the process of associating a trajectory with one class from a predefined set of classes, e.g., associating a trajectory with its modality that could be either biking, walking, or driving [38, 44, 45, 49, 65, 66], which is very crucial to traffic analysis. Trajectory classification can be seen as analogous to the *"text classification"* problem in NLP, which is usually solved using a BERT model. Given a social media post (e.g., tweet) and a set of categories (e.g., sports, politics, and technology), a BERT model can classify the tweet into one of the given categories. To do so, BERT is first trained on a large number of *unlabeled* sentences to learn about words in general. Then, it is fine-tuned using a relatively smaller *labeled* sentences as <*tweet*, *category*>. Hence, one way to solve the trajectory classification problem is to train a BERT model using unlabeled trajectories and fine-tune it using labeled trajectories of the form <*trajectory*, *modality*>, then use the trained model to find the modality for any given new trajectory.

Trajectory Simplification. Trajectory simplification, sometimes seen as the opposite of trajectory imputation, is the task of reducing the number of trajectory GPS points while preserving their essential information [20, 23, 33, 34, 53, 61]. It is used to significantly reduce the cost of query processing and data transmissions of complex trajectories. Trajectory simplification can be seen as analogous to the *"text summarization"* problem in NLP, which is usually solved using a BERT model. Given a document of words, BERT can be used to summarize the document by a short description to be used for newsletters, video descriptions, or brief highlights. To do so, similar to the case of text classification, BERT would be trained on documents, then, on a pair of documents and their simplified summaries. Hence, one way to solve the trajectory simplification problem is to train and fine-tune a BERT model using pairs of <*raw trajectory*, *simplified trajectory*>, and then use it for new trajectories. Figure 1(c) gives an example of a trajectory/statement composed of nine points/words (shown in gray), where BERT is applied to come up with a simplified/summarized trajectory/statement composed of four points/words (depicted in black).

Trajectory Similarity. Trajectory similarity is the process of computing a similarity score between two trajectories based on their sampled GPS points [10, 26, 31, 56]. This is a cornerstone in various trajectory analysis modules, including clustering, outlier detection, and map matching. Trajectory similarity can be seen as analogous to the *"text similarity"* problem in NLP, which is solved using BERT model. Given two statements, BERT represents them as two vectors of the same size, regardless of the number of words in each statement. The vectors then go through a simple mathematical operation (e.g., cosine similarity or Euclidean distance), to measure the vectors (and hence the statements) similarity. To do so, BERT is first trained on large datasets of statements so it can represent similar statements by similar vectors. Hence, one way to solve the trajectory similarity problem is to train BERT on a large trajectory dataset to represent similar trajectories by similar vectors.

4 CUSTOMIZING BERT FOR TRAJECTORIES

This section advocates for going beyond the idea of applying BERT as is for trajectory analysis (Section 3) to actually customizing it for a better accuracy. This mainly addresses the following challenges that came out from a direct deployment of BERT to trajectories:

Ratio of training datasets to possible words. BERT is used for languages, where the number of possible words is of limited size, with an abundance of available data. BERT was actually trained on ~3.3B word corpus (2.5B of them are from Wikipedia and 800M from Books Corpus [69]) composed of ~30K distinct words [11]. Meanwhile, trajectory data have a significantly larger number of distinct words/points with a smaller number of corpus words/points. For example, a trajectory dataset from Oregon State, obtained from UCR STAR [47] has ~1.3M distinct GPS points (about three orders of magnitude *more* than English) with ~1.75M total points (about three orders of magnitude *less* than English). With these numbers, each English word appears ~100K times in the BERT training set, while each GPS point appears only once in the trajectory training set, which gives five orders of magnitude more advantage to words than GPS points. To overcome this challenge, we can exploit two directions: (a) partition the space into a set of fine-grained hexagons, using Uber's H3 Hexagonal Hierarchical Spatial Index [4]. Then, all points within the same hexagon will be assigned to the same GPS value, which is the hexagon centroid. This brings the number of possible words/points down to 18K (each will appear ~100 times in the corpus), while accuracy is still preserved due to the fine-grained nature of the hexagons. (b) use our available real trajectory data to generate additional trajectories [43, 62] and enrich our corpus.

Noisy data. Trajectory data is more subject to noise than language data. While a noise in a language would take place as typos or grammatical errors, noise in trajectories is inherent with inaccurate, erroneous, or even missing GPS points. For example, in the trajectory imputation example of Figure 1(a), it is common to insert multiple points between every two consecutive points, making the number of imputed points even more than the number of real ones. This is different from finding the missing word, where the number of missed words is much less than the number of available words. Therefore, we need to customize BERT to support such noise. One way to do so for trajectory imputation in Figure 1(a) is to call BERT iteratively. The first call predicts point O_3, which lies between I_4 and I_5. Then, we call BERT twice to get a point O_1 between O_3 and I_4 and point O_4 between O_3 and I_5. We repeat to get O_2 and O_5.

Long and unrelated consecutive statements. Statements are usually composed of few words, while paragraphs and documents are composed of a sequence of related statements. Meanwhile, trajectories may include hundreds of points, and subsequent trajectories may not be very related, e.g., a series of taxi trips. This may make it hard to use BERT for some trajectory operations, including prediction (Figure 1(b)), that rely on the relation between subsequent statements. One way to overcome this is to split long trajectories into a set of shorter subtrajectories. In that case, a trajectory would actually act as a paragraph rather than a statement.

Spatial and temporal constraints. Trajectories have their own spatial and temporal constraints that may guide some of the analysis operations. For example, in the trajectory simplification example of Figure 1(c), we would need to ensure that the first and last points appear in the simplified trajectory to maintain the trajectory spatial and temporal properties. Such properties would not be needed in case of using BERT in text summarization. To address this issue, we would need to manually set aside the first and last points, then, run BERT on the remaining points.

5 THE VISION: A BERT-LIKE SYSTEM FOR TRAJECTORIES

To evaluate the potential of our vision, we ran an initial experiment of using BERT model for trajectory imputation of the GISCUP'17 dataset [1], which include 5M GPS points in San Francisco, grouped into 18K hexagons with 66 meters edge length for each. We train BERT model on 80% of the points and keep the remaining 20% for testing, in which we down-sample the trajectories by dropping three-quarters of the points of each trajectory and then run BERT to fill the gaps by imputing the missing points. Since we know the ground truth trajectories, we measure the error by computing the shortest Euclidean distance between the imputed points and the actual trajectories, which is similar to what other studies have used [27]. The mean and median distances were 37.9 and 38.9 meters, which represent a promising accuracy. The results obtained from our initial set of experiments show that the ideas we have in Sections 3 and 4 are paving the way for our vision.

Our vision is that the spatial community would be working together towards a full-fledged BERT-like system for a myriad of trajectory analysis operations. This does not have to be building a new system from scratch. Instead, we need to change the core of the BERT itself to make it deal with spatial data in general and trajectories in particular as first-class citizens. BERT would need to understand that spatial data is special, and support its characteristics. *We envision that in a few years, we will have such system, where no one needs to worry again about each specific trajectory analysis operation. Whether it is trajectory imputation, similarly, clustering, or whatever, it would be one system that researchers, developers, and practitioners can deploy to get high accuracy for their operations.* The system would always be extensible in a way that can accommodate new operations contributed by the community at large.

REFERENCES

[1] ACM SIGSPATIAL CUP 2017. http://sigspatial2017.sigspatial.org/giscup2017.

[2] R. Alseghayer. Racoon: Rapid Contact Tracing of Moving Objects Using Smart Indexes. In *MDM*, 2021.

[3] I. R. Brilhante, J. A. F. de Macêdo, F. M. Nardini, R. Perego, and C. Renso. Planning Sightseeing Tours Using Crowdsensed Trajectories. *ACM SIGSPATIAL Special*, 7(1), 2015.

[4] I. Brodsky. H3: Uber's Hexagonal Hierarchical Spatial Index. https://eng.uber.com/h3/.

[5] L. Chen, S. Shang, C. S. Jensen, B. Yao, and P. Kalnis. Parallel Semantic Trajectory Similarity Join. In *ICDE*, 2020.

[6] L. Chen, S. Shang, C. S. Jensen, B. Yao, Z. Zhang, and L. Shao. Effective and Efficient Reuse of Past Travel Behavior for Route Recommendation. In *KDD*, 2019.

[7] Y. Chen, Z. Gan, Y. Cheng, J. Liu, and J. Liu. Distilling Knowledge Learned in BERT for Text Generation. In *ACL*, 2020.

[8] Y. Chen and F. H. Zulkernine. BIRD-QA: A BERT-based Information Retrieval Approach to Domain Specific Question Answering. In *IEEE Big Data*, 2021.

[9] D. Croce, G. Castellucci, and R. Basili. GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples. In *ACL*, 2020.

[10] R. S. de Sousa, A. Boukerche, and A. A. F. Loureiro. Vehicle Trajectory Similarity: Models, Methods, and Applications. *ACM Comput. Surv.*, 53(5), 2020.

[11] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018.

[12] C. Du, H. Sun, J. Wang, Q. Qi, and J. Liao. Adversarial and Domain-Aware BERT for Cross-Domain Sentiment Analysis. In *ACL*, 2020.

[13] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *WWW*, 2018.

[14] T. Fu and W. Lee. Trembr: Exploring Road Networks for Trajectory Representation Learning. *ACM Trans. Intell. Syst. Technol.*, 11(1), 2020.

[15] Q. Gao, G. Trajcevski, F. Zhou, K. Zhang, T. Zhong, and F. Zhang. DeepTrip: Adversarially Understanding Human Mobility for Trip Recommendation. In *SIGSPATIAL*, 2019.

[16] S. He, F. Bastani, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, and S. Madden. Roadrunner: Improving the Precision of Road Network Inference from GPS Trajectories. In *SIGSPATIAL*, 2018.

[17] T. He, J. Bao, S. Ruan, R. Li, Y. Li, H. He, and Y. Zheng. Interactive Bike Lane Planning Using Sharing Bikes' Trajectories. *TKDE*, 32(8), 2020.

[18] B. Hossain, K. A. Adnan, M. F. Rabbi, and M. E. Ali. Modelling Road Traffic Congestion from Trajectories. In *DSIT*, 2020.

[19] A. Karatzoglou, A. Jablonski, and M. Beigl. A Seq2Seq Learning Approach for Modeling Semantic Trajectories and Predicting the Next Location. In *SIGSPATIAL*, 2018.

[20] P. Katsikouli, R. Sarkar, and J. Gao. Persistence Based Online Signal and Trajectory Simplification for Mobile Devices. In *SIGSPATIAL*, 2014.

[21] B. B. Krogh, O. Andersen, E. Lewis-Kelham, N. Pelekis, Y. Theodoridis, and K. Torp. Trajectory Based Traffic Analysis. In *SIGSPATIAL*, 2013.

[22] C. Kuo, S. Luo, and K. Chen. An Audio-Enriched BERT-Based Framework for Spoken Multiple-Choice Question Answering. In *Interspeech*, 2020.

[23] M. Laass, M. Kiermeier, and M. Werner. Improving Persistence Based Trajectory Simplification. In *MDM*, 2021.

[24] B. Li, Z. Cai, M. Kang, S. Su, S. Zhang, L. Jiang, and Y. Ge. A Trajectory Restoration Algorithm for Low-sampling-rate Floating Car Data and Complex Urban Road Networks. *International Journal of GIS*, 35(4), 2021.

[25] B. Li, Y. Miao, Y. Wang, Y. Sun, and W. Wang. Improving the Efficiency and Effectiveness for BERT-based Entity Resolution. In *AAAI*, 2021.

[26] X. Li, K. Zhao, G. Cong, C. S. Jensen, and W. Wei. Deep Representation Learning for Trajectory Similarity Computation. In *ICDE*, 2018.

[27] Y. Li, Y. Li, D. Gunopulos, and L. J. Guibas. Knowledge-based Trajectory Completion from Sparse GPS Samples. In *SIGSPATIAL*, 2016.

[28] Y. Li, J. Luo, C. Chow, K. Chan, Y. Ding, and F. Zhang. Growing The Charging Station Network For Electric Vehicles With Trajectory Data Analytics. In *ICDE*, 2015.

[29] F. Lin and H. Hsieh. An Intelligent And Interactive Route Planning Maker For Deploying New Transportation Services. In *SIGSPATIAL*, 2018.

[30] Y. Lin, H. Wan, S. Guo, and Y. Lin. Pre-training Context and Time Aware Location Embeddings from Spatial-Temporal Trajectories for User Next Location Prediction. In *AAAI*, 2021.

[31] A. Liu, Y. Zhang, X. Zhang, G. Liu, Y. Zhang, Z. Li, L. Zhao, Q. Li, and X. Zhou. Representation Learning With Multi-Level Attention for Activity Trajectory Similarity Computation. *TKDE*, 34(5), 2022.

[32] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. In *AAAI*, 2016.

[33] C. Long, R. C. Wong, and H. V. Jagadish. Direction-Preserving Trajectory Simplification. *PVLDB*, 6(10), 2013.

[34] C. Long, R. C. Wong, and H. V. Jagadish. Trajectory Simplification: On Minimizing the Direction-based Error. *PVLDB*, 8(1), 2014.

[35] J. A. Long. Kinematic Interpolation of Movement Data. *International Journal of Geographical Information Science*, 30(5), 2016.

[36] C. Meng, X. Yi, L. Su, J. Gao, and Y. Zheng. City-wide Traffic Volume Inference with Loop Detector Data and Taxi Trajectories. In *SIGSPATIAL*, 2017.

[37] M. F. Mokbel, S. Abbar, and R. Stanojevic. Contact Tracing: Beyond the Apps. *ACM SIGSPATIAL Special*, 12(2), 2020.

[38] D. Montoya, S. Abiteboul, and P. Senellart. Hup-Me: Inferring and Reconciling a Timeline of User Activity from Rich Smartphone Data. In *SIGSPATIAL*, 2015.

[39] M. Musleh, S. Abbar, R. Stanojevic, and M. F. Mokbel. QARTA: An ML-based System for Accurate Map Services. *PVLDB*, 14(11), 2021.

[40] N. Peinelt, D. Nguyen, and M. Liakata. tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection. In *ACL*, 2020.

[41] L. M. Petry, C. A. Ferrero, L. O. Alvares, C. Renso, and V. Bogorny. Towards Semantic-Aware Multiple-Aspect Trajectory Similarity Measuring. *Transactions in GIS*, 23(5), 2019.

[42] R. A. Pugoy and H. Kao. Unsupervised Extractive Summarization-Based Representations for Accurate and Explainable Collaborative Filtering. In *ACL*, 2021.

[43] J. Rao, S. Gao, and X. Zhu. VTSV: A Privacy-Preserving Vehicle Trajectory Simulation and Visualization Platform Using Deep Reinforcement Learning. In *ACM SIGSPATIAL Workshop on AI for Geo Knowledge Discovery, GeoAI*, 2021.

[44] S. Reddy, M. Y. Mun, J. Burke, D. Estrin, M. H. Hansen, and M. B. Srivastava. Using Mobile Phones to Determine Transportation Modes. *ACM Transactions on Sensors Networks*, 6(2), 2010.

[45] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu. Transportation Mode Detection Using Mobile Phones and GIS Information. In *SIGSPATIAL*, 2011.

[46] K. Tsunematsu, J. Effendi, S. Sakti, and S. Nakamura. Neural Speech Completion. In *Interspeech*, 2020.

[47] UCR STAR: The UCR Spatio-temporal Active Repository. https://star.cs.ucr.edu.

[48] A. Vahedian, X. Zhou, L. Tong, Y. Li, and J. Luo. Forecasting Gathering Events through Continuous Destination Prediction on Big Trajectory Data. In *SIGSPATIAL*, 2017.

[49] B. Wang, Y. Wang, K. Qin, and Q. Xia. Detecting transportation modes based on lightgbm classifier from GPS trajectory data. In *Geoinformatics*, 2018.

[50] J. Wang, J. Jiang, W. Jiang, C. Li, and W. X. Zhao. LibCity: An Open Library for Traffic Prediction. In *SIGSPATIAL*, 2021.

[51] J. Wang, N. Wu, X. Lu, W. X. Zhao, and K. Feng. Deep Trajectory Recovery with Fine-Grained Calibration using Kalman Filter. *TKDE*, 33(3), 2021.

[52] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong. A Survey on Trajectory Data Management, Analytics, and Learning. *ACM Comput. Surv.*, 54(2), 2021.

[53] Z. Wang, C. Long, and G. Cong. Trajectory Simplification with Reinforcement Learning. In *ICDE*, 2021.

[54] H. Wu, J. Mao, W. Sun, B. Zheng, H. Zhang, Z. Chen, and W. Wang. Probabilistic Robust Route Recovery with Spatio-Temporal Dynamics. In *KDD*, 2016.

[55] L. Xiong, C. Shahabi, Y. Da, R. Ahuja, V. Hertzberg, L. Waller, X. Jiang, and A. Franklin. REACT: Real-Time Contact Tracing and Risk Monitoring Using Privacy-enhanced Mobile Tracking. *ACM SIGSPATIAL Special*, 12(2), 2020.

[56] P. Yang, H. Wang, Y. Zhang, L. Qin, W. Zhang, and X. Lin. T3S: Effective Representation Learning for Trajectory Similarity Computation. In *ICDE*, 2021.

[57] J. J. Ying, W. Lee, T. Weng, and V. S. Tseng. Semantic Trajectory Mining for Location Prediction. In *SIGSPATIAL*, 2011.

[58] H. Yuan and G. Li. Distributed In-memory Trajectory Similarity Search and Join on Road Network. In *ICDE*, 2019.

[59] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-Drive: Driving Directions Based on Taxi Trajectories. In *SIGSPATIAL*, 2010.

[60] S. Zhang, H. Huang, J. Liu, and H. Li. Spelling Error Correction with Soft-Masked BERT. In *ACL*, 2020.

[61] Y. Zhao, S. Shang, Y. Wang, B. Zheng, Q. V. H. Nguyen, and K. Zheng. REST: A Reference-based Framework for Spatio-temporal Trajectory Compression. In *KDD*, 2018.

[62] G. Zheng, H. Liu, K. Xu, and Z. Li. Learning to Simulate Vehicle Trajectories from Demonstrations. In *ICDE*, 2020.

[63] K. Zheng, Y. Zheng, X. Xie, and X. Zhou. Reducing Uncertainty of Low-Sampling-Rate Trajectories. In *ICDE*, 2012.

[64] Y. Zheng. Trajectory Data Mining: An Overview. *ACM TIST*, 6(3), 2015.

[65] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W. Ma. Understanding Transportation Modes Based on GPS Data for Web Applications. *ACM Tran. on Web*, 4(1), 2010.

[66] Y. Zheng, L. Liu, L. Wang, and X. Xie. Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. In *WWW*, 2008.

[67] F. Zhou, H. Wu, G. Trajcevski, A. A. Khokhar, and K. Zhang. Semi-supervised Trajectory Understanding with POI Attention for End-to-End Trip Recommendation. *TSAS*, 6(2), 2020.

[68] N. Zhou, W. X. Zhao, X. Zhang, J. Wen, and S. Wang. A General Multi-Context Embedding Model for Mining Human Trajectory Data. *TKDE*, 28(8), 2016.

[69] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *ICCV*, 2015.