nature computational science

Article

https://doi.org/10.1038/s43588-023-00573-5

Using sequences of life-events to predict human lives

Received: 6 June 2023

Accepted: 15 November 2023

Published online: 18 December 2023

Check for updates

Germans Savcisens[1], Tina Eliassi-Rad[2,3], Lars Kai Hansen[1], Laust Hvas Mortensen[4,5], Lau Lilleholt[6,7], Anna Rogers[8], Ingo Zettler[6,7] & Sune Lehmann[1,7]

Here we represent human lives in a way that shares structural similarity to language, and we exploit this similarity to adapt natural language processing techniques to examine the evolution and predictability of human lives based on detailed event sequences. We do this by drawing on a comprehensive registry dataset, which is available for Denmark across several years, and that includes information about life-events related to health, education, occupation, income, address and working hours, recorded with day-to-day resolution. We create embeddings of life-events in a single vector space, showing that this embedding space is robust and highly structured. Our models allow us to predict diverse outcomes ranging from early mortality to personality nuances, outperforming state-of-the-art models by a wide margin. Using methods for interpreting deep learning models, we probe the algorithm to understand the factors that enable our predictions. Our framework allows researchers to discover potential mechanisms that impact life outcomes as well as the associated possibilities for personalized interventions.

We live in the age of algorithm-driven prediction of human behavior. The predictions range from those at the global and population level, with societies allocating vast resources to predicting phenomena such as global warming[1] or the spread of infectious diseases[2], all the way to the constant flow of individual micro-predictions that shape our reality and behavior as we use social media[3]. When it comes to individual life outcomes, however, the picture is more complex. Sociodemographic factors play an important role in human lives[4], but, based on independent analyses of the same dataset, a recent collaboration of 160 teams has recently argued for practical upper limits for the predictions of life outcomes[5].

In this Article we find that, with highly detailed data, a different picture of individual-level predictability emerges. Drawing on a unique dataset consisting of detailed individual-level day-by-day records[6,7] describing the six million inhabitants of Denmark, and spanning a decade interval, we show that accurate individual predictions are indeed possible. Our dataset includes a host of indicators, such as health, professional occupation and affiliation, income level, residency, working hours and education (Dataset section).

The main reason why we are currently experiencing this 'age of human prediction' is the advent of massive datasets and powerful machine learning algorithms[8,9]. Over the past decade, machine learning has revolutionized the image- and text-processing fields by accessing ever larger datasets that have enabled increasingly complex models[10,11]. Language processing has evolved particularly rapidly, and transformer architectures have proven successful at capturing complex patterns in massive and unstructured sequences of words[12–14]. Although these models originated in natural language processing, their ability to capture structure in human language generalizes to other sequences[15–19] that share properties with language, for example, where sequence

[1]DTU Compute, Technical University of Denmark, Lyngby, Denmark. [2]Network Science Institute, Northeastern University, Boston, MA, USA. [3]Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA. [4]Data Science Lab, Statistics Denmark, Copenhagen, Denmark. [5]Department of Public Health, University of Copenhagen, Copenhagen, Denmark. [6]Department of Psychology, University of Copenhagen, Copenhagen, Denmark. [7]Copenhagen Center for Social Data Science (SODAS), University of Copenhagen, Copenhagen, Denmark. [8]Computer Science Department, IT University of Copenhagen, Copenhagen, Denmark. e-mail: sljo@dtu.dk

https://doi.org/10.1038/s43588-023-00573-5

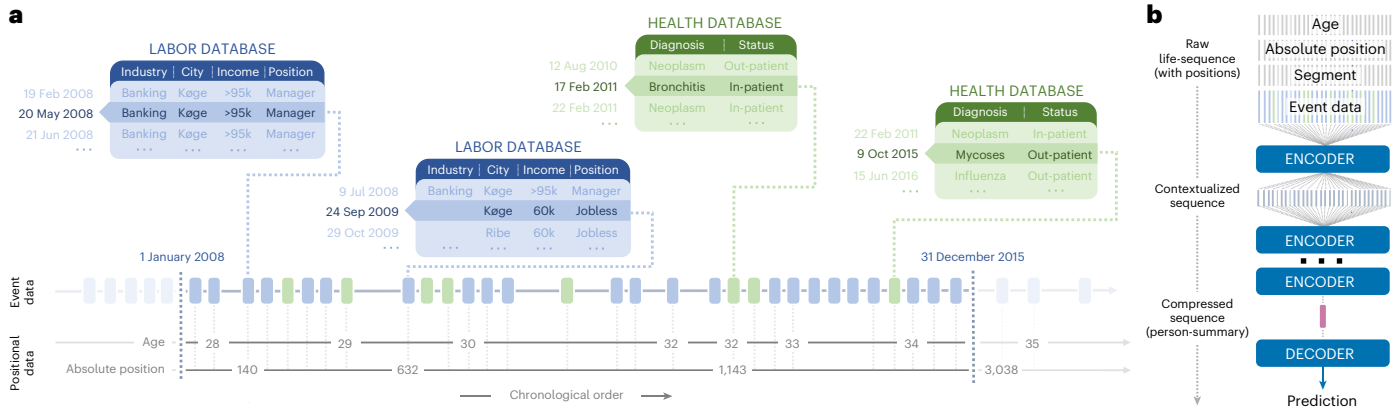

Fig. 1 | A schematic individual-level data representation for the life2vec model. a,**b**, We organize socio-economic and health data from the Danish national registers from 1 January 2008 to 31 December 2015 into a single chronologically ordered life-sequence (**a**). Each database entry becomes an event in the sequence, where an event has associated positional and contextual data. The contextual data include variables associated with the entry (for example, industry, city, income and job type). The positional data include the person's age (expressed in full years) and absolute position (number of days since 1 January 2008). The raw life-sequence is then passed to the model described in **b**. The model consists of multiple stacked encoders. The first encoder combines contextual and positional information to produce a contextual representation of each life-event. The following encoders output deep contextual representations of each life-event (considering the overall content of the life-sequence). The final encoder layer fuses the representations of life-events to produce the representation of a life-sequence. The decoder uses the latter to make predictions.

ordering is essential, and elements in the sequence can have meaning on many different levels. Importantly, due to the absence of large-scale data, transformer models have not been applied to multi-modal socio-economic data outside industry.

Our dataset changes this. The scale of our dataset allows us to construct sequence-level representations of individual human life-trajectories, which detail how each person moves through time. We can observe how individual lives evolve in a space of diverse event types (information about a heart attack is mixed with salary increases or information about moving from an urban to a rural area). The time resolution within each sequence and the total number of sequences are large enough that we can meaningfully apply transformer-based models to make predictions of life outcomes. This means that representation learning can be applied to an entirely new domain to develop a new understanding of the evolution and predictability of human lives. Specifically, we adopt a BERT-like architecture[20,21] (BERT, bidirectional encoder representations from transformers) to predict two very different aspects of human lives: time of death and personality nuances (additional predictions are presented in Supplementary Table 7). To make these predictions, our model relies on a common embedding space for all events in the life-trajectories. Just as embedding spaces in language models can be studied to provide a novel understanding of human languages[22,23], we study the concept of embedding space to reveal non-trivial relationships between life-events.

Results

Approach overview

We represent the progression of individual lives as 'life-sequences' (Fig. 1). The life-sequences are constructed based on labor and health records from Danish national registers[6,7], which contain highly detailed data for all approximately six million Danish citizens. Our 'labor' dataset[24] includes records about income, such as salary, scholarship, job type[25], industry[26], social benefits and so on. The 'health' dataset[6] includes records about visits to healthcare professionals or hospitals, accompanied by the diagnosis (hierarchically organized via the so-called ICD-10 system[27]), patient type and urgency. Life-sequences evolve over time and provide rich information about life-events with high temporal resolution. Our full dataset runs from 2008 to 2020 and includes all individuals who live in Denmark, but, for the analyses discussed in the following, we filter the dataset, focusing on the period 2008–2016 and an age-limited subset of individuals.

The raw stream of temporal data has traditionally posed substantial methodological challenges, such as irregular sampling rates, sparsity, complex interactions between features, and a large number of dimensions[28]. Classical methods for time-series analysis[29,30] become cumbersome because they are challenging to scale, inflexible, and require considerable preprocessing. Transformer methods allow us to avoid hand-crafted features and instead encode the data in a way that exploits the similarity to language[15,18]. Further, transformers are well-suited for representing life-sequences due to their ability to compress contextual information[13,31] and take into account temporal and positional information[18,32]. We call our transformer architecture[20,21,33–37] life2vec.

As we establish the life-sequences, each category of discrete features and discretized continuous features form a vocabulary, and in that sense we can create a kind of synthetic language. This vocabulary—along with an encoding of time—allows us to represent each life-event (including its detailed qualifying information) as a 'sentence' composed of synthetic words, or 'concept tokens'. We attach two temporal indicators to every event: One that specifies the individual's age at the time of the event and one that captures absolute time (Fig. 1 and Supplementary Fig. 1).

Thus, our synthetic language can capture information along the lines of 'In September 2012, Francisco received twenty thousand Danish kroner as a guard at a castle in Elsinore' or 'During her third year at secondary boarding school, Hermione followed five elective classes'. Using this approach, we can form individual life-sequences that allow us to encode detailed information about events in individual lives without sacrificing the content and structure of the raw data.

Understanding relations between concepts

Just as large language models establish word embeddings that capture complex relationships between words[20], pretraining life2vec (Training procedure section) establishes a shared concept space that contains everything from diagnoses via job types and place of residence to income levels. This concept space forms the foundation for the predictions we make using the life2vec model.

Before making predictions, we explore the concept space. This is important for two reasons. First, an understanding of the concept space will help us understand what enables the model to make accurate

https://doi.org/10.1038/s43588-023-00573-5

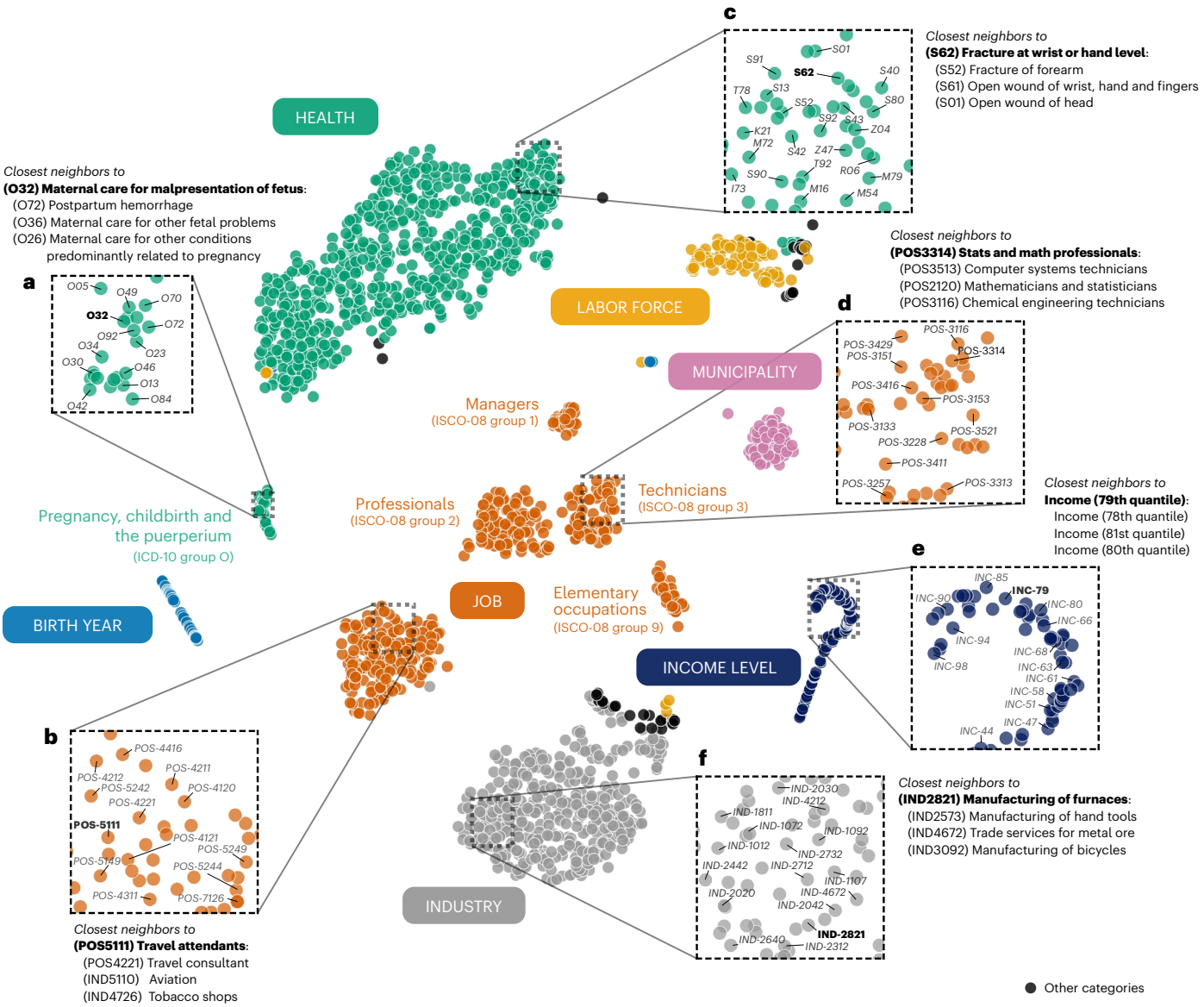

Fig. 2 | Two-dimensional projection of the concept space (using PaCMAP). Each point corresponds to a concept token in the vocabulary (n = 2,043). Points are colored based on the concept types (infrequent types are represented as black points). Each region provides a zoom of a part of the concept space. The top three closest neighbors for selected tokens (based on the cosine distance) are also displayed. **a**, Diagnoses related to pregnancy, childbirth and the puerperium in ICD-10[27]. **b**, Job concepts related to service and sales workers (corresponds to job category 5 of ISCO-08[25]). **c**, Injury-related diagnoses in ICD-10[27]. **d**, Job concepts related to technicians and associate professionals (corresponds to job category 3 of ISCO-08[25]). **e**, Income-related concepts. life2vec arranges these concepts in increasing ordinal order. **f**, Concepts related to the manufacturing industry in DB07[26].

predictions. Second, the concept space contains information about the relationships between individual concepts, so, by exploring this space, we can learn about the world that has generated the life-sequences. In Fig. 2, the original 280-dimensional concepts are projected onto a two-dimensional manifold with the use of PaCMAP[38], which preserves the local and global structures of the high-dimensional space.

In Fig. 2, each concept is colored according to its type. This coloring makes it clear that the overall structure is organized according to the key concepts of the synthetic language—health, job type, municipality and so on—but with interesting subdivisions, separating birth year, income, social status and other key demographic pieces of information. The structure of this space is highly robust and emerges reliably under a range of conditions (Robustness of the concept space section and Supplementary Tables 5 and 6).

Digging deeper than the global layout, we find that the model has learned intricate associations between nearby concepts. We investigate these local structures via neighbor analysis, which draws on the cosine distance between concepts in the original high-dimensional representations as a similarity measure. A key area to consider is the cluster formed by income (Fig. 2, dark blue points). What the model sees is 100 concept tokens, each describing a level of income. Before training, it has no a priori idea of what each one means; each token is simply an arbitrary string of text among other strings. From the life-sequences, the model not only learns that income is different from other concepts (the dark blue points are isolated), but it also perfectly sorts the 100 levels. The blue curve starts with the token corresponding to the first percentile salaries and organizes them up to the 100th. Thus, the concepts most similar to the 59th percentile of income are the 58th and the 60th. Similarly, for birth years (Fig. 2, light blue), the closest concepts to the birth year 1963 are 1962 and 1964, and so on.

The health-type cluster (Fig. 2, green points) has a compact local structure. Diagnoses belonging to the same ICD-10[27] chapters cluster

according to their chapter. For example, the concept 'malignant neoplasm of stomach' (C16 in ICD-10) is surrounded by other C-chapter concepts, such as 'malignant neoplasm of lungs' (C34) and 'malignant neoplasm of colon' (C18). As shown in Fig. 2a, one of the clearly separated health clusters relates to pregnancies and childbirth diagnoses (that is, O-chapter concepts).

The concepts of professional occupations also cluster into smaller groups. These groups roughly correspond to the major groups of the International Standard Classification of Occupations (ISCO-08)[25]. Clearly defined clusters exist for the first (managerial and executive positions), second (professionals), third (technicians and associate professionals) and ninth (elementary occupations) groups.

Not all concept tokens are surrounded by tokens of the same category, but, even in these cases, the neighborhoods are meaningful. In Fig. 2b, the job concept of a 'travel agent' is surrounded by the job concept of a 'travel consultant' and an industry concept of 'aviation'.

Similarly, when the model does mix up ICD-10 codes, the 'mistakes' are meaningful. For example, the concept of Z95 (presence of cardiac and vascular implants and grafts) is surrounded by concepts corresponding to ICD-10 chapter I[27], for example, I42 (cardiomyopathy), I50 (heart failure) and I25 (chronic ischemic heart disease). The model's ability to group similar concepts that are not necessarily close in the standard classification systems is one of the strengths of our approach. Understanding which life-events play equivalent roles in human lives is one of the aspects that allow for improved classification and recommendation.

Predicting early mortality

Having confirmed that the concept space is robust and indeed captures meaningful structure in the data, we tested the ability of life2vec to make accurate predictions. Specifically, we estimated the likelihood of a person surviving the four years following 1 January 2016 (we have data up to 2020, but only train on data up to 2016 to avoid information leakage). Mortality prediction is an oft-used task within statistical modeling[39], which is closely related to other health-prediction tasks and therefore requires life2vec to model the progression of individual health-sequences as well as labor history to predict the correct outcome successfully. Specifically, given a sequence representation, life2vec infers the probability of a person surviving the four years following the end of our sequences (1 January 2016). For this task, we focus on making predictions for a young cohort of individuals in the age range 35–65 years, where mortality is challenging to predict. We note that our embeddings are robust to changes in the training data (Robustness of the concept space section).

This prediction task has an additional level of complexity, as the data contain people with unknown outcomes (that is, emigrants and missing individuals). We thus use positive-unlabeled learning[40,41], which provides a corrected performance metric for the model evaluation.

The performance of life2vec in relation to a range of baseline models[42]—actuarial life tables, logistic regression, feed-forward neural networks and recurrent neural networks (RNNs)—is shown in Fig. 3 and summarized in Supplementary Table 2 (additional life2vec performance details are provided in Supplementary Figs. 3–7).

We illustrate the performance of models using the corrected Matthews correlation coefficient (C-MCC[43]; Early mortality prediction section), which adjusts the MCC value for unlabeled samples. With a mean C-MCC score of 0.41 (95% confidence interval [0.40, 0.42]), life2vec outperforms the baselines by 11% (Fig. 3; note that increasing the size of RNN models does not improve their performance).

Our study population is heterogeneous in terms of age and sex across the eight-year period. Individuals may also have many or few tokens available. To understand the effects of this heterogeneity, Fig. 3b breaks down the performance for various subgroups: intersectional groups based on age and sex, as well as groups based on sequence length (Supplementary Information section 1).

In terms of age and gender, the model performs better on a younger cohort of the population and on a cohort of women. Furthermore, sequence length (a proxy for the number of life-events in a sequence) does not have a substantial impact on the performance of a model (Fig. 3b).

Task-specific representations of individuals

When we make predictions using life2vec, we establish a new vector space specific to the prediction task. In this vector space, each life-sequence is summarized by the information most useful for the prediction task. This person-summary is a single vector that encapsulates the essential aspects of an individual's entire sequence of life-events relative to a certain prediction. In the following, we focus on person-summaries for the case of mortality likelihood, but person-summaries relative to, for example, change in the area of residence or choice of the university would be drastically different.

By exploring the structure of the space of person-summaries, we can understand which factors drive a certain prediction, revealing how life2vec uses information from the concept space.

The space of person-summaries is visualized in Fig. 4a–g. Relative to the mortality prediction, the model organizes individuals on a continuum from low to high estimated probability of mortality (the point cloud in Fig 4d). In Fig. 4 we show true deceased by purple diamonds, and the confidence of predictions[44] is demonstrated by the radius of points (for example, dots with a small radius are low-confidence predictions). Furthermore, the estimated probability is displayed using a color map from yellow to green. We zoom in on two regions: region 1, which shows an area with a high probability of the 'survive' outcome, and region 2, which has a high probability of the 'death' outcome. We see that, although region 2 has a majority of elderly individuals, we still see a large fraction of younger individuals (Fig. 4f), and it contains a large fraction of true targets (Fig. 4g). Region 1 has a largely opposite structure, with a majority of young individuals but a substantial number of older individuals as well (Fig. 4b), and only a single actual death (Fig. 4c). When we look into actual deaths in the low-probability region, we find that the five deaths nearest to and in region 1 have the following causes—two accidents, malignant neoplasm of the brain (C71.9), malignant neoplasm of cervix uteri (C53.8) and myocardial infarction (I21.9). All these are causes of death that we would expect to be difficult to predict from life-event sequences.

Testing with concept activation vectors (TCAV)[45] provides a way to understand the meaning of directions in the person embedding space using labeled data. The idea behind TCAV is to use binary labeled data (for example, the labels 'employed'/'unemployed') and identify the hyperplane that best separates those labels. The vector orthogonal to this hyperplane gives us a direction for 'employed'–'unemployed' in the embedding space (the concept activation vector[45]). We then use this employment direction to understand how that label impacts decisions. Specifically, we measure how moving our decision boundary along this direction changes predictions. How the prediction reacts to these changes is called the 'concept sensitivity'.

Figure 4h,i shows the concept sensitivity scores for several labels relative to the mortality prediction task. Here we show a two-dimensional projection using DensMap[46], but a range of other low-dimensional projections (*t*-SNE, UMAP and PaCMAP[38]) are visualized in Supplementary Fig. 8. We focus on health-related labels such as mental health, the nervous system and parasites. Similarly, we use socio-economic attributes as labels to measure the model's sensitivity to major occupational groups and sex. Figure 4h shows labels in relation to the prediction 'survive', and Fig. 4i shows concepts with respect to the prediction 'death' within the four years following our sequence. Values close to one imply that moving in the topic direction indicates that moving in the label-direction increases the probability of a specific outcome, and values close to zero indicate no effect on an outcome. The gray areas are what we would expect if we moved in a random

https://doi.org/10.1038/s43588-023-00573-5

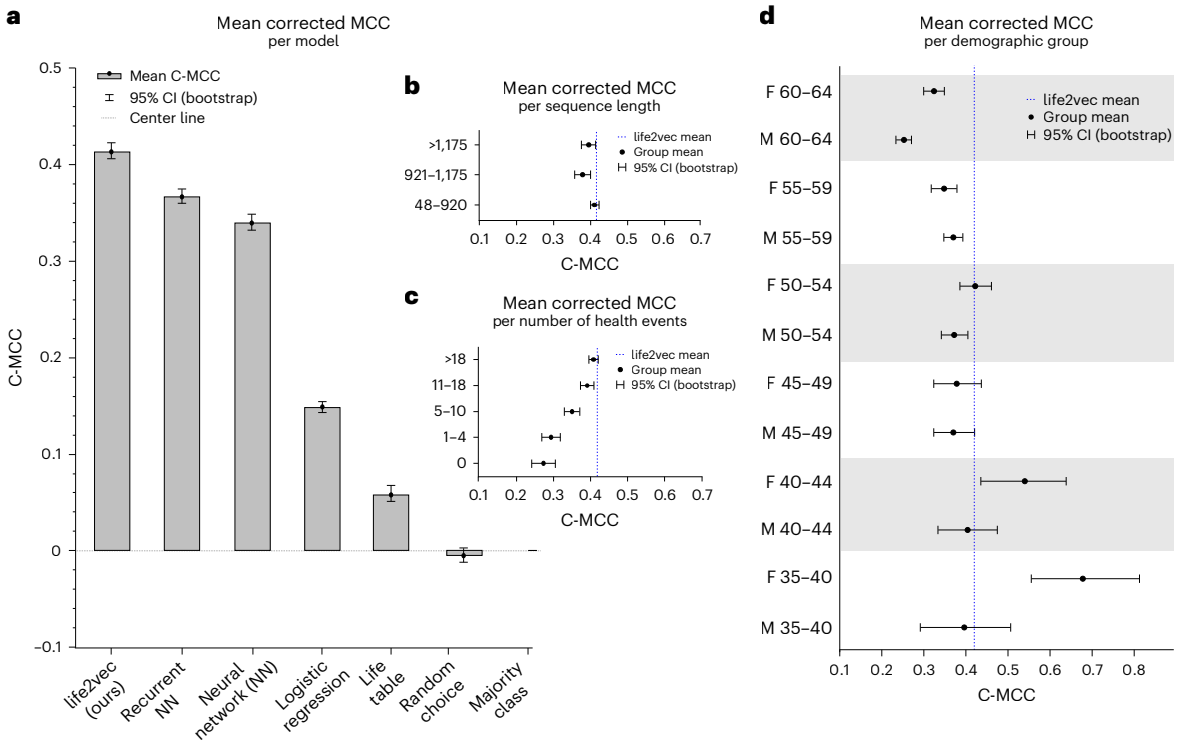

Fig. 3 | Performance of models on the mortality prediction task quantified with the mean C-MCC with 95% confidence interval. a, Comparison of life2vec performance to baselines (n = 100,000). **b**–**d**, Performance of life2vec on different cohorts of the population: performance of life2vec per sequence length (**b**), performance of life2vec based on the number of health events in a sequence (**c**) and performance of life2vec per intersectional group (based on age group and sex) (**d**). F, female; M, male.

direction. We see that directions of possessing a managerial position or having a high income nudge the model towards the 'survive' decisions (Fig. 4h), while being male, a skilled worker, or having a mental diagnosis has the opposite effect (Fig. 4i). Note that, although the bar charts in Fig. 4h,i are almost mirrors, they are created based on different datasets, validating robustness.

To further confirm the validity of the sensitivity scores, we performed extensive significance testing (Interpretability of the early mortality predictions section). Our final approach to understanding the person-summaries is via inspection of the model's attention to individual sequences[47,48]—this confirms the findings discussed above (Supplementary Information section 5).

life2vec as a foundation model

The power of life2vec is that it is a 'foundation model'[49] in the sense that the concept space can serve as a foundation for many different predictions, similar to the role played by word embeddings in large language models. In this section, we discuss aspects of how life2vec generalizes.

Death as a prediction target is well-defined and eminently measurable. To showcase the versatility of life2vec, we now predict personality, an outcome at the other end of the measurement spectrum, something that is internal to an individual and typically measured via questionnaires. In spite of the difficulty in measurement, personality is an important feature, related to people's thoughts, feelings and behavior, that shapes life outcomes[50].

Specifically, we predict all ten 'personality nuances' in the extraversion dimension. Nuances are actual responses on a 1–5 scale from 'strongly disagree' to 'strongly agree' to specific personality questionnaire items. We focus on individual nuances rather than aggregated personality-scores that average multiple questionnaire items. This choice is motivated by recent literature within personality psychology that emphasizes how nuances associate more strongly with life outcomes than aggregate measures[51]. We focus on extraversion, because the corresponding personality nuances are part of virtually all comprehensive models of the basic personality structure that have emerged over the last century, including the Big Five[52] and HEXACO[53] frameworks.

For prediction targets, we draw on data collected for a large and largely representative group of individuals in 'The Danish Personality and Social Behavior Panel' (POSAP) study[54] (Dataset section), and we make predictions for individuals in the age range 25–70 years and for the time period from 2008 to 2016. We predict all ten extraversion nuances.

Figure 5 shows that applying life2vec to life-sequences not only allows us to predict early mortality, but it is versatile enough to also capture personality nuances (Task-specific finetuning section). life2vec produces better scores than the RNN for most items, but the difference is only statistically significant on questionnaire items 3, 6, 8 and 9 (Fig. 5 provides the item wording). For item 7, the RNN does significantly better than random, whereas life2vec does not.

We illustrated the versatility of life2vec further by means of additional prediction tasks (Supplementary Table 7).

Note that we do not a priori expect life2vec to perform better than RNNs. Both models are trained on the same data representation, and what makes life2vec a more exciting model is not just the predictive power, but that its concept space is entirely general and thus an interesting object to analyze in its own right. In contrast, RNNs are task-specific, and their embedding spaces are only organized with respect to a single outcome.

The reason life2vec performs better than the RNN is likely because the self-attention mechanism allows individual tokens to interact across the entire sequence, capturing nuanced long-term effects[13]. This means that the more general model is able to form a superior representation of the complex and high-dimensional data.

Our current benchmarks compare life2vec to other models applied to the same dataset. However, this comparison does not illuminate the role of the multifaceted dataset itself in making accurate predictions.

https://doi.org/10.1038/s43588-023-00573-5

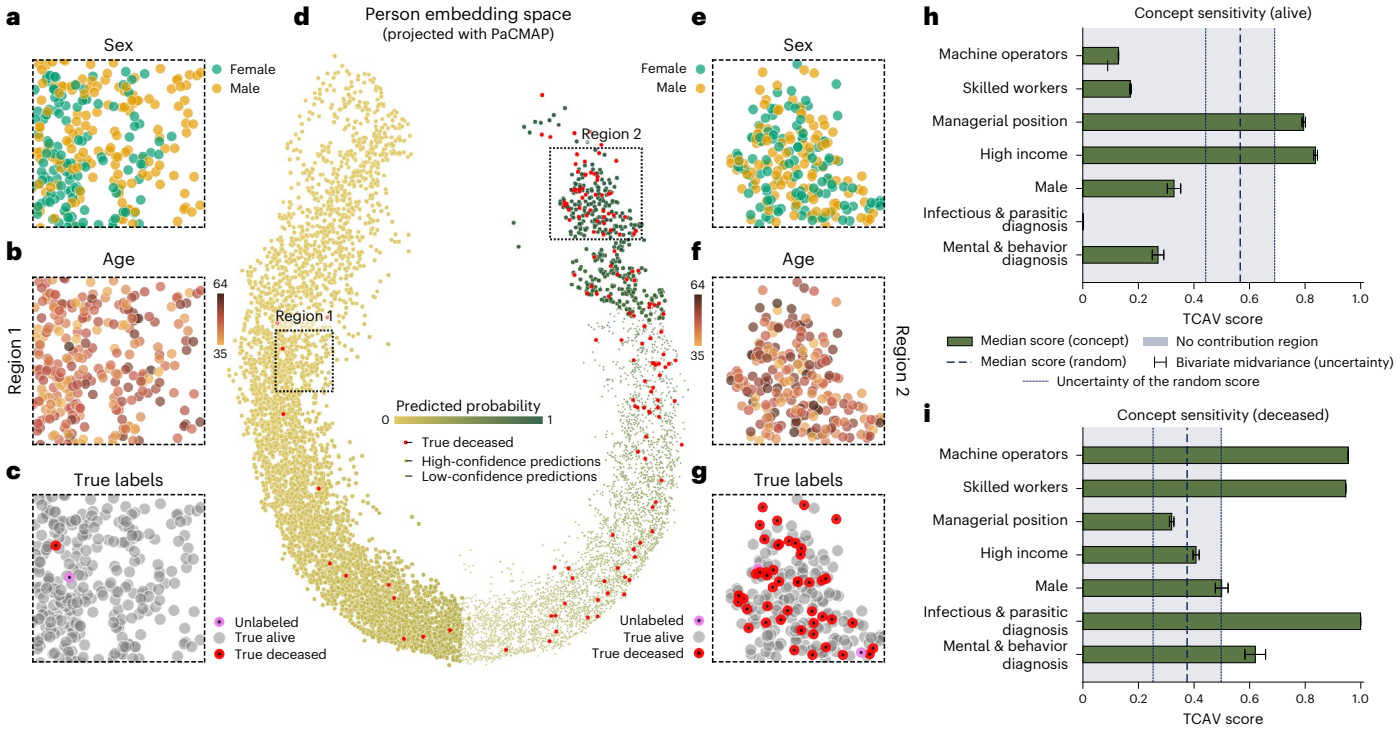

Fig. 4 | Representation of life-sequences conditioned on mortality predictions. a–g, Two-dimensional projection of 280-dimensional life representations using the DensMap method[46]. The full projection in **d** is colored based on the estimated probability of mortality. Red points stand for the true deceased targets. Points with a smaller radius are uncertain predictions. **a–c** and **e–g** show zoomed-in regions with additional aspects associated with the life-sequence. Region 1 contains points with a low probability of mortality (**a–c**), and region 2 contains points with a high probability (**e–g**). **h,i**, Bar plots of the concept sensitivity of life2vec with respect to the 'alive' prediction (**h**) and with respect to the 'deceased' prediction (**i**). Blue dashed lines show the median score for random concept directions. The dotted blue lines specify the bivariate midvariance (uncertainty) of the scores associated with the random concept direction ($n = 10{,}000$). The light blue area specifies the region without significant contribution towards the particular prediction.

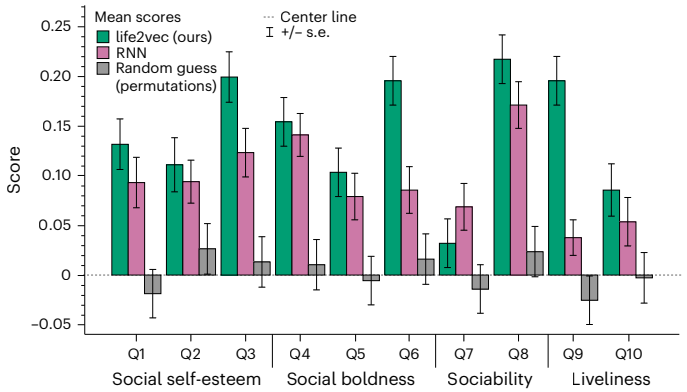

Fig. 5 | Performance evaluation for the personality nuances task. Cohen's quadratic kappa score, κ, for each of the ten extraversion questionnaire items ($n = 1{,}417$). The bars represent κ for life2vec (green), RNN (purple) and a random guess that draws predictions from the actual distribution of targets (gray). The error bars and whiskers correspond to ±1 s.e. of κ. The dashed line corresponds to $\kappa = 0$. The question wordings are provided in the Personality nuances prediction task section.

To understand the role of the various aspects of the data, we evaluated the performance of life2vec on four data variations to determine the contribution of various aspects of the data (Supplementary Table 4). Specifically, we consider full labor, partial labor (a subset of labor that removes information related to the employer), partial labor and health (including all the health data) and full labor and health, and we keep the cohort constant across all predictions to understand the effect of changing the underlying data.

This analysis confirms that our performance really does depend on having all of the data. Performance continues to improve as we add new data. The predictive power arises not from one single factor, but from a combination of all of the facets of data we include. For example, it is interesting to see that using the full labor data makes a large difference, both with and without the health data.

The data used in this Article are unique to Denmark, so it is interesting to consider how well the embedding spaces might reflect other populations. Just as in the case of large language models it is possible to use transfer learning or start from pretrained embeddings, could we use the life2vec embedding spaces for other populations? We cannot answer this question definitively, but note that in economic and sociological work on labor markets, a large body of literature has examined the work trajectories of individuals across Europe. This literature shows that the experiences generalize between contexts[55,56]. Similar general socio-economic positions and health patterns are also shared among a diverse set of countries[57,58]. These results suggest, therefore, that life2vec could be relevant in the context of other European countries and perhaps beyond (Ethics and broader impacts section).

Discussion

Our dataset is vast in size and covers every single person in a small nation. That said, there are still limitations. For now, we can only look at data across an eight-year period and for a subset of users aged 25–70 years (and 35–65 years for early mortality prediction) (Dataset section). Furthermore, although every person in Denmark appears in the registries, there may be sociodemographic biases in the sampling. For example, if someone does not have a salary—or chooses not to engage with the healthcare systems—we do not have access to their data (Ethics and broader impacts section).

Beyond this Article, life2vec opens a range of possibilities within the social and health sciences. By means of a rich dataset, we can capture complex patterns and trends in individual lives and represent their stories in a compact vector representation. Event sequences are a common data format in the social sciences[59], and our work shows how powerful transformer methods can be in unveiling the patterns encoded in such data. In our case, the embedding vectors represent a new type of comprehensive linkage between social and health outcomes. The output of our model, coupled with causality tools, shows a path to (1) systematically explore how different data modalities are correlated and interlinked and (2) use these interlinkages to explicitly explore how life impacts our health and vice versa.

It is entirely possible to imagine incorporating other types of information, from the unstructured behavioral data seen in online behavior to mobility data, or even the complex networks of social relationships. Our framework thus allows computational social science researchers to establish comprehensive models of human lives in a single representation. In this sense, we can open the door to a new and more profound interplay between the social and health sciences.

Finally, we stress that our work is an exploration of what is possible, but it should only be used in real-world applications under regulations that protect the rights of individuals (Ethics and broader impacts section).

Methods

Ethics and broader impacts

The data analysis was conducted at Statistics Denmark, the Danish National Statistical Institution, under the Danish Data Protection Act and the General Data Protection Regulation (GDPR)[60]. In this context, because the data were used for scientific and statistical purposes, the usage is partially exempt from the GDPR[60] (for example, from the right to be forgotten). Denmark-based academic researchers, government agencies, NGOs and private companies can be given access to Statistics Denmark data, but access is only granted under strict information security and data confidentiality policies (https://www.dst.dk/en/OmDS/strategi-og-kvalitet/datasikkerhed-i-danmarks-statistik) that ensure that data on individual entities are not leaked or used for purposes other than scientific. This focus on safekeeping data is shared with most other national statistical institutions that provide similar services. Using scientific/statistical 'products' such as life2vec for automated individual decision-making, profiling or accessing individual-level data that may be memorized by the model is strictly disallowed. Aggregate statistics, including those coming from model predictions, may be used for research and to inform policy development.

We stress that life2vec is a research prototype, and, in its current state, it is not meant to be deployed in any concrete real-world tasks. Before it could be used, for example, to inform public policies in Denmark, it should be audited, in particular, to ensure the demographic fairness[61] of its predictions (with respect to the appropriate fairness metrics for the given context) and explainability[62] (for example, if used for assisting decision-making based on synthetic/counterfactual data). Such audits will probably soon be mandated by the AI Act[63], focusing on the safe use of 'high-risk' models. Further auditing information is provided in Supplementary Information section 1.

Finally, we note that, although it is possible that phenomena captured by life2vec reflect phenomena that have similar distributions outside Denmark (for example, labor market trajectories and individual health trajectories), we urge caution with extrapolation to other populations, as we have not explored how our findings translate beyond the current study population.

Dataset

We worked with the Labour Market Account (AMRUN)[24] and National Patient Registry (LPR) datasets[6,27]. Within the Labour Market Account dataset are event data for every resident of Denmark. For Danish residents who have been in contact with secondary healthcare services, primarily hospitals, the events are recorded in the National Patient Registry. We limited ourselves to data recorded in the period from 2008 until the end of 2015. The datasets were pseudonymized before our work by de-identifying addresses, Central Person Register numbers (CPRs) and names. The data are stored within Statistics Denmark, and all access/use of data is logged.

The total number of residents in the filtered dataset was 3,252,086 (1,630,082 men and 1,622,004 women). For our research, we chose people who (1) were alive and lived in Denmark on 31 December 2015, (2) had at least 12 records in the labor data during 2015 (corresponds to 12 incomes over one year, for example salary, pension and so on; we did not set requirements on the health-set, as not every resident had any records in the health dataset), (3) had consistent sex and birthday attributes over the whole residency period, (4) were between 25 and 70 years old on 31 December 2015.

These prerequisites applied for both stages—pretraining and finetuning (that is, early mortality and personality nuances prediction tasks).

For the mortality prediction task, we excluded young individuals with very low death rates and older individuals with a high background probability of death. Thus, we narrowed the specification of requirement 4 and limited the dataset to people who were between 35 and 65 years old on 31 December 2015 (limiting us to 2,301,993 individuals, with 1,153,443 men and 1,148,550 women).

For the personality nuances prediction task, we did not alter the requirement for pretraining (ages 25–70 years) (4) but added new requirements on top of the original ones: (5) residents should have participated in the POSAP Study[54] and (6) none of the scores associated with any HEXACO personality nuance (facet, dimension) were missing. This resulted in analyzing the responses of 9,794 people (4,393 men and 5,401 women, aged the 25 to 75 years).

Specifically, in the POSAP study, HEXACO-60[53,54] was administered, comprising 60 items (each representing one personality nuance) that could be further aggregated into 24 personality facets and, in turn, six personality dimensions (honesty-humility, emotionality, extraversion, agreeableness versus anger, conscientiousness, openness to experience).

Labor data. The Labour Marked Accounts dataset[24] contains data on each taxable income a resident receives, such as salary, state scholarship, pension and so on. Each taxable income has multiple associated features, and we focused on 16 features (Supplementary Table 2). Some of these features are linked to the workplace: type of enterprise[64], industry code[26]. Others describe personal attributes: professional positions[25], labor force status, labor force status modifier, residential municipality, income, working hours, tax bracket, age, country of origin and sex.

The 'type of enterprise' feature is based on the European System of Accounts (ESA2010)[64], whereas the industry codes are encoded in the Danish Industry Code (DB07)[26]. Industry codes provide information about the type of services a company offers. For example, code 108400 stands for 'Preparation of flavorings and spices' and 643040 for 'Venture companies and private equity funds'. ESA2010 has a nested structure, which allows us to use more general categories (that is, only the first four digits of a code).

Job types are classified via the International Standard Classification of Occupations (ISCO-08)[25]. The system encodes job types with four digits, for example, code 2111 references 'physicists and astronomer' and code 5141 references 'barbers'. However, several codes have lengths exceeding 4, and, because ISCO-08 also has hierarchies, we can collapse those to four-digit codes.

The Labour Force Status provides information about a person's attachment to the labor market. The attachment does not solely include different forms of employment. For example, for a person enrolled in an

official higher-education program, the status would be 'student'. Being unemployed is also a type of attachment, even though the financial compensation is not a salary. Some labor-force statuses have additional information in the form of a modifier. If present, the modifier gives specifications for the labor-force status. If the labor-force status is student, the modifier might specify a 'foreign student'. A person can have multiple labor-force statuses in the same period of time. Using the student example again, a student can also have employment alongside studying, and both would be accounted for in the dataset.

Because we want to have a concept token representation of continuous variables, such as income and labor-force period, we discretize them based on quantiles. For example, the income variable is split into 100 categories. Another continuous variable is the labor-force period. It is a percentage of days in a month that the labor force status is relevant for (binned in ten categories). We also reserve concept tokens for each birth year and birth month.

Health data. The health data pertain to all ambulatory and inpatient contacts with hospitals in Denmark. The country has a publicly funded healthcare system that caters to all citizens. The data are encoded using the ICD-10 system[27], an internationally authorized World Health Organization system for classifying procedures and diseases. This system encompasses ~70,000 procedures and 69,000 diseases, each term represented by up to seven symbols. The first symbol denotes the chapter, which represents a specific type of diagnosis. The first three symbols combined provide the category. For example, code S86 is in chapter S, which stands for 'injuries and poisoning' and S86, combined, stands for the 'injury of muscle, fascia, and tendon at lower leg level'. By adding or removing symbols, one can control the specificity of the term.

To reduce the vocabulary size, we collapsed all codes to the category level, which resulted in 704 terms. The data include patient type, emergency status and urgency, in addition to diagnoses. Patient type denotes the admission type, that is, inpatient, outpatient or emergency. Emergency status indicates a patient admitted via an emergency care unit, and urgency specifies whether the cause of admission was an acute onset.

Preprocessing. Each health and labor record is translated into a sentence, where each associated attribute (for example, diagnosis, job type) is converted to a concept token. For example, if a labor record is connected to the job type 'Work with archiving and copying' (code 9210 in ISCO-08[25]), we convert it to POS_9210.

As a result, we have two types of sentence: labor sentences and health sentences. For each resident, we also create a background sentence that contains information about the birth month, birth year, country of origin status and sex (Supplementary Table 2).

Sentence and document structure. We assembled a chronological sequence of labor and health events for each resident $r \in \{1, 2, 3, \ldots, R\}$ in dataset $\mathcal{D}$. Each life-sequence has a form $S_r = \{s_r^0, s_r^1, s_r^2, \ldots, s_r^{n_r}\}$, where s_r^i is the ith life-event of the rth resident. Each event, s, contains tokens $v \in \mathcal{V}$ associated with a particular life-event, where $\mathcal{V}$ is a vocabulary of our artificial language. Along with the concept tokens, each event has associated temporal information such as absolute position, age and segment. $\mathcal{P}$ is a set of possible absolute temporal positions, where p is the number of days passed between event s and the origin point of 1 January 2008 (the day our dataset starts). If an event happened on 24 February 2012, then $p = 1{,}516$. $\mathcal{A}$ is a set of possible age values, where a specifies the number of full years passed since the person's birthday up until the date of the event, s. In terms of the life2vec model, p contextualizes events on a global timescale, whereas a contextualizes events on the individual timeline.

Finally, $\mathcal{G}$ is a set of segments. In the case where two or more events happen on the same day, all associated tokens share the same age and absolute position—essentially, the model cannot pinpoint where the token comes from. Segments allow additional differentiation between events. We have three distinct segments because it is highly unlikely that more than three events will be encountered simultaneously on the same day (in our dataset).

The segment assignment starts with A (each token of the first event is marked as segment A), the next event is marked B (even if this event happens on the next day), the next is marked C, the next is marked A, and so on. It ensures that (1) in the case where two or three events happen on the same day, each event has a different segment, (2) the number of segments A, B and C in a sequence is somewhat equal (otherwise, segment B only appears in days with two events and segment C only in days with three or more events).

The vocabulary set, $\mathcal{V}$, also includes several special tokens. For example, [CLS] starts a sequence and is later used to encapsulate a dense representation of the sequence. The [SEP] token stands between events, and [UNK] substitutes concept tokens that are not in our vocabulary (for example, tokens that were removed due to a low appearance frequency).

When we refer to the sentence length, $\|s\|$, we refer to the number of the corresponding concept token. The length of every sentence, s, varies depending on the type of event it describes. Health events range from two to three tokens, and labor events from three to seven concept tokens. Thus, the final length of the sequence, $\|S_r\|$, is a sum of the length of all the events, plus the number of special tokens such as [CLS] and [SEP].

The first sentence in the sequence, s_r^0, is a background sentence and it does not have an associated age or absolute time position, but it does have segment information.

The maximum length of the document is 2,560 concept tokens. In the rare cases (~1% of sequences) where the length of the document $\|S_r\|$ is above the specified limit, we remove earlier events (without removing a background sentence) until we can fit all the tokens of the last sentence (plus the last [SEP]). In the case where the length of the document is below the limit, we add padding tokens, [PAD], at the end of the sequence to fill up the empty spaces.

Data split. We randomly split the dataset (filtered according to initial requirements 1, 2, 3 and 4) into training, validation and test sets in the ratio 70:15:15. This random split is independent of any features of the sequence (entirely at random). The global training set had 2,276,460 people, the global validation set 487,812 people and the global test set 487,812 people.

Data augmentation. We introduced several data augmentation strategies to stabilize the performance of life2vec. These strategies alter sequences before a model sees them during the training stage and help to boost the performance of life2vec and baseline models. The augmentation techniques include subsampling sentences and tokens, adding noise to the temporal information, and masking the background sentence (Supplementary Information section 4).

Model architecture

The model consists of three components: an embedding layer, encoders[20] and task-specific decoders. The encoder is a transformer-based model, and the decoders are fully connected neural networks.

Inputs and embedding layer. The embedding layer transforms the raw life-sequence into the format that life2vec can process. Given a sequence S_p, we look up representations of tokens in the embedding matrix $\mathcal{E}_\mathcal{V} : \mathcal{V} \to \mathbb{R}^d$, where each row of $\mathcal{E}_\mathcal{V}$ corresponds to a token in the vocabulary (d is the number of hidden dimensions). Additionally, we look up the segment embedding in the $\mathcal{E}_\mathcal{G} : \mathcal{G} \to \mathbb{R}^d$ matrix. Both $\mathcal{E}_\mathcal{V}$ and $\mathcal{E}_\mathcal{G}$ matrices are optimized during the model training. To improve the representation of rare concept tokens and the overall isotropy of the

concept embedding space[65], we remove the global mean from each row of the $\mathcal{E}_\mathcal{V}$ matrix[65]. That is, each time we look up the token embedding, we subtract the mean.

We used Time2Vec[32] to model the linear and periodic progression of both age and absolute time positions. This introduces two learnable parameters, $\boldsymbol{\omega}$ and $\boldsymbol{\varphi}$, which determine the frequency and phase of periodic functions. The dense representations of age and position are calculated with the following equation, where z specifies the number of dimensions. We initialize two separate sets of Time2Vec parameters—one for the age, $\mathcal{T}_\mathcal{A} : \mathcal{A} \to \mathbb{R}^d$, and one for the absolute time position, $\mathcal{T}_\mathcal{P} : \mathcal{P} \to \mathbb{R}^d$. In both cases, we use the cosine function:

$$\mathcal{T}(x)[z] = \begin{cases} \boldsymbol{\omega}_z x + \boldsymbol{\varphi}_z, & \text{if } z = 0 \\ \cos(\boldsymbol{\omega}_z x + \boldsymbol{\varphi}_z), & \text{if } 1 \le i \le k \end{cases}$$

The temporal representation of a sentence, s_r, is calculated according to equation (1). Scalars α, β and γ are trainable parameters[33] initialized at a zero value:

$$\mathcal{E}_{\text{temp}}(s_r) = \alpha \times \mathcal{T}_\mathcal{A}(a) + \beta \times \mathcal{T}_\mathcal{P}(p) + \gamma \times \mathcal{E}_\mathcal{G}(g) \quad (1)$$

For each token v in s, we sum the associated token embedding in $\mathcal{E}_\mathcal{V}(v)$ and the temporal embedding of the sentence, $\mathcal{E}_{\text{temp}}(s_r^i)$. The input to the life2vec model is a concatenated sequence of these token representations (that is, a multidimensional tensor).

Encoder component. Like the original BERT[20], life2vec consists of multiple encoder blocks. Each block processes input representations and passes the results to the next encoder (or decoder). The architecture of each block is identical and consists of multi-head attention, a position-wise layer, and two residual connections (Supplementary Fig. 2).

The multi-head attention module consists of several attention heads, which separately process the input representations. The original BERT[20] uses softmax self-attention heads. Each head takes input representations and transforms these with several dense layers—query, key and value. These layers output linearly transformed representations $Q, K, V \in \mathbb{R}^{L \times d}$, where L is the length of the sequence and d is the dimensionality of embeddings. The contextualized representations are computed as (note that $\mathbf{1}_L$ is a vector of ones with length L)

$$\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^\text{T}}{\sqrt{d}}\right)V \iff D^{-1}AV, \quad (2)$$

$$\text{where } A = \exp\left(\frac{QK^\text{T}}{\sqrt{d}}\right), D = \text{diag}\left(A\mathbf{1}_L\right) \quad (3)$$

Softmax attention is suboptimal for sequences of length more than 512 tokens[66]. Therefore, we use softmax attention heads only to model local interactions; that is, we limit the span of these heads to 38 neighboring tokens.

To capture global interactions, we use performer-style attention heads[21], as they can handle longer sequences. Instead of calculating the precise attention matrix $A \in \mathbb{R}^{L \times L}$, performer-heads approximate it via matrix factorization. Entries of the approximated attention matrix are computed using kernels $A'(i, j) = K(\mathbf{q}_i^\text{T}, \mathbf{k}_j^\text{T})$ (indexes stand for the rows of matrices). The kernel function is defined as $K(\mathbf{x}, \mathbf{y}) = \mathbb{E}[\phi(\mathbf{x})^\text{T}, \phi(\mathbf{y})]$, where $\phi(\mathbf{u})$ is a random feature map that projects input into the r-dimensional space. Random mapping ϕ is constrained to contain features that are positive and exactly orthogonal (for details, see ref. 21). If we apply ϕ to Q, K, we get $Q', K' \in \mathbb{R}^{L \times r}$, where $r \ll L$. The attention is now defined as

$$\overline{\text{Att}}(Q, K, V) = \hat{D}^{-1}(Q'(K'^\text{T}V)), \text{ where } \hat{D} = \text{diag}\left(Q'(K'\mathbf{1}_L)\right) \quad (4)$$

Each multi-head attention module of the life2vec has four performer-style attention heads and four softmax attention heads (Supplementary Fig. 9). The output of these heads is concatenated and transformed with one more dense layer.

The encoder blocks also have a position-wise feed-forward module (PFF). This consists of two fully connected feed-forward layers that apply additional nonlinear transformations to each representation: $f_{\text{PFF}}(\mathbf{x}) = \text{swish}(\mathbf{x}W_1 + \mathbf{b}_1)\, W_2 + \mathbf{b}_2$, where $\text{swish}(\mathbf{x}) = \mathbf{x} \cdot \text{sigmoid}(\mathbf{x})$ (ref. 34).

Typically, the output representations of each module add up to the input representations via so-called residual connections: $\mathbf{y} = \mathbf{x} + f(\mathbf{x})$ (ref. 20), where f is a multi-head attention module or a position-wise feed-forward module. In our work we use ReZero connections[33], which consist of a single scalar, α. This scalar controls the fraction of information that each layer contributes to the contextualized representations: $\mathbf{y} = \mathbf{x} + \alpha \cdot f(\mathbf{x})$. At the start of training, each α is initialized to zero (meaning none of the encoder layers contribute at the beginning). We introduced several modifications to the BERT architecture, such as ReZero[33], ScaleNorm[35], Swish[34] and Weight Tying[36] to speed up the convergence and reduce the size of the model.

Training procedure

We split the training procedure into two stages: learning the overall structure of the data (pretraining) and performing task-specific inference (finetuning).

Pretraining—learning the structure of the data. We pretrain life2vec by simultaneously using masked language modeling (MLM) and sequence ordering prediction (SOP) tasks[13,20]. The pretraining creates a concept *space* and optimizes the parameters of the model. We perform the hyperparameter optimization to find the optimal values for the number of global and local attention heads, the number of encoder blocks, the hidden size, the size of the local window (for the local attention), the number of random features (in the global attention heads) and the size of the PFF layer (Supplementary Table 8).

The masked language modeling task forces the model to learn relations between concept tokens. We randomly choose 30% of the tokens in the input sequence[67], then 80% of the chosen tokens are substituted with [MASK], 10% are unchanged, and 10% are substituted with random tokens[20]. We do not mask any special tokens such as [CLS], [SEP], [PAD] or [UNK] (nor do we use them as random tokens). We use altered sequences as inputs to life2vec. Using the contextual output representations of tokens, the model should infer the masked tokens.

The MLM decoder consists of two fully connected layers (f_1 and f_2). Each contextual representation, $\mathbf{x}_i$, is transformed via $f_1(\mathbf{x}) = \tanh(\mathbf{x}W_1 + \mathbf{b}_1)$, followed by l2-normalization, $\text{norm}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|$. The weights of the final layer, f_2, are tied to the embedding matrix, $\mathcal{E}_\mathcal{V}$, which is further normalized to preserve only directions[36]. The resulting scores are scaled by α to sharpen the distribution[35]:

$$\text{MLM}(\mathbf{x}) = \alpha \times f_2\left(\text{norm}(f_1(\mathbf{x}))\right) \quad (5)$$

For each masked token the model must uncover, the decoder returns the likelihood distribution over the entire vocabulary. The likelihood (in our case) is a product of the scaled cosine distance between the contextualized representation of a token and the original representations of tokens in $\mathcal{E}_\mathcal{V}$ (ref. 36).

The sequence order prediction task forces the model to consider the progression of a life-sequence. It is an adapted version of the next sentence prediction task[13]. Each life-event in the sequence has four attributes: concept tokens, segments, absolute time position and age. In 10% of cases, we exchange concept tokens of one life-event with the concept tokens of another life-event (while preserving the positional and temporal information). In half of these cases, the exchange reverses the sequence so that the first life-event exchanges tokens with the last

life-event, the second life-event exchanges tokens with the second-to-last event, and so on. In the other half, we randomly pick pairs of life-events to exchange the concept tokens.

The SOP decoder pulls the contextual representation of the [CLS] token from the last encoder layer and passes it through two feed-forward layers to make a final prediction:

$$\mathrm{SOP}(\mathbf{x}) = \mathrm{ScaleNorm}\left[\mathrm{swish}(\mathbf{x}W_1 + \mathbf{b}_1)\right]W_2 + \mathbf{b}_2 \quad (6)$$

Task-specific finetuning

In this step, life2vec learns person-summaries conditional on the classification task; the model identifies and compresses patterns that maximize the certainty around a given downstream task[68]. To do so, we initialize the model with the parameters from the pretraining stage, assign a new task, and initialize a new decoder block (plus, remove MLM and SOP decoders).

We use pretrained life2vec in two settings: 'early mortality prediction' and 'personality nuances prediction task'. In both cases, life2vec pools the contextualized representation of each token in the sequence (that is, the output of the last encoder layer) and uses a weighted average of these to generate person-summaries. These summaries are later used to make predictions (Supplementary Fig. 2).

The weights of the encoder blocks are updated during the finetuning. However, deeper encoders have a lower learning rate to avoid 'catastrophic forgetting'[69]. We also freeze the parameters of $\mathcal{E}_{\mathcal{V}}$, except for the parameters associated with the [CLS], [SEP] and [UNK] tokens.

Early mortality prediction. Early mortality prediction is a binary classification task. The goal is to infer the mortality likelihood within the next four years after 1 January 2016 (that is, labels are 'alive' and 'deceased').

Optimization details. The crucial aspect of the mortality prediction is the loss function. The data we use (Dataset section) include people who might have left the country or disappeared before the end of 2020. Hence, we have a handful of right-censored outcomes. Using a cross-entropy loss would bias the predictions as we do not know the true outcome of all the sequences. Thus, we view the task as a positive-unlabeled learning[41] problem. We assume that all negative samples and samples with missing labels make up the unlabeled set, while all positive samples make a positive-labeled set (Supplementary sections 2 and 3).

Optimization metric. In the PU-Learning setting, we use the area-under-the-lift (AUL) to determine the end of finetuning as suggested in ref. 40. AUL can be interpreted as the 'probability of correctly ranking a random positive sample versus a random negative sample'[70].

Evaluation metric. We cannot use standard metrics to evaluate the model without introducing a bias[43], instead we apply the C-MCC (see ref. 43 for details) and use bootstrapping to estimate the 95% confidence intervals for C-MCC. We also provide values for AUL, corrected balanced accuracy score and corrected F1-score (Supplementary Table 3).

Baseline models. We use six baseline models, including majority class prediction, random guess, mortality tables, logistic regression, feed-forward neural network and RNN[39,71] to compare the performance of the early mortality task. For several models, we perform a hyperparameter optimization similar to the one we have done for the life2vec model (Supplementary Tables 9 and 10).

- Logistic regression is a generalized linear regression model. We optimize it using asymmetrical cross-entropy loss[41] with the ridge penalty and stochastic gradient descent. As an input to the model, we use a counts vector, that is, the number of times each token appears in a sequence over a one-year interval.
- Life tables is a logistic regression model that uses only age and sex as covariates.
- A feed-forward network uses the above-mentioned counts vector and has multiple feed-forward layers stacked over each other. It has a similar optimization setting as a logistic regression.
- An RNN model uses the same input as the life2vec model and same optimization settings. The RNN model outputs the contextual representation of each token, which we then pass through a decoder network (identical to the one in life2vec).

Personality nuances prediction task. The personality nuances prediction task is an ordinal classification task where labels correspond to the five levels of agreement with a particular item/statement. We predict the response to ten different items corresponding to the extraversion facet (Fig. 5):

1. I feel that I am an unpopular person,
2. I feel reasonably satisfied with myself overall,
3. I sometimes feel that I am a worthless person,
4. When I'm in a group of people, I'm often the one who speaks on behalf of the group,
5. In social situations, I'm usually the one who makes the first move,
6. I rarely express my opinions in group meetings,
7. The first thing that I always do in a new place is to make friends,
8. I prefer jobs that involve active social interaction to those that involve working alone,
9. Most people are more upbeat and dynamic than I generally am,
10. On most days, I feel cheerful and optimistic.

Questions 1–3 correspond to social self-esteem, 4–6 to social boldness (feeling comfortable in diverse social settings), 7–8 to sociability, or enjoyment of social interactions, and, finally, 9–10 evaluates liveliness (which includes enthusiasm and overall energy)[72].

Predicting agreement levels poses two technical issues. First, responses are unevenly distributed across possible answers, with a majority choosing non-extreme answers, and second, the level of agreement has an ordinal nature.

We therefore slightly modify the training procedure. To prevent overfitting to the majority class, we use instance difficulty-based resampling[73]—samples that are hard to predict would be subsampled more frequently (Supplementary Information section 3). To account for the ordinal and imbalanced nature of the data, we combine three loss functions[74]—class distance weighted cross-entropy[75], focal loss[76] with label smoothing penalty[77] (Supplementary Information section 2), and use a modified softmax function[37] and loss weighting[78].

For an optimization and evaluation metric We use Cohens's quadratic kappa (CQK) score to terminate the finetuning and evaluate the final performance[75].

Baseline models include a random guess that draws predictions from the uniform distribution (Supplementary Fig. 11), a random guess that draws predictions from the distribution of targets (that is, by permuting the actual targets) and the RNN model. Both life2vec and RNN use the same decoder architecture (Supplementary Fig. 2).

Interpretability and robustness

Here, we provide an overview of methods to interpret early mortality predictions as well as to evaluate the robustness of the concept space.

Interpretability of the early mortality predictions. Local interpretations. To provide the local interpretability, we use the gradient-based saliency score with L2-normalization[47,48]. The saliency score highlights the sensitivity of the output with respect to each input token; that is, the higher the sensitivity score, the more the output changes if we change the token representation (Supplementary Information section 5).

Global interpretations. Gradient-based saliency is unreliable when it comes to the global sensitivity of a model towards certain concepts. The person-summaries (provided by life2vec) form a complex

multidimensional space, and the dimensions of this space do not necessarily have human-interpretable meaning. Thus, we use TCAV[45] to estimate the overall sensitivity.

We define a high-level concept as a subsample of life-sequences that share specific attributes (such as 'individual has an F-diagnosis in the sequence'). We can take sequence representations of this subsample and train a linear classifier to discriminate between sequences in concept and random subsamples. The normal to the decision hyperplane is a concept direction. To calculate the TCAV scores, we rely on the procedure described in ref. 45 and Supplementary Information section 5. In Supplementary Tables 11 and 12 we provide an evaluation of the TCAV-based concept sensitivities.

Robustness of the concept space. Although the structure of the concept space (Fig. 2) seems reasonable under manual inspection, we provide further statistical proof for the robustness with the randomization test[79] and hubness test[65,80].

Randomization test. Here, we pretrained life2vec under different conditions by changing the random initialization seed or the training data.

After pretraining, we extracted the concept embeddings and calculated the cosine distances between every token. For every instance of life2vec, we ended up with a distance matrix $\mathcal{M}$. By following the procedure described in ref. 79, we can determine whether a pair of matrices ($\mathcal{M}_i$, $\mathcal{M}_j$) are correlated and hence prove that the concept space of two models share structure. The test includes the following steps:

1. Calculate Spearman's correlation $r_{\text{true}} = \text{corr}(\mathcal{M}_i, \mathcal{M}_j)$.
2. Permute rows and columns of ith matrix, and recalculate $r_p = \text{corr}(\mathcal{M}_i^p, \mathcal{M}_j)$.
3. Perform the second step 5,000 times.
4. Calculate the P value as

$$P = \frac{1}{5,000+1}\left(\sum I(r_P > r_{\text{true}}) + 1\right)$$

where I is an indicator function and equals one if the statement is true (and vice versa). We perform the continuity correction by adding 1 to the numerator and denominator.

5. We reject the null hypothesis if $P < 0.05$, thus confirming that the two matrices are correlated. If the experiment involves multiple comparisons, we use the Benjamini–Hochberg procedure.

Robustness with respect to initialization and sampling. We first applied the randomization test to check whether the random initialization and the samples that the model sees during the training lead to different concept spaces. We initialized three life2vec instances with different random initialization seeds and trained them on unique subsets of the original training data for ten epochs. After completing the pairwise comparisons, we rejected the null hypothesis with $P \approx 3.3 \times 10^{-4}$ in all cases (Supplementary Table 5).

Robustness with respect to training data. So far, we have only trained on data from 2008–2016 and studied those eight years of a cohort with ages in the range of 25–70 years. This choice might introduce biases. To better understand the implication of these choices, we implemented models for different age cohorts and trained on shorter time intervals (for example, 2008–2011). The randomization test also rejected the null hypothesis (Supplementary Table 6) in all the cases.

Hubness of the concept space. The embedding spaces produced by machine learning models often degenerate due to the presence of low-frequency tokens[65,80]. The model places most tokens along a similar direction, leading to less meaningful representations. The presence of hubs (tokens with an abnormal number of neighbors) is a proposed proxy for the degeneration of the embedding space[81].

To identify hubs in the embedding matrix, $\mathcal{E}_\mathcal{V}$, we found the five closest neighbors of each node based on cosine similarity and created a directed graph. Hubs can be identified by counting the incoming edges, which are the tokens with a large number of incoming edges. However, we did not find any hubs (that is, nodes with an abnormally large number of incoming connections). The [PAD] token has the highest number of incoming connections (that is, 49 links), [CLS] has 40 links, [SEP] 39 links, followed by [Female] (25) and [Male] (24)—the token with the most incoming edges is neighbor to less than 2% of tokens. Thus, we do not find proof of a degenerated concept space.

In summary, our evaluation shows that the concept space converges to a similar space structure for each subset of a dataset, and life2vec produces a robust representation of the synthetic language.

Statistics and reproducibility

This is a complex and multifaceted study, as is the overall study design. To support reproducibility, we provide an overview of the components of the study design below. The labor and health datasets are described in the Dataset section—these data are from the Danish National Registry, and no compensation is provided to participants (see Supplementary Information section 1 for more details). The POSAP[54] study participants (data used for the personality nuance prediction task) were offered automatic feedback on their scores in basic personality dimensions as well as the chance to win one of 15 electronic gift cards worth 5,000 DKK each (participation was voluntary).

In terms of statistical analyses, we did not use any methods to determine the effect size. To evaluate the robustness of the concept space (Supplementary Tables 5 and 6) we used the permutation test described in the Robustness of the concept space section.

To estimate the 95% confidence intervals of the C-MCC (early mortality prediction), C-MCC, corrected accuracy and corrected F1-score, we used stratified bootstrapping (Supplementary Tables 3, 4 and 7). The number of bootstrapped sets was 5,000 (each set had 100,000 samples that were randomly sampled with replacement). A detailed overview of the performance of the life2vec model for early mortality prediction is provided in Supplementary Information section 1. To estimate the uncertainty of the CQK (personality nuances prediction task), we used standard error. To compute TCAV scores, we used the procedure described in Supplementary Information section 5, and we estimated uncertainties via the bivariate midvariance. The finetuning of life2vec and the baseline models is described in Supplementary Tables 8–10. Additional information on the finetuning is provided in Supplementary Information sections 2 and 3. The data augmentation techniques are described in Supplementary Information section 4. An overview of notations used in the paper is provided in Supplementary Table 1.

Finally, we note that the experiments were not randomized, and the investigators were not blinded to allocation during experiments and outcome assessment. For more information, see https://github.com/SocialComplexityLab/life2vec for the set-up of the statistical analysis and the model.

The model, statistical tests and accompanying visualizations were developed in Python. The core packages were

1. bayesian-optimization 1.2
2. captum 0.5
3. coral-pytorch 1.4
4. cudatoolkit 11.6
5. dask 2022.9.1
6. focal-loss-pytorch 0.0.3
7. focal-loss-torch 0.1.0
8. h5py 3.7.0
9. hdf5 3.7.0
10. hydra-core 1.2.0
11. jupyterlab 3.4.7
12. matplotlib 3.6.0

13. numpy 1.22.3
14. pacmap 0.6.5
15. pandas 1.4.4
16. performer-pytorch 1.1.4 (customized, see https://github.com/SocialComplexityLab/life2vec/blob/main/src/transformer/performer.py)
17. pytorch 1.12.1
18. pytorch-lightning 1.7.6
19. scikit-learn 1.1.2
20. scikit-optimize 0.9.0
21. scipy 1.9.1.
22. seaborn 0.12.0
23. statsmodel 0.13.2
24. tensorboard 2.9.1
25. torchmetrics 0.10.0
26. umap 0.1.1

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this Article.

Data availability

The data used in this study are not publicly available due to Danish Data Protection regulations. Access to the data can be obtained via Statistics Denmark for Researchers in accordance with the rules of Statistics Denmark's Research Scheme: https://www.dst.dk/en/TilSalg/Forskningsservice/Dataadgang. Source data are provided with this paper.

Code availability

The source code for the data processing, life2vec training, statistical analysis and visualization is available on GitHub at https://github.com/SocialComplexityLab/life2vec (ref. 82). The model weights, experiment logs and associated model outputs can be obtained in accordance with the rules of Statistics Denmark's Research Scheme: https://www.dst.dk/en/TilSalg/Forskningsservice/Dataadgang.

References

1. Mansfield, L. A. et al. Predicting global patterns of long-term climate change from short-term simulations using machine learning. *NPJ Clim. Atmos. Sci.* **3**, 44 (2020).
2. Alali, Y., Harrou, F. & Sun, Y. A proficient approach to forecast COVID-19 spread via optimized dynamic machine learning models. *Sci. Rep.* **12**, 2467 (2022).
3. Zuboff, S. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power* (PublicAffairs, 2019).
4. Weber, M. *The Theory of Social and Economic Organization* (Simon & Schuster, 2009).
5. Salganik, M. J. et al. Measuring the predictability of life outcomes with a scientific mass collaboration. *Proc. Natl Acad. Sci. USA* **117**, 8398–8403 (2020).
6. Lynge, E., Sandegaard, J. L. & Rebolj, M. The Danish National Patient Register. *Scand. J. Public Health* **39**, 30–33 (2011).
7. Pedersen, C. B. The Danish civil registration system. *Scand. J. Public Health* **39**, 22–25 (2011).
8. Salganik, M. J. *Bit by Bit: Social Research in the Digital Age* (Princeton Univ. Press, 2019).
9. Grimmer, J., Roberts, M. E. & Stewart, B. M. *Text as Data: A New Framework for Machine Learning and the Social Sciences* (Princeton Univ. Press, 2022).
10. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), (ed. O'Conner L.) 770–778 (IEEE, 2016).
11. Silver, D. et al. Mastering the game of go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
12. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
13. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 5999–6009 (2017).
14. Brown, T. et al. Language models are few-shot learners. *Proc. NeurIPS* **33**, 1877–1901 (2020).
15. Grechishnikova, D. Transformer neural network for protein-specific de novo drug generation as a machine translation problem. *Sci. Rep.* **11**, 321 (2021).
16. Li, Y. et al. BEHRT: transformer for electronic health records. *Sci. Rep.* **10**, 7155 (2020).
17. Bojesomo, A., Al-Marzouqi, H. & Liatsis, P. Spatiotemporal vision transformer for short time weather forecasting. In *Proc. 2021 IEEE International Conference on Big Data* (*Big Data*) (eds. Chen Y. et al.) 5741–5746 (IEEE, 2021).
18. Huang, C.-Z. A. et al. Music transformer: generating music with long-term structure. Preprint at https://openreview.net/forum?id=rJe4ShAcF7 (2023).
19. Vafa, K. et al. CAREER: Economic prediction of labor sequence data under distribution shift. In *NeurIPS 2022 Workshop DistShift Spotlight* (2022).
20. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *Proc. NAANCL Hum. Lang. Tech.* **1**, 4171–4186 (2019).
21. Choromanski, K. M. et al. Rethinking attention with performers. Preprint at https://openreview.net/forum?id=Ua6zuk0WRH (2023).
22. Kozlowski, A. C., Taddy, M. & Evans, J. A. The geometry of culture: analyzing the meanings of class through word embeddings. *Am. Sociol. Rev.* **84**, 905–949 (2019).
23. Pilehvar, M. T. & Camacho-Collados, J. Embeddings in natural language processing: theory and advances in vector representations of meaning. *Synth. Lect. Hum. Lang. Technol.* **13**, 1–175 (2020).
24. *Arbejdsmarkedsregnskab* (Danmarks Statistik, 2022); https://www.dst.dk/da/Statistik/emner/arbejde-og-indkomst/befolkningens-arbejdsmarkedsstatus/arbejdsmarkedsregnskab
25. *International Standard Classification of Occupations: ISCO-08* (International Labour Office, 2012).
26. *Dansk Branchekode 2007: DB07* (*Danish Industrial Classification of All Economic Activities 2007*) v3 edn (Danmarks Statistik, 2015).
27. *International Classification of Diseases, 10th Revision* (*ICD-10*) (World Health Organization, 1994).
28. Yadav, P., Steinbach, M., Kumar, V. & Simon, G. Mining electronic health records (EHRS) a survey. *ACM Comput. Surv.* **50**, 1–40 (2018).
29. Han, Z., Zhao, J., Leung, H., Ma, K. F. & Wang, W. A review of deep learning models for time series prediction. *IEEE Sens. J.* **21**, 7833–7848 (2019).
30. Moncada-Torres, A., van Maaren, M. C., Hendriks, M. P., Siesling, S. & Geleijnse, G. Explainable machine learning can outperform Cox regression predictions and provide insights in breast cancer survival. *Sci. Rep.* **11**, 6968 (2021).
31. Rogers, A., Kovaleva, O. & Rumshisky, A. A primer in BERTology: what we know about how BERT works. *Trans. Assoc. Comput. Ling.* **8**, 842–866 (2021).
32. Kazemi, S. M. et al. Time2Vec: learning a vector representation of time. Preprint at https://openreview.net/forum?id=rklklCVYvB (2023).
33. Bachlechner, T., Majumder, B. P., Mao, H., Cottrell, G. & McAuley, J. ReZero is all you need: fast convergence at large depth. *Proc. Conf. Uncertainty Artif. Intell.* **161**, 1352–1361 (2021).
34. Ramachandran, P., Zoph, B. & Le, Q. V. Searching for activation functions. Preprint at https://openreview.net/forum?id=SkBYYyZRZ (2023).

35. Nguyen, T. Q. & Salazar, J. Transformers without tears: improving the normalization of self-attention. *Proc. 16th International Conference on Spoken Language Translation* (eds Niehues, J. et al.) 2019.iwslt-1.17 (ACL, 2019).
36. Pappas, N., Miculicich, L. & Henderson, J. Beyond weight tying: learning joint input-output embeddings for neural machine translation. *Proc. Third Conference on Machine Translation* (eds Borar, O. et al.) W18-6308 (ACL, 2018).
37. Kanai, S., Fujiwara, Y., Yamanaka, Y. & Adachi, S. Sigsoftmax: reanalysis of the softmax bottleneck. *Proc. NeurIPS* (eds Bengio S. et al.) **31**, 286–296 (2018).
38. Wang, Y., Huang, H., Rudin, C. & Shaposhnik, Y. Understanding how dimension reduction tools work: an empirical approach to deciphering t-SNE, UMAP, TriMAP and PaCMAP for data visualization. *JMLR* **22**, 9129–9201 (2021).
39. Naemi, A. et al. Machine learning techniques for mortality prediction in emergency departments: a systematic review. *BMJ Open* **11**, e052663 (2021).
40. Jiang, L., Li, D., Wang, Q., Wang, S. & Wang, S. Improving positive unlabeled learning: practical AUL estimation and new training method for extremely imbalanced data sets. Preprint at https://arxiv.org/abs/2004.09820 (2020).
41. Wang, C., Pu, J., Xu, Z. & Zhang, J. Asymmetric loss for positive-unlabeled learning. In *Proc. 2021 IEEE International Conference on Multimedia and Expo* (*ICME*) 1–6 (IEEE, 2021).
42. Hansen, A. V., Mortensen, L. H., Ekstrøm, C. T., Trompet, S. & Westendorp, R. Predicting mortality and visualizing health care spending by predicted mortality in Danes over age 65. *Sci. Rep.* **13**, 1203 (2023).
43. Ramola, R., Jain, S. & Radivojac, P. Estimating classification accuracy in positive-unlabeled learning: characterization and correction strategies. *Pac. Symp. Biocomput.* **24**, 124–135 (2019).
44. Geifman, Y. & El-Yaniv, R. Selective classification for deep neural networks. In *Proc Advances in Neural Information Processing Systems* (eds Guyon, I et al.) 30 (Curran Associates, 2017).
45. Kim, B. et al. Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). *Proc. ICML* **30**, 2668–2677 (2018).
46. Narayan, A., Berger, B. & Cho, H. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nat. Biotechnol.* **39**, 765–774 (2021).
47. Atanasova, P., Simonsen, J. G., Lioma, C. & Augenstein, I. A diagnostic study of explainability techniques for text classification. In *Proc. 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*) (eds Webber, B. et al.) 3256–3274 (ACL, 2020).
48. Bastings, J. & Filippova, K. The elephant in the interpretability room: why use attention as explanation when we have saliency methods? In *Proc. Third Blackbox NLP Workshop on Analyzing and Interpreting Neural Networks for NLP* (eds. Alishashi A. et al.) 149–155 (ACL, 2020).
49. Bommasani, R. et al. On the opportunities and risks of foundation models. Preprint at https://arxiv.org/abs/2108.07258 (2021).
50. Roberts, B. W., Kuncel, N. R., Shiner, R., Caspi, A. & Goldberg, L. R. The power of personality: the comparative validity of personality traits, socioeconomic status, and cognitive ability for predicting important life outcomes. *Perspect. Psychol. Sci.* **2**, 313–345 (2007).
51. Stewart, R. D., Mõttus, R., Seeboth, A., Soto, C. J. & Johnson, W. The finer details? The predictability of life outcomes from Big Five domains, facets and nuances. *J. Pers.* **90**, 167–182 (2022).
52. McCrae, R. R. & Costa, P. T. Jr. in *Handbook of Personality: Theory and Research* (eds John, O. P. & Robins, R. W.) 159–181 (Guilford Press, 2008).
53. Zettler, I., Thielmann, I., Hilbig, B. E. & Moshagen, M. The nomological net of the HEXACO model of personality: a large-scale meta-analytic investigation. *Perspect. Psychol. Sci.* **15**, 723–760 (2020).
54. *Det Danske Personligheds Og Sociale Adfærdspanel* https://copsy.dk/posap/ (accessed 21 March 2021).
55. Gangl, M. Changing labour markets and early career outcomes: labour market entry in Europe over the past decade. *Work Employ. Soc.* **16**, 67–90 (2002).
56. Halleröd, B., Ekbrand, H. & Bengtsson, M. In-work poverty and labour market trajectories: poverty risks among the working population in 22 European countries. *J. Eur. Public Policy* **25**, 473–488 (2015).
57. Mackenbach, J. P. et al. Socioeconomic inequalities in health in 22 European countries. *N. Engl. J. Med.* **358**, 2468–2481 (2008).
58. Adler, N. E. & Ostrove, J. M. Socioeconomic status and health: what we know and what we don't. *Ann. N. Y. Acad. Sci.* **896**, 3–15 (1999).
59. Liao, T. F. et al. Sequence analysis: its past, present and future. *Soc. Sci. Res.* **107**, 102772 (2022).
60. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (European Parliament & Council of the European Union); https://data.europa.eu/eli/reg/2016/679/oj
61. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K. & Galstyan, A. A survey on bias and fairness in machine learning. *ACM Comput. Surv.* **54**, 115 (2021).
62. Burkart, N. & Huber, M. F. A survey on the explainability of supervised machine learning. *J. Artif. Intell. Res.* **70**, 245–317 (2021).
63. Madiega, T. Artificial Intelligence Act (European Parliament, 2023); https://www.europarl.europa.eu/thinktank/en/document/EPRS_BRI(2021)698792
64. Eurostat. European system of accounts. ESA 2010 Publications Office of the European Union, 2013. *Off. J. Eur. Un.* **174**, 56 (2013).
65. Biś, D., Podkorytov, M. & Liu, X. Too much in common: shifting of embeddings in transformer language models and its implications. In *Proc. 2021 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies* (eds Toutanova, K. et al.) 5117–5130 (ACL, 2021).
66. Beltagy, I., Peters, M. E. & Cohan, A. Longformer: the long-document transformer. Preprint at https://arxiv.org/abs/2004.05150 (2020).
67. Wettig, A., Gao, T., Zhong, Z. & Chen, D. Should you mask 15% in masked language modeling? In *Proc. 17th Conference of the European Chapter of the Association for Computational Linguistics* (eds Vlachos, A. & Augenstein, I.) 2985–3000 (ACL, 2023).
68. Jawahar, G., Sagot, B. & Seddah, D. What does BERT learn about the structure of language? In *Proc. 57th Annual Meeting of the Association for Computational Linguistics* (eds Korhonen, A. et al.) 3651–3657 (ACL, 2019).
69. Sun, C., Qiu, X., Xu, Y. & Huang, X. How to fine-tune BERT for text classification? *Proc. CCl* **11856**, 194–206 (2019).
70. Huang, S., Wang, S., Li, D. & Jiang, L. AUL is a better optimization metric in PU learning. Preprint at https://openreview.net/forum?id=2NU7a9AHo-6 (2023).
71. Wilmoth, J. R. et al. in *Methods Protocol for the Human Mortality Database* 10–11 (Univ. California Berkeley and Max Planck Institute for Demographic Research, 2007).
72. Lee, K. & Ashton, M. C. Psychometric properties of the HEXACO personality inventory. *Multivariate Behav. Res.* **39**, 329–358 (2004).

73. Yu, S. et al. A re-balancing strategy for class-imbalanced classification based on instance difficulty. In *Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*) (ed. O'Conner L.) 70–79 (IEEE, 2022).
74. Müller, R., Kornblith, S. & Hinton, G. E. When does label smoothing help? In *Adv. Neural Information Processing Systems 32 (NeurIPS 2019)* (eds H. Wallach. et al.). **32**, 4694–4703 (Curran Associates, 2019).
75. Polat, G. et al. Class distance weighted cross-entropy loss for ulcerative colitis severity estimation. *Proc. MIUA* **13413**, 157–171 (2022).
76. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. *Proc. IEEE PAMI* **2**, 318–327 (2018).
77. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. In *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (ed. O'Conner L.)* (*CVPR*) 2818–2826 (IEEE, 2016).
78. Groenendijk, R., Karaoglu, S., Gevers, T. & Mensink, T. Multi-loss weighting with coefficient of variations. In *Proc. 2021 IEEE Winter Conference on Applications of Computer Vision* (*WACV*) 1468–1477 (IEEE, 2021).
79. Kriegeskorte, N., Mur, M. & Bandettini, P. A. Representational similarity analysis-connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* **2**, 4 (2008).
80. Liang, Y., Cao, R., Zheng, J., Ren, J. & Gao, L. Learning to remove: towards isotropic pre-trained BERT embedding. In *Proc. Artificial Neural Networks and Machine Learning – ICANN 2021*: *30th International Conference on Artificial Neural Networks* (eds. Farkaš I. et al.) 448–459 (ACM, 2021).
81. Mu, J., Bhat, S. & Viswanath, P. All-but-the-top: simple and effective postprocessing for word representations. Preprint at https://openreview.net/forum?id=HkuGJ3kCb (2023).
82. Savcisens, G. Socialcomplexitylab/life2vec. *Zenodo* https://doi.org/10.5281/zenodo.10118621 (2023).

Acknowledgements

We thank S. M. Hartmann for help with structuring and refactoring the code and M. F. Odgaard as well as the entire Social Complexity Lab for helpful feedback and discussions. The work was funded by the Villum Foundation Grant Nation-Scale Social Networks (to S.L.).

Author contributions

S.L. and G.S. conceived and designed the analysis. G.S. implemented the computational framework and performed the analysis, and T.E.-R., L.K.H., L.H.M., L.L., A.R., I.Z. and S.L. supported the analysis. T.E.-R., A.R. and L.K.H. contributed to the algorithmic auditing. A.R. contributed to the methodology of the transformer architecture. S.L., L.K.H. and L.H.M. refined the statistical evaluations and methodology. L.L. and I.Z. contributed data and refined the methodology for the personality nuances predictions. All authors read and approved the manuscript.

Competing interests

The authors declare no competing interests.

Inclusion and ethics statement

This study relies on secondary analysis of administrative data and does not require approval from the Danish committee system established under the Danish Act on Research Ethics Review of Health Research Projects. The data analysis was conducted in accordance with the rules set by the Danish Data Protection Agency and the information security and data confidentiality policies of Statistics Denmark. See Methods section Ethics and broader impacts, for further information.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-023-00573-5.

Correspondence and requests for materials should be addressed to Sune Lehmann.

Peer review information *Nature Computational Science* thanks Michal Kosinski, Denis Helic and Dashun Wang for their contribution to the peer review of this work. Primary Handling Editor: Fernando Chirigati, in collaboration with the *Nature Computational Science* team. Peer reviewer reports are available.

nature portfolio

Corresponding author(s): Sune Lehmann

Last updated by author(s): 12 June, 2023

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☐	☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☐	☒ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☐	☒ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	No software was used to collect the data.
Data analysis	Our code is based on Python 3.9.13. Here we list the core packages (we do not specify any dependencies): bayesian-optimization 1.2, captum 0.5, coral-pytorch 1.4, cudatoolkit 11.6, dacsk 2022.9.1 focal-loss-pytorch 0.0.3 focal-loss-torch 0.1.0 h5py 3.7.0, hdf5 3.7.0, hydra-core 1.2.0, jupyterlab 3.4.7, matplotlib 3.6.0, numpy 1.22.3, pacmap 0.6.5, pandas 1.4.4, performer-pytorch 1.1.4 (customized, see https://github.com/SocialComplexityLab/life2vec/blob/main/src/transformer/performer.py), pytorch 1.12.1, pytorch-lightning 1.7.6,

scikit-learn 1.1.2,
scikit-optimize 0.9.0,
scipy 1.9.1.,
seaborn 0.12.0,
statsmodel 0.13.2,
tensorboard 2.9.1,
torchmetrics 0.10.0,
umap 0.1.1,

Code can be found at https://github.com/SocialComplexityLab/life2vec/

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The data used in this study are not publicly available due to the Danish Data Protection regulations. Access to the data can be obtained via "Statistics Denmark for researchers" in accordance with the rules of Statistics Denmark's Research Scheme (https://www.dst.dk/en/TilSalg/Forskningsservice/Dataadgang).

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☐ Life sciences ☒ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Behavioural & social sciences study design

All studies must disclose on these points even when the disclosure is negative.

Study description	We applied a transformer model to detailed labor & health data to predict life outcomes for individuals. The study analyzes the embedding spaces to illuminate how and why the model makes predictions. All data are quantitative.
Research sample	The overall dataset is every Danish resident who was part of the labor market in the period 2008-2016, and in this sense the dataset is perfectly representative. For some experiments, we used age-limited subsets (we train on ages 25-70, and perform mortality prediction on ages 35-65). See methods/dataset for full details.
Sampling strategy	We have the full population available, but perform standard random splits for cross validation.
Data collection	We use official Danish national statistics. Data arise from official records (e.g health records, tax records, etc.).
Timing	Jan 2008 - Dec 2020. No gaps
Data exclusions	We exclude individuals who do not have consistent records of birth date, who do not have consistent sex throughout our study

Data exclusions	period, who have fewer than 12 labor records in 2015.
Non-participation	None.
Randomization	We split the population randomly into test and validation sets.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging