REPRESENTATION LEARNING OF DAILY MOVEMENT DATA USING TEXT ENCODERS

Alexander Capstick
Imperial College London
alexander.capstick19@imperial.ac.uk

Tianyu Cui
Imperial College London
t.cui23@imperial.ac.uk

Yu Chen
Imperial College London
yu.chen@imperial.ac.uk

Payam Barnaghi
Imperial College London
p.barnaghi@imperial.ac.uk

ABSTRACT

Time-series representation learning is a key area of research for remote healthcare monitoring applications. In this work, we focus on a dataset of recordings of in-home activity from people living with Dementia. We design a representation learning method based on converting activity to text strings that can be encoded using a language model fine-tuned to transform data from the same participants within a 30-day window to similar embeddings in the vector space. This allows for clustering and vector searching over participants and days, and the identification of activity deviations to aid with personalised delivery of care.

1 INTRODUCTION

In remote healthcare monitoring applications, time-series data is continuously collected using wearables or Internet of Things devices. This scale of data collection necessitates machine learning techniques for analysis and delivery of care. Additionally, time-series data is challenging to label (Ching et al., 2018), requiring representation learning through self-supervised, semi-supervised, or unsupervised methods to extract insights. This transforms data into useful encodings for downstream tasks and analysis and can often be performed on datasets with few or no labels (Franceschi et al., 2019; Eldele et al., 2021; Zerveas et al., 2021; Yang & Hong, 2022).

Considering a neural network as a feature extractor followed by a predictor (Murphy, 2022), supervised learning becomes joint representation learning and predictive modelling. This fact inspires most of the self-supervised or semi-supervised representation learning literature to date since predictive tasks can be defined using the time-series itself, using techniques such as data masking, shuffling, or contrastive learning (Balestriero et al., 2023).

In this work, we focus on collections of time-series containing irregularly appearing discrete values. These could be time-series of electronic healthcare records or remote monitoring activity. We present initial results of a representation learning method that benefits from advancements in large language modelling, text encoding, and vector search to aid in clustering, finding similar clinical cases, identification of changes in data patterns, and to support personalised care delivery. In this way, our method utilises a language model's pre-trained representations of different in-home locations which we hypothesise allow for a more meaningful encoding of data.

1.1 BACKGROUND

In representation learning, the goal is to encode raw data as vectors that are useful for further analysis, usually by harnessing deep learning. The literature on this topic is broad, including theoretical techniques concerned with few-shot prediction (Wang et al., 2020b), continual learning (Hadsell et al., 2020), multimodal learning (Baltrušaitis et al., 2019; Xu et al., 2023), and applications to healthcare (Alanazi, 2022), language modelling (Vaswani et al., 2017), and image generation (Radford et al., 2021). In many ways, all neural networks utilise representation learning.

For time-series data, varied methods of learning data representations have been proposed. For example, Franceschi et al. (2019), inspired by Mikolov et al. (2013), train an encoder based model using a triplet loss by selecting random subsets of the same time-series recording as similar, and random subsets of other recordings as dissimilar; Eldele et al. (2021) propose training models to maximise the similarity between embeddings of weak and strong augmentations of data, which they show are useful for downstream tasks; and Zerveas et al. (2021) use a transformer-encoder architecture (Vaswani et al., 2017) and a linear projection over a time-series to learn representations by masking inputs and predicting the hidden values.

Large language models (LLMs), based on self-attention (Vaswani et al., 2017), are examples of time-series representation learning models that have found significant success due to their scalability and parallel design (Lin et al., 2022). Fine-tuning these models have allowed for the transfer of learning to specialised tasks initially outside of their intended uses (Brown et al., 2020; Wei et al., 2021; Agrawal et al., 2022; Ding et al., 2023). Additionally, since language models are trained on billions of text documents from varied domains, they have been shown to encode relational knowledge (Petroni et al., 2019) of text without being explicitly trained for these tasks.

Progress has been made in adapting LLMs for use outside of language modelling. Gruver et al. (2023) experiment with GPT-3 (Brown et al., 2020) and LLaMA-2 (Touvron et al., 2023), finding they have a surprisingly good ability to zero-shot forecast time-series encoded as strings of numerical digits. Moreover, Jin et al. (2024) use prompt engineering and reprogramming to adapt LLMs for classification tasks. In both, the authors test numerical time-series data focused on predictive tasks.

1.2 Our contribution

Focusing on in-home activity data collected passively using low-cost sensors from the homes of people with Dementia, we learn representations of time-series for tasks such as activity clustering, searching for similar participants and patterns, and measuring deviations in activity. This technology would aid clinical teams in personalised care delivery planning or assessing risks to morbidities by referencing similar participants with known health trajectories and care requirements. Appendix A.1 discusses the availability of code and data for the reproduction of this work.

2 Methods

2.1 Language Modelling

Following Gruver et al. (2023), language models are typically auto-regressive models trained on a collection of sequences, $\mathcal{U} = \{U_1, ..., U_N\}$, where $U_i = (u_1, ..., u_{n_i})$ and each u_i is a token representing a single piece of text from a vocabulary $\mathcal{V}$. Since each predicted token is dependent on all previous tokens in the sequence, for a model parameterised by θ, we have $\log p(U_i|\theta) = \sum_{j=1}^{n_i} \log p(u_j|u_{0:j-1}, \theta)$. The model parameters, θ are learnt by maximising this likelihood over the entire dataset (i.e, jointly over $\mathcal{U}$). The tokeniser component of a language model encodes an input string into a sequence of tokens from $\mathcal{V}$.

The transformer architecture (Vaswani et al., 2017), including an encoder and decoder, is a popular choice for language modelling. BERT (Devlin et al., 2019), based on the transformer encoder architecture only, is trained with $\log p(U_i|\theta) = \sum_{j=1}^{n_i} \log p(u_j|u_{-j}, \theta)$ (with u_{-j} as all tokens except u_j) and designed to be fine-tuned (Murphy, 2022). This model outputs an embedding for each input token, contextualising them within a sequence.

Reimers & Gurevych (2019) apply mean-pooling to token embeddings to produce sentence embeddings; a single vector (of length given by the token embedding length) representing an entire sequence (with a maximum length defined by the BERT architecture). In parallel, Wang et al. (2020a) present a method for distilling a BERT-like model into smaller models useful for fine-tuning. The authors make this distilled model, MiniLM (and subsequently MiniLM V2) publicly available. These works lead to a sentence embedding model based on a fine-tuned MiniLM V2 model (Reimers & Gurevych, 2019; Wang et al., 2020a; Wolf et al., 2020) referred to here as "SE-MiniLM".

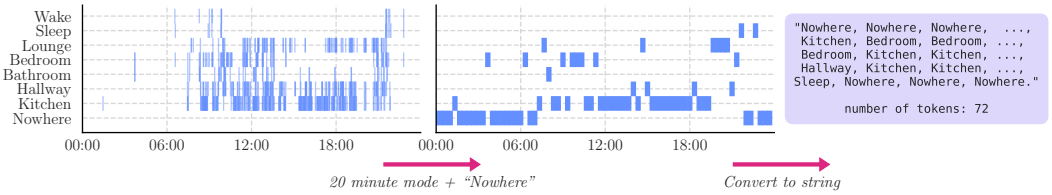

Figure 1: **Preprocessing of data.** For a single participant, we present their data collected over a single day. The left graph shows the raw measurements; the central graph shows the measurements after the mode of each 20 minute window is taken and a token for no activity is assigned ("Nowhere"); and the right graph shows the day as a single text string that can be interpreted by a language model.

Sequence (or sentence) embedding models can be fine-tuned using a triplet loss (Hoffer & Ailon, 2015), measuring the cosine similarity between the embeddings of an anchor sequence and a similar and dissimilar sequence, such that similar sequences have similar embeddings.

Semantic vector search uses the learnt embeddings of objects to find similar objects. This can be done using various similarity measures. In this work, we use cosine similarity.

2.2 The Dataset

We have access to a dataset collected from 134 people living with Dementia, containing in-home data recorded between 2021-07-01 and 2024-01-30. Each participant in this study has passive infrared (PIR) sensors positioned around their house, allowing us to collect for each day, the times that locations in the home were visited by occupants. Additionally, the dataset contains data collected from sleep mats that record the time the person living with Dementia enters or leaves their bed. On average, the number of days recorded per participant is 492 (a minimum of 5, median of 513, and maximum of 943) and in total, this dataset contains 65,962 recorded days. Further information is given in Appendix A.2.

2.3 Our Method

This work presents initial results from a system capable of encoding days of recorded activity data into vectors that are useful for care delivery. We transform each participant's daily time-series of events into strings of text that are encoded using a sentence embedding model (Figure 1).

We first transform each day into a fixed length vector by using the most recorded (modal) location for each 20 minute window and by assigning a a token, "Nowhere", to all windows without any sensor readings. This is then transformed into a text string. The SE-MiniLM model described in Section 2.1 is fine-tuned on this collection of day-strings using a triplet loss and by assigning a similar and dissimilar example for every string. Here, we make use of the pre-trained word embeddings' learnt semantic relationship between locations which we hypothesise allows for more meaningful encoding of activity. Inspired by Franceschi et al. (2019), for a given day-string we uniformly sample another string generated from the same participant within a 30-day window and label it as similar, and uniformly sample a string from a different participant to label as dissimilar. In this way, we are training the model to learn personalised representations of day-strings that allow for the detection of activity changes. Further details are available in Appendix A.3.

Daily embeddings can now be clustered using unsupervised techniques, searched over using vector searches, or similarity measured over time to estimate changes in behaviour.

3 Experiments

After using k-means to cluster the learnt embeddings, we found 5 clusters (Appendix A.4), revealing 5 categories of days. Figure 2 shows these clusters after embeddings are transformed using t-SNE (van der Maaten & Hinton, 2008). By viewing these t-SNE embeddings for different participants (on the upper right-hand side), we can visualise 4 participants' journeys in the embedding space (further participants are visualised in Appendix A.6). The lower right-hand plot shows the cluster value

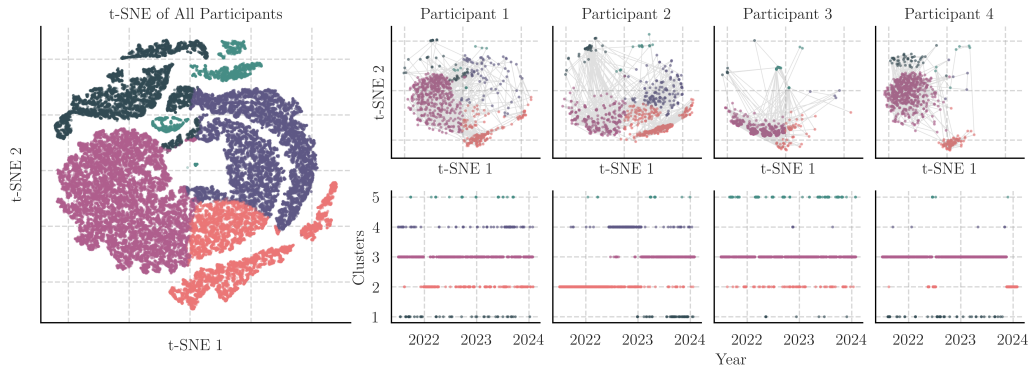

Figure 2: **t-SNE of embeddings.** The left plot shows the t-SNE transformation of all day-string embeddings, coloured by their cluster value. On the upper row of the right hand figures, we show the t-SNE embeddings for the 4 participants with the most data, where a line indicates consecutive days. On the lower row, we show the daily cluster values in time for the same participants.

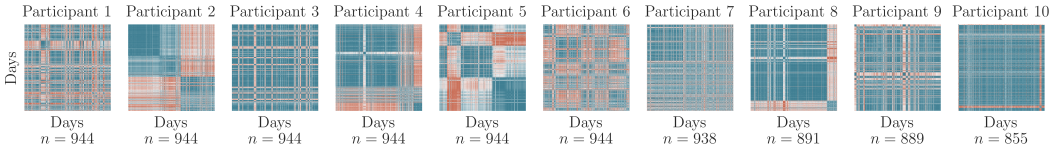

Figure 3: **Day similarity.** Each image shows the cosine similarity between each recorded day for 10 participants. Blue, white, and red correspond to a cosine similarity of 1, 0, and -1 respectively. The number of days represented by each axis of the plots is given by n.

for each day of data for 4 participants, which tells us how frequently different participants move between clusters or when activity changes. By exploring with clinicians what these clusters represent semantically, we can gain insight into participants' behaviour and care requirements over time. For example, we found that cluster 1 is characteristic of small amounts of recorded activity (Appendix A.5) and potentially adverse health conditions. In Appendix A.7, we visualise the proportion of participants in each cluster over time, gaining an insight into the behaviour of the cohort as a whole.

Using the cosine similarity between embeddings, for each day of data we can identify the most similar other days (Appendix A.8). We can also use this to find the days that are most similar to those labelled with adverse health conditions. For example, since we know the days that participants were tested for a urinary tract infection (UTI), we can calculate the similarity between days labelled as positive and negative for each participant. We find that the mean ($\pm$ standard deviation) of the average intra-participant cosine similarity between positive and negative days of UTI is 0.53 (± 0.36) and 0.39 (± 0.50) respectively, suggesting that for each participant, positive days of UTI are more similar than to negative days of UTI (Appendix A.9).

Furthermore, we can measure the cosine similarity between each day and each participant to visualise how their activity varies. This is shown in Figure 3, which reveals participants with frequent changes in home activity and participants with more consistent activity. For example, participants 1 and 6 have frequent changes in daily activity with small regions of consistency in their activity. However, participant 2 had very consistent activity that switched suddenly to a different state of consistent activity, suggesting a large change in daily routines. Visualising intra-participant similarity, as we have here, could notify healthcare monitoring teams of changes in behaviour that might require changes in care delivery or further investigation.

There are a few directions for future research that are worth highlighting. Firstly, we would like to explore other ways of converting our days of activity to text strings and avoid the aggregation of data using the mode (shown in Figure 1), and evaluate how synonyms of the location tokens might change the learnt representations (for example, if "lounge" is replaced with "living room"). By experimenting with synonyms of the locations used during training, we could gain insights into how

a fine-tuned model could be used in differing environments (for instance, we could explore whether a model fine-tuned on in-home data generates meaningful representations of data collected from an assisted living environment). Further, it would be useful to experiment with different assumptions for generating positive and negative samples when fine-tuning the language model and explore the limitations in the current assumptions. We could also consider whether a state space model (Gu & Dao, 2023), which avoids requiring a maximum sequence length, might be a useful replacement for the BERT-based model presented in this work. Additionally, we would like to explore alternative methods for clustering that involve domain knowledge from clinicians to improve the utility of activity clustering, and further understand the use-case for identifying similar days to those labelled with adverse health conditions. To improve our evaluation of the learnt representations, we aim to test their usefulness in a series of classification tasks to gain additional insights into the information they encode about a participant and their activity, as well as compare the results with methods not relying on pre-trained embeddings or language models.

4 Conclusion

This work presents the initial results of a discrete-valued time-series representation learning method, focused on embedding days of in-home activity data, and with applications to other domains. We show that the learnt embeddings are useful for clustering, vector search, and measuring behavioural change. We believe this method shows promise, and we look forward to exploring this further.

Author contributions

AC: Conceptualisation, Methodology, Software, Data Processing, Investigation, Writing - Original Draft, Review and Editing, Visualisation; **TC, YC:** Writing - Review and Editing; **PB:** Conceptualisation, Methodology, Writing - Original Draft, Review and Editing, Supervision, Funding Acquisition.

Acknowledgments

This study is funded by the UK Dementia Research Institute (UKDRI) Care Research and Technology Centre funded by the Medical Research Council (MRC), Alzheimer's Research UK, Alzheimer's Society (grant number: UKDRI–7002), and the UKRI Engineering and Physical Sciences Research Council (EPSRC) PROTECT Project (grant number: EP/W031892/1). Infrastructure support for this research was provided by the NIHR Imperial Biomedical Research Centre (BRC) and the UKRI Medical Research Council (MRC). The funders were not involved in the study design, data collection, data analysis or writing the manuscript.

References

Monica Agrawal, Stefan Hegselmann, Hunter Lang, et al. Large language models are few-shot clinical information extractors. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 1998–2022, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.130. URL https://aclanthology.org/2022.emnlp-main.130.

Abdullah Alanazi. Using machine learning for healthcare challenges and opportunities. *Informatics in Medicine Unlocked*, 30:100924, January 2022. ISSN 2352-9148. doi: 10.1016/j.imu.2022.100924.

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pp. 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2): 423–443, February 2019. ISSN 1939-3539. doi: 10.1109/TPAMI.2018.2798607.

Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141): 20170387, April 2018. doi: 10.1098/rsif.2017.0387.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Ning Ding, Yujia Qin, Guang Yang, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(33):220–235, March 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00626-4.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, et al. Time-series representation learning via temporal and contextual contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 2352–2359, Montreal, Canada, August 2021. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-9-6. doi: 10.24963/ijcai.2021/324. URL https://www.ijcai.org/proceedings/2021/324.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/53c6de78244e9f528eb3e1cda69699bb-Abstract.html.

Nate Gruver, Marc Anton Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Raia Hadsell, Dushyant Rao, Andrei A. Rusu, et al. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 24(12):1028–1040, December 2020. ISSN 13646613. doi: 10.1016/j.tics.2020.09.004.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, pp. 84–92. Springer, 2015.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, et al. A survey of transformers. *AI Open*, 3:111–132, 2022. ISSN 26666510. doi: 10.1016/j.aiopen.2022.10.001.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Tomas Mikolov, Ilya Sutskever, Kai Chen, et al. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://papers.nips.cc/paper_files/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

Kevin P. Murphy. *Probabilistic machine learning: an introduction*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts London, England, 2022. ISBN 978-0-262-04682-4.

The pandas development team. pandas-dev/pandas: Pandas, 2023. URL https://doi.org/10.5281/zenodo.10045529.

Adam Paszke, Sam Gross, Francisco Massa, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, et al. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763. PMLR, July 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Ashish Vaswani, Noam Shazeer, and Niki others Parmar. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wenhui Wang, Furu Wei, Li Dong, et al. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5776–5788. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yaqing Wang, Quanming Yao, James T. Kwok, et al. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), jun 2020b. ISSN 0360-0300. doi: 10.1145/3386252. URL https://doi.org/10.1145/3386252.

Jason Wei, Maarten Bosma, Vincent Zhao, et al. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

Thomas Wolf, Lysandre Debut, Victor Sanh, et al. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Peng Xu, Xiatian Zhu, and David A. Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2023. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2023.3275156.

Ling Yang and Shenda Hong. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 25038–25054. PMLR, June 2022. URL https://proceedings.mlr.press/v162/yang22e.html.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, et al. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pp. 2114–2124, New York, NY, USA, August 2021. Association for Computing Machinery. ISBN 978-1-4503-8332-5. doi: 10.1145/3447548.3467401. URL https://dl.acm.org/doi/10.1145/3447548.3467401.

A Appendix

A.1 Availability of the dataset and code

The code to fine-tune the model presented in this work is made available[1]. The dataset and IPython notebook used to generate the figures will not be made publicly available due to their sensitive nature.

Experiments were conducted using python 3.11.5, torch 2.1.0 (Paszke et al., 2019), transformers 4.34.1 (Wolf et al., 2020), sentence-transformers 2.4.0 (Reimers & Gurevych, 2019), scikit-learn 1.3.2 (Pedregosa et al., 2011), numpy 1.26.1 (Harris et al., 2020), and pandas 2.1.2 (pandas development team, 2023).

A.2 Further information on the dataset

Within this work, we evaluate methods on a dataset containing sensor recordings from the homes of people living with Dementia.

Passive infrared (PIR) sensors are installed at multiple locations within the homes of people living with Dementia, and a sleep mat is positioned under the mattress of the person with Dementia. The PIR sensors can detect motion within 9 metres and with a maximum angle of $45°$ and the sleep mat device can monitor when the participant enters or exits the bed. Within this work, we focused on PIR sensors located in the Lounge, Kitchen, Hallway, Bedroom, and Bathroom.

We analyse data collected between 2021-07-01 and 2024-01-30, containing 65,962 days from 134 participants. The average number of days recorded per participant is 492 (with a minimum of 5, median of 513, and maximum of 943). Figure 4 shows the distribution of the number of days recorded from each person living with Dementia, and 5 shows the sensor recordings by time of day, illustrating the richness of the dataset.

Each data point contains the participant ID, the timestamp of recording (to the nearest second), and the location at which activity was detected or whether the sleep mat recorded someone entering or exiting the bed. This amounted to 24,467,307 data points, shown in Figure 5.

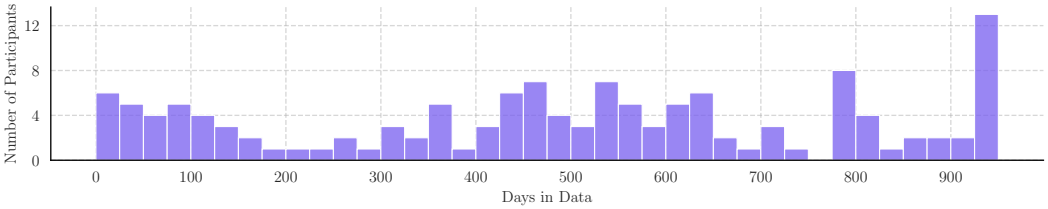

Figure 4: **Number of days recorded.** Histogram showing the number of participants with a given number of days of data recorded.

As part of the data collection, a subset of participants opted to regularly provide urine samples, which were analysed for suspicion of urinary tract infection. To calculate the similarity between labelled days presented in Section 3, we required participants to have at least 1 negative result and 2 positive results (discussed further in Appendix A.9). This left us with 154 days labelled as UTI positive (77 days) or negative (77 days).

A.3 Further details of our method

For each participant and day, we have access to the sensor recordings to the nearest second. As the tested language model is based on a predefined transformer architecture, it requires an input sequence of less than 256 tokens in length. Due to the irregular frequency of our time-series, we must aggregate recordings so that they have a fixed length. We do this by calculating the modal sensor reading for each 20 minute period (and uniformly sample from these modes when more than

[1] `https://github.com/alexcapstick/Text-Encoders-For-Daily-Movement-Data`

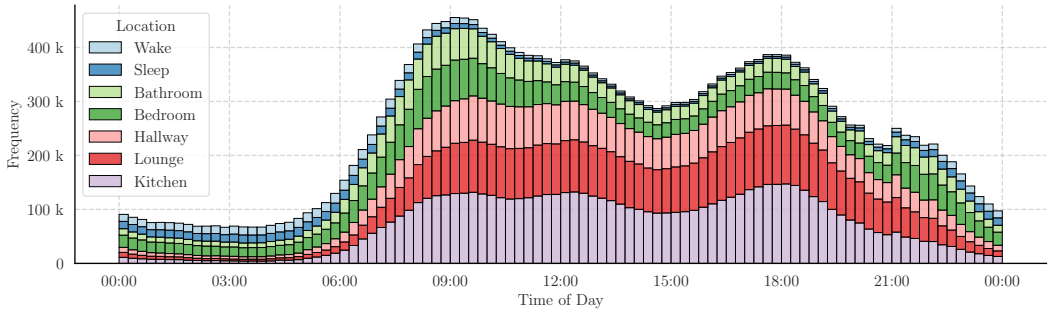

Figure 5: **Time of location recordings.** Histogram showing the number of sensor recordings by time of day.

one exists), which transforms each day into a set of tokens of length 72 (which is a discrete-valued and regular-frequency time-series). This time-series is then converted into a text string, which can be processed by a language model. Using a pre-trained model allows us to make use of the differences in the meaning of, for example, "Lounge", "Bedroom", and "Kitchen" and the literature surrounding sentence embeddings.

These text strings are processed by SE-MiniLM, a sentence embedding model based on a fine-tuned MiniLM V2 model (which itself was pre-trained) (Reimers & Gurevych, 2019; Wang et al., 2020a; Wolf et al., 2020) [2]. We further fine-tune this model by labelling day-strings produced within 30 days of each other by the same participant as similar and day-strings produced by different participants as dissimilar. This was done to learn differences between long-term changes in behaviour, and to group participants with similar daily routines into similar regions in the vector space defined by the sentence embeddings.

To fine-tune SE-MiniLM, we randomly sample triplets of text strings from the processed daily activity: one query sentence, one similar sentence, and one dissimilar sentence. For each epoch, we randomly sample 100,000 triplets from the dataset and assign them to batches of size 256. We evaluate the sentence embeddings using a triplet loss applied over each triplet, which was optimised using the AdamW algorithm (Loshchilov & Hutter, 2019) with a learning rate of 2×10^{-5} and a weight decay of 0.01. A linear warm-up learning rate scheduler was used with a warm-up of 10,000 steps.

A.4 Further clustering results

Table 1: **Silhouette score for k-means clustering on the base and fine-tuned models.**

Fine-tuning	2	3	4	5	6	7	8	9	10
True	0.39	0.47	0.51	0.57	0.54	0.53	0.52	0.52	0.51
False	0.42	0.21	0.18	0.16	0.16	0.14	0.14	0.17	0.16

k-means is used to cluster the embedded days of data. We use the k-means ++ algorithm (Arthur & Vassilvitskii, 2007) with a euclidean distance metric, and numbers of clusters varying from 2 to 10. Table A.4 shows the silhouette scores of this experiment, in which 5 is chosen as optimal.

Embeddings are normalised using the mean and standard deviation of each feature before clustering is performed.

A.5 Brief exploration of the meaning of clusters

In Figure, 6, we show days of data in cluster 5, which correspond to days with low levels of recorded activity, which are likely due to sensor recording failures, participants being away from home, or days with little in-home activity (possibly due to adverse health conditions).

[2] `https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2`

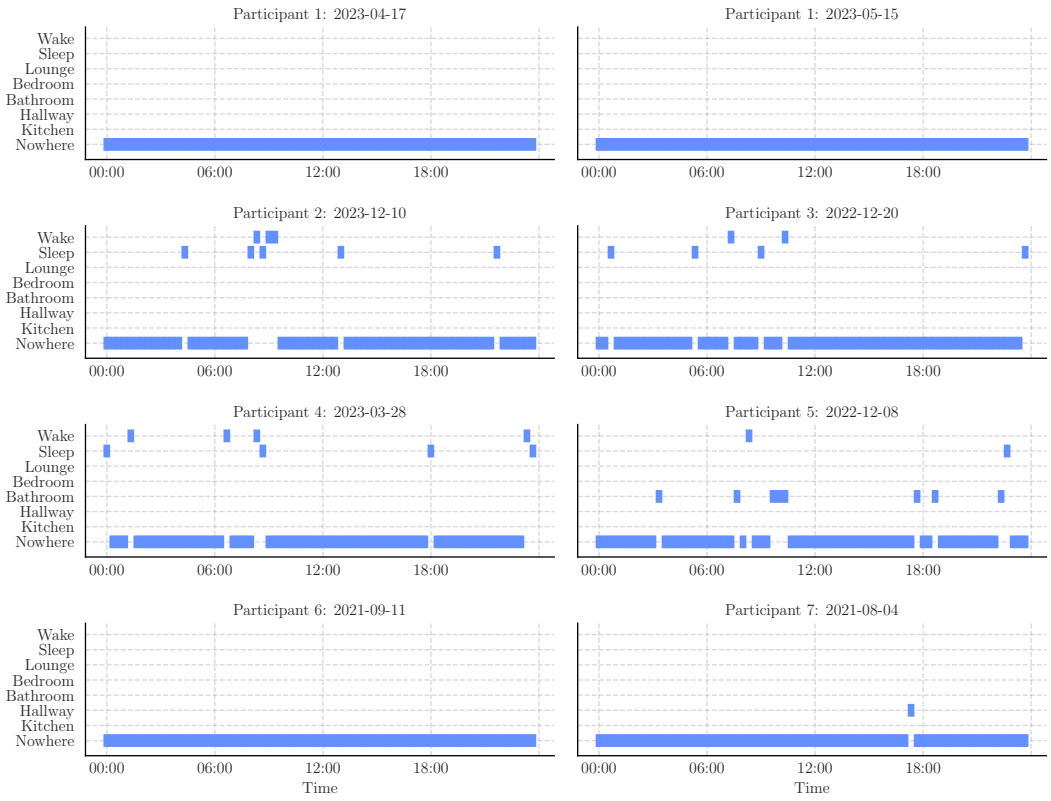

Figure 6: **Days of data in cluster 5.** We present 8 randomly chosen days of data from cluster 5, which is characteristic of sparse recorded activity.

Similarly, Figure 7 shows 8 random days from cluster 1, which could represent those days with high amounts of kitchen or bedroom activity. Although this cluster could be used to semantically classify a day, it does not seem to have meaningful clinical utility, suggesting more sophisticated methods for learning clusters are required.

A.6 Further t-SNE plots

To visualise the embedding space in two dimensions, we used t-SNE van der Maaten & Hinton (2008), a common dimension reduction method for visualisation. We set the perplexity to 30, early exaggeration factor to 12, and the learning rate to 1377 (number of samples divided by 48) (Pedregosa et al., 2011).

Embeddings are normalised using the mean and standard deviation of each feature before applying t-SNE.

In Figure 8 we present some additional participants and their journeys in the embedding space. Here we plot the 25 participants with the most recorded days of data. Notice that some participants' activity is significantly more localised than others, suggesting less variability in their activity day-to-day. We also observe some participants that have similar journeys in the transformed embedding space. Participants 15 and 20, for example, occupy similar areas in the t-SNE transformed vector space.

A.7 Clustering over time

In Figure 9, we present the proportion of participants with a given cluster label for each day in the dataset. Given a semantic understanding of the different days that each cluster represents, we can visualise the behaviour of the participant cohort as a whole. We observe that cluster 2 generally

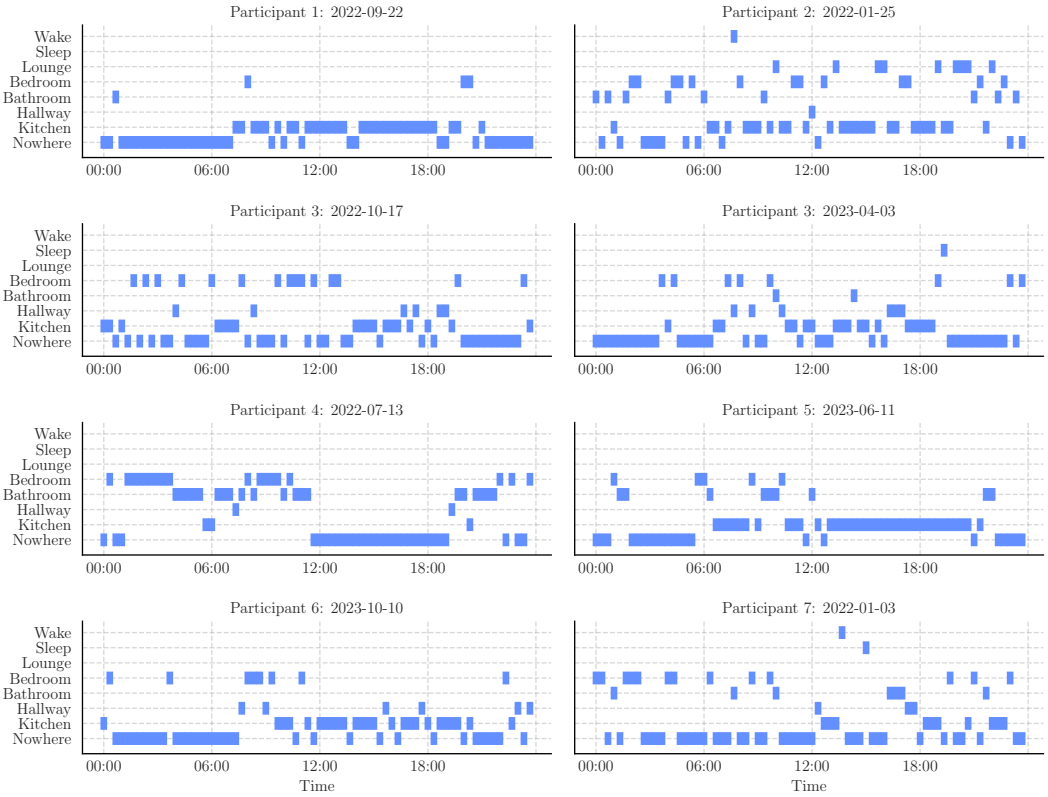

Figure 7: **Days of data in cluster 1.** We present 8 randomly chosen days of data from cluster 1, which is characteristic of frequent bedroom and kitchen activity.

increases in proportion over the length of the dataset, whilst cluster 3 seems to rise and fall in proportion between 2021-09 and 2022-09.

A.8 Vector Search over Days

To present the use of our method for finding similar days of activity, in Figure 10 we show a day of data (chosen at random) and the 9 most similar days as measured by cosine similarity. These days show similar aspects, and all have large levels of Kitchen and Hallway activity during the day, with small amounts of Lounge, Bedroom, and Bathroom activity.

A.9 Similarity in Days with Urinary Tract Infection Labels

Here, we will clarify the methods used to calculate the average cosine similarity between days positively labelled with urinary tract infections (UTIs), and those negatively labelled.

We firstly identify the participants and dates associated with a UTI label. We then keep only the participants with at least two positive UTI labels and one negative UTI label. This is done to ensure that for a given participant, we can calculate the similarity between the positive labels (excluding the cosine similarity between the same days) and the similarity between the positively labelled days and the negatively labelled days. After this, we are left with 77 positively and 77 negatively labelled days produced by 14 participants. For each of these participants, we calculate the average cosine similarity between positive days of UTI and the average cosine similarity between the positive and negative days of UTI. We then find the mean and standard deviation of these averages to produce the final values given in Section 3.

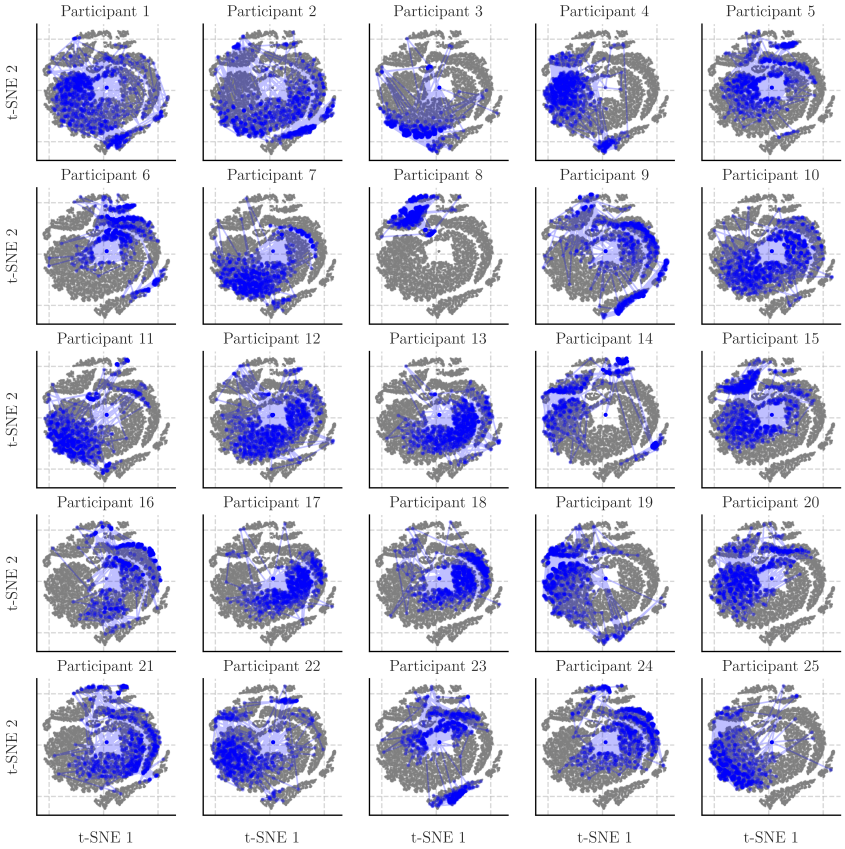

Figure 8: **t-SNE of embeddings and journeys.** The t-SNE embeddings of all of the data (in grey) with the embeddings of a given participant's data overlayed (in blue) with consecutive days connected by a line. Here, we show the 25 participants with the most recorded days of data.

Figure 9: **Cluster proportions.** Here, for each day in the dataset we show the proportion of days clustered with each label.

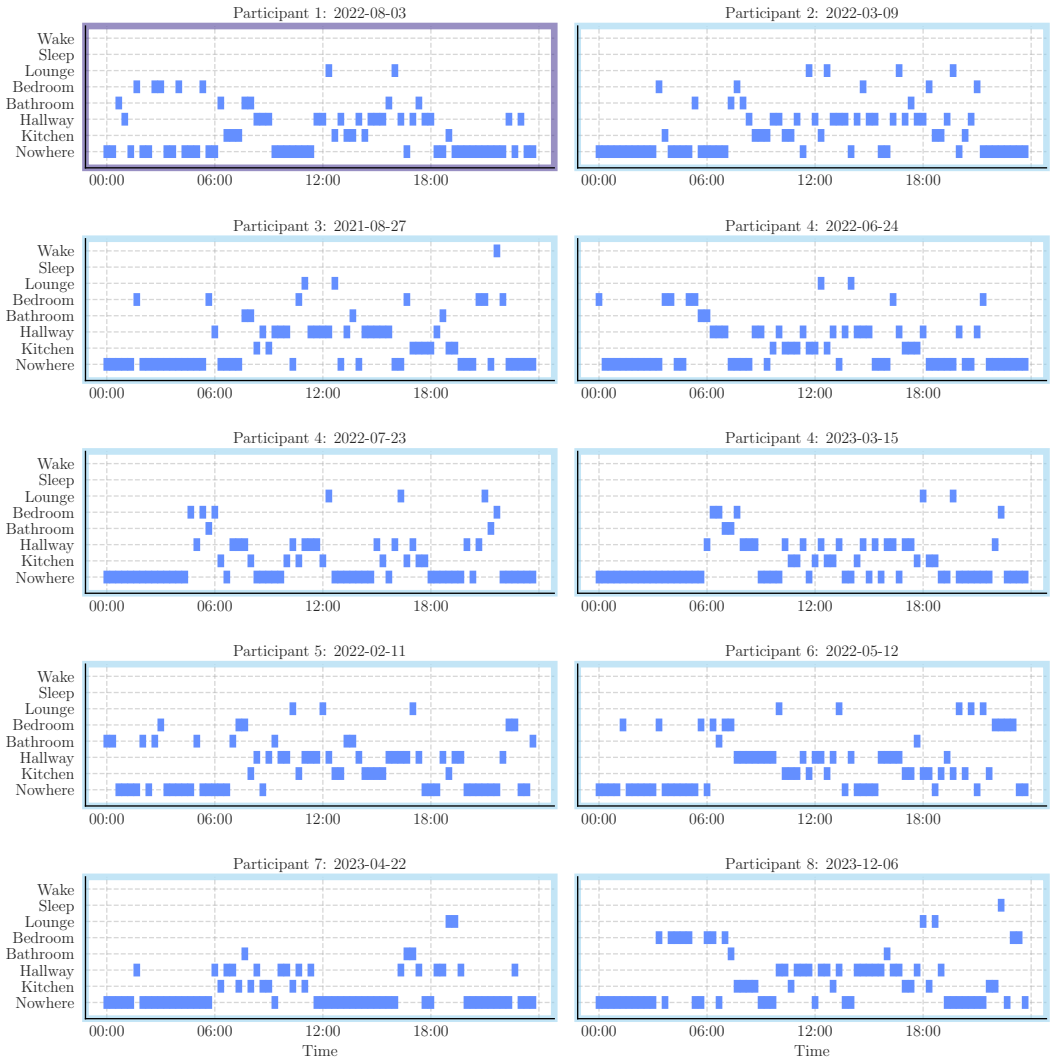

Figure 10: **Vector search over days.** Given a day of data (shown in purple in the top left), we present the 9 most similar other days as measured by cosine similarity (shown in blue).