

Using sequences of life-events to predict human lives

In the format provided by the
authors and unedited

List of Supplementary Sections

1	Model Card: Early Mortality Prediction	3
2	Loss Functions	7
3	Resampling based on the Instance Difficulty	9
4	Data Augmentation	10
5	Interpretability	11

List of Supplementary Figures

1	Preprocessing a life-event to a sequence	13
2	Architecture of <code>life2vec</code>	14
3	Evaluation of the <code>life2vec</code> based on the year (early mortality)	15
4	Evaluation of the <code>life2vec</code> based on the sequence length (early mortality) . .	15
5	Evaluation of the <code>life2vec</code> based on the socio-demographic attributes (early mortality)	16
6	Evaluation of the <code>life2vec</code> based on the intersection of sex and the number of health events (early mortality)	17
7	Evaluation of the <code>life2vec</code> based on the intersection of sex and the age groups (early mortality)	18
8	Projections of Embedding Spaces with UMAP, DensMap and PaCMAP	19
9	Attention Mechanism (local and global)	20
10	Interoperability with attention and saliency	21

List of Supplementary Tables

1	Table of Notations	22
2	Specification of features and their sources.	23
3	Performance of <code>life2vec</code> and baselines (early mortality)	24

4	Vocabulary and robustness of the life2vec (early mortality)	24
5	Robustness of the concept space: dataset and random seed	25
6	Robustness of the concept space: data inclusion requirements	25
7	Performance of life2vec and baselines (emigration)	25
8	Hyperparameter optimization for the life2vec (pretraining)	26
9	Hyperparameter optimization for the RNN (early mortality)	27
10	Hyperparameter optimization for the neural network (early mortality)	28
11	TCAV Sensitivity scores for the "deceased" prediction	29
12	TCAV Sensitivity scores for the "alive" prediction	30

Supplementary Sections

1 Model Card: Early Mortality Prediction

The model card refers to **two** models[1]:

1. *life2vec (base)* model pre-trained on the unsupervised masked language modeling and sequence order tasks, and
2. *life2vec (mortality)* model finetuned on the Mortality Prediction Task.

1.1 Model Details

Person or organization developing model. *life2vec (base and mortality)* model is developed as part of the Nation-Scale Social Network Project. Specifically as part of the Ph.D. studies of Germans Savcicens at the Technical University of Denmark. The involved organizations are the Technical University of Denmark (Department of Applied Mathematics and Computer Science), Danmarks Statistik (Data Science Lab), and the University of Copenhagen (Copenhagen Center for Social Data Science).

Development period. September 2020 - October 2023.

Model version. v1 (*base*) and v1 (*mortality*).

Model type. Transformer-based deep neural network.

Methods

Architecture details of the *base* model:

- **Architecture:** BERT-like encoder network
- **Optimization tasks:** Masked Language Modeling (MLM) and Sequence Order Classification (SOC),
- **MLM Decoder Network:** Two-layer neural network (Supplementary Figure 2). The final layer is tied to the Concept token embedding matrix (Supplementary Figure 2),
- **SOC Decoder Network:** Two-layer neural network (Supplementary Figure 2),
- **Attention mechanism:** Performer-style attention heads, with local softmax-attention heads (Supplementary Figure 2 and 9),
- Other **architecture modifications** include ReZero & ScaleNorm normalization, Swish, Input-Output Embedding Tying, *time2vec*-based encoding of temporal information, sequence order prediction task, and cross-entropy loss with the label smoothing. These modifications ensure fast convergence and relatively small size of the model,
- **Optimization strategy:** AdamW Optimizer with the OneCycle LR Annealing. Trained

for 30 epochs, where one epoch covers 30,000 randomly sampled (and augmented) sequences from the training dataset.

Architecture details of the *mortality* model:

- **Architecture:** BERT-like encoder network (aka pre-trained *base* model),
- **Optimization task:** Binary mortality prediction task (early mortality within the next four years),
- **Classification decoder:** Two-layer network with the weighted averaging of concept representations (Supplementary Figure 2),
- Other **architecture modifications:** Asymmetric Cross-Entropy (Supplementary Section 2) for the Positive Unlabeled data sets and Sigsoftmax (as an alternative to Softmax),
- **Optimisation strategy:** RAdam optimizer with exponential LR annealing ($\gamma = 0.8$). Base LR for the decoder is 0.01, and LR for each consecutive encoder layer reduces by 5%. Token embeddings are frozen except for the [CLS], [SEP], [UNK] tokens. Temporal and Segment Embeddings are not frozen. For the decoder network, we set weight decay to 0.01. For the encoder layers, we set weight decay to 0.001,
- **Data Sampling:** We re-sample positive and negative samples to get approximately an equal fraction of both targets.

Fairness Constraints. None.

License: Not for public use or distribution.

Primary intended uses. The following information covers both **base** and **mortality** models:

- Use for scientific and research purposes only,
- Verify the validity of MLP inspired *Socio-economic* data representation,
- Verify the validity and performance of transformer-based architectures in the context of longitudinal socio-economic data
- Explore interactions between life-events and outcomes (i.e., mortality prediction) *on a global scale*.
- Use person-summaries as node-features in the Large-scale *Danish* population graphs
- Use person-summaries embeddings to study causal relationships between life-events.

Primary intended users. The following information covers both **base** and **mortality** models:

1. Denmark-based academics in Computational Social Science, Economics, Healthcare, Sociology, or Network Science,
2. Employees of Denmark Statistics (specifically Data Science Lab).

Out-of-scope use cases. The following information covers both **base** and **mortality** models:

1. **Not intended** as a tool to make judgments about specific individuals,
2. **Not intended** for a public release or deployment in governmental or private institu-

tions.

1.2 Factors

This section describes the factors (e.g., groups, socio-economic attributes, sequence structure, etc.) that might lead to discrepancies in the model performance. **The section covers the *mortality* model.**

Potential **relevant factors** are

1. **Level of interaction with the healthcare system** – the fact that people use and consult healthcare providers with different frequencies (e.g., a person avoids interaction with the healthcare system or interacts only in severe cases),
2. **Socio-demographic attributes**: age, sex, and residency status (different groups, e.g., immigrants, ex-pats, natural-born citizens, and other residents, might have various access to public services and various sets of opportunities and limitations).
3. **Sequence Length** – longer sequences might contain more information (that model can use). The results confirm that the level of interaction with the healthcare system *does have* an impact on the quality of predictions.
4. **Data drift and time** – we cannot guarantee the robust performance of the model beyond 2020 (e.g., COVID and human behavior),
5. **Cause of death.**

Evaluation factors. Since **age** and **sex** are highly correlated with mortality outcomes, we want to evaluate the model’s performance on unitary and intersectional splits of these groups (to probe the `life2vec`’s sensitivity to these features). Regarding the **residency status**, we are limited to the split based on the birthplace (i.e., in Denmark or outside). Thirdly, we want to evaluate how robust `life2vec` is regarding various life-sequences structures (aka length- and event-wise). Thus, we look at the **number of health-related events** in a sequence and the **length of the sequence**. Lastly, we do not have access to data beyond 2020. Thus, we cannot estimate the effect of the data drift on the `life2vec` model. However, we can evaluate how well the model predicts *distant* deaths.

1.3 Metric

Pre-training. We look at the perplexity score to evaluate and choose the most optimal `life2vec` (base) model (not presented in the model card).

Mortality Prediction. We frame the mortality prediction task as a positive-unlabeled problem. To optimize the `life2vec` (mortality) model, we use Area-Under-the-Lift (AUL), i.e., the early-stopping mechanism uses the AUL score. The primary performance evaluation

metric for the `life2vec` model is Corrected Mathew’s Correlation Coefficient (C-MCC) with a 95% Confidence Interval (we use bootstrapping). We use correction to account for the unlabeled samples in the test dataset. Along, we provide the Corrected Balanced Accuracy and Corrected F1-Score (refer to the Supplementary Table 3). All metric is reported at the .5 probability cutoff (not applicable to the AUL). All the metrics presented in this model card are based on the test subset.

Quantitative Analysis. We estimate the C-MCC score on the test data split (not the full one, but a semi-random subsample of 100,000 people). Fig. 3 shows the recall as a function of the year – the performance degrades as we get further away from the 31st of December 2015. Meanwhile, we provide detailed breakdowns of the performance per various groups:

- Unitary groups based on the **sequence length** in the Supplementary Figure 4: the length of the sequence does not seem to impact the performance of the model),
- Unitary groups based on the **country of origin**, **sex**, and **age group** in the Supplementary Figure 5. The age attribute: generally, older people have a higher probability of death, and the performance metric is worse for older people. The sex attribute: the model’s performance is similar regarding sex attributes. Residency attribute: a large difference between DK and Non-DK groups, which might be connected to the imbalanced representation of groups),
- Intersectional groups based on the **age group** and the **number of health events** in the Supplementary Figure 6,
- Intersectional groups based on the **age groups** and **sex** in the Supplementary Figure 7.

1.4 Data

Labor [2] and health data [3, 4] are provided by Danmarks Statistik (DST). These datasets include socio-economic, longitudinal information about the residents of Denmark. The use of the data [2, 3, 4] is regulated by (1) the EU Regulation on European Statistics, (2) the General Data Protection Regulation (GDPR), (3) the Danish Data Protection Act, (4) the Danish Public Administration Act, (5) the Danish Access to Public Administration Files Act, (6) the Danish Criminal Code, and (7) the Act on Statistics Denmark (DST). DST ensures that data of Danish residents and businesses is used **only for scientific purposes**.

Preprocessing. Refer to the Methods Section in the Original Paper.

Data split. We split data into training, validation, and test datasets (completely at random). **Training** subset is used to optimize the model. **Validation** subset is used to evaluate the model’s performance at a specific epoch - we terminate the model’s training if the performance metric on the validation data does not improve. **Test** subset estimates the final model performance.

Ethical Considerations. Refer to the Methods Section in the Manuscript.

2 Loss Functions

Based on the task, we use various loss functions to optimize the model.

Pretraining. Here we have two tasks – Masked Language Model (MLM) and Sequence Order Prediction (SOP). For the MLM, we use *cross-entropy loss*

$$\mathcal{L}_{CE}(\mathbf{y}, \mathbf{x}) = - \sum y_i \log(f(\mathbf{x})_i) \quad (1)$$

Here y is a one-hot encoded true target, and $f(x)$ is the output of the model. For the SOP task, we use weighted *cross-entropy loss with the label smoothing* [5]

$$\mathcal{L}_{CE-LS}(\mathbf{y}, \mathbf{x}) = (1 - \alpha) \times - \sum w_i y_i \log(f(\mathbf{x})_i) + \frac{\alpha}{n} \times \sum w_i \log(f(\mathbf{x})_i) \quad (2)$$

Here n is the number of classes, w_i is a weight assigned to class i , and α is scalar to control for a mixture of CE and Label Smoothing components (i.e., $w = [1.1, 10, 10]$, $n = 3$, and $\alpha = 0.1$). In Eq. 1, $\forall i w_i = 1$.

Mortality Prediction and Emigration Prediction. We use *Asymmetric Cross-Entropy Loss* [6]. It accounts for the fact that the unlabeled set might contain positive samples. It also does not require any knowledge of a prior for the frequency of positive samples in the unlabeled set.

$$\mathcal{L}_{ACE}(\mathbf{x}) = -\frac{1}{p} \sum_{i=1}^p \log(g(f(\mathbf{x}_i))_1) - \frac{1}{n-p} \sum_{i=p+1}^n \log\left(g(f(\mathbf{x}_i) + \begin{bmatrix} c \\ 0 \end{bmatrix}^T)_0\right) \quad (3)$$

In Eq. 3, $\mathbf{x}$ contains a batch of sequences, f is a model that outputs logits, $g(\cdot)_1$ and $g(\cdot)_0$ denote output (normalized scores) for positive and unlabeled classes (e.g. softmax function); while n is a total number of samples in a batch and p is the number of positive samples. $c \geq 0$ is a constant added to a logit of a unlabeled sample; we choose it by optimising AUL metric.

Predicting personality nuances. We use mixture of *Class Distance Weighted Cross-Entropy Loss* [7], *Focal Loss* [8] and *Label Smoothing* [5].

Class Distance Weighted Cross-Entropy Loss [7] handles imbalanced ordinal classification tasks. Instead of maximizing the likelihood of the true class, we minimize the likelihood of incorrect labels weighted by the absolute distance to the true class (Eq. 4, where y is a true label of a sample (*not* one-hot encoded). CDW-CE has one hyperparameter – α , a distance penalty, which we set to 1.5.

$$\mathcal{L}_{CDW-CE}(y, \mathbf{x}) = - \sum_{i=0}^{N-1} \log(1 - f(\mathbf{x})_i) \times |i - y|^\alpha \quad (4)$$

Focal Loss [8] is another version of cross-entropy loss that focuses on the samples that are hard to predict (Eq. 5), we set $\gamma = 5$. *Label Smoothing* (LS as in Eq. [9]) penalizes *overconfident* values (here $\mathbf{y}$ is a one-hot encoded target)

$$\mathcal{L}_F(\mathbf{y}, \mathbf{x}) = - \sum \log(\mathbf{y}_i - f(\mathbf{x})_i)^\gamma \log(f(\mathbf{x})_i) \quad (5)$$

To calculate $\mathcal{L}_i$ associated with the i th nuance, we use the weighted average of above mentioned losses, where the weights α_i , β_i and γ_i are learned during the finetuning (see Eq. 6) [10]. Without these modifications, the `life2vec` model converges to a majority prediction. The full loss is the average of all $\mathcal{L}_i$.

$$\mathcal{L}_i(y, \mathbf{x}) = \alpha_i \times \mathcal{L}_{CDW-CE}(y, \mathbf{x}) + \beta_i \times \mathcal{L}_F(\text{onehot}(y), \mathbf{x}) + \gamma_i \times \text{LS}(\text{onehot}(y), \mathbf{x}) \quad (6)$$

3 Resampling based on the Instance Difficulty

We use the resampling strategy [11] as one of the methods to account for a large imbalance of classes (Personality nuances task). This particular resampling method is described for a single target case, where the instance difficulty (or sampling rate) of an i th sample after t epochs is defined as [11] (where N is the number of training samples)

$$D_{i,T} = \frac{1}{N} + \sum_t d_{i,t}$$

In our case, we modify the calculation of the instance difficulties, $D_{i,T}$ due to following reasons: (1) we have a multi-target task (i.e., we have ten nuances), (2) the division of learning and unlearning rates ($d_{i,t} = \frac{dl_{i,t}}{du_{i,t}}$) leads to an unstable solution due to float overflow and underflow. First, we decompose the calculation of the i th sample at the t th epoch as (j stands for a difficulty associated with the j -th nuance)

$$d_{i,t} = \sum_j d_{i,t,j}$$

Since the difficulty score is described as the slope of the difficulty vector $\mathbf{d}_{i,t,j} = (dl_{i,t,j}, du_{i,t,j})$ [11], we use the cosine function to avoid under- and overflow (plus, use the power transformation to further up-weight the difficult instances). We calculate the values of $dl_{i,t,j}$ and $du_{i,t,j}$ according to Eq. 6, 7 in [11] followed by our modification

$$d_{i,t,j} = \left(1 - \frac{dl_{i,t,j}}{dl_{i,t,j}^2 + du_{i,t,j}^2}\right)^4$$

Lastly, we change the calculation of $D_{i,T}$ by introducing the exponential weighted average with $\alpha = .5$

$$D_{i,T} = \alpha \cdot d_{i,t} + (1 - \alpha) \cdot D_{i,T-1}$$

These operations help to stabilize the sampling weights (for our multi-target case).

4 Data Augmentation

We use data augmentation during the pretraining and fine-tuning stages. These techniques ensure robust performance and better generalizability of the model.

Sequence Downsampling. We randomly pick a life-sequence and randomly remove up to 50% of life-events.

Temporal Noise As labor-events mainly occur on the last day of a month, we smooth the distribution of the absolute time. We randomly pick a life-sequence and alter the absolute time by injecting noise, $\mathcal{U}(-5, 5)$, into each life-event.

Background Masking. We randomly pick a life-sequence and mask the background information (i.e., sex, origin, and birthday) with the [UNK] token.

Token Downsampling. We randomly pick a life-sequence and randomly remove tokens from the life-events. This procedure does not affect [CLS] and [SEP] tokens.

The augmentation procedures are independent of each other. Thus, some sequences might be altered by multiple procedures – the order of application: Sequence Downsampling, Temporal Noise, Background Masking, and Token Downsampling.

5 Interpretability

Attention Score When we use the Pooled Attention Decoder (see Supplementary Fig. 2, we can extract the attention weights associated with each concept token in a sequence.

Saliency score indicates the degree of change (and is directly connected to the partial derivatives). The contribution is calculated by back-propagating through the network, starting from the output score toward each token in the sequence. The higher the gradient associated with a token, the higher the contribution towards the predicted values since small changes in a particular concept token embedding would lead to a higher degree of change in the output. To achieve robust scores and minimize the noise associated with the gradient descent. x_i is an embedding of a token, $f(x_{1:n})$ is the output of the model, n is the number of noisy samples.

$$S(x_i) = \frac{1}{n} \sum_n \left\| \frac{\partial}{\partial x_i} f(x_{1:n}) x_i + e \right\|_2, \quad e \sim \mathcal{N}(0, \sigma^2)$$

Attention scores and Saliency Scores help us understand how much each token contributes to the final prediction. The example is shown in the Supplementary Figure 10.

Discovery of the concept directions via TCAV method. The following provides a step-by-step workflow to (1) find the direction of the concept and (2) evaluate these directions:

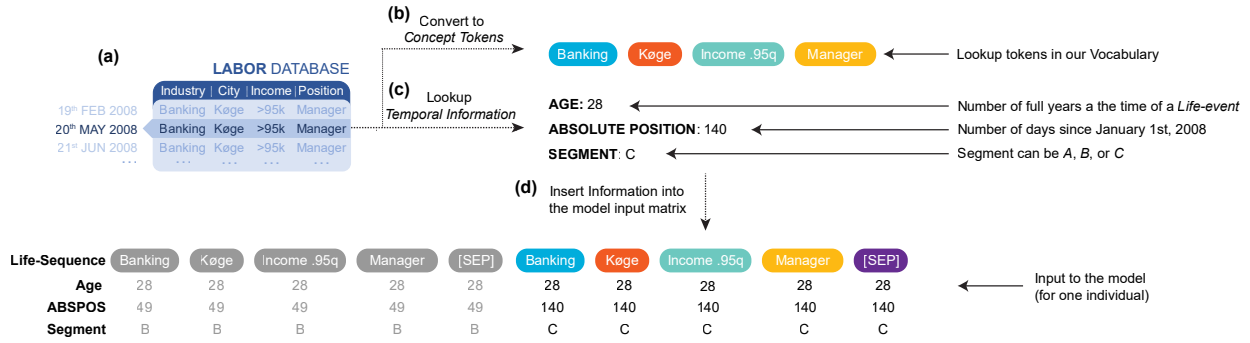
1. Specify a concept, $\mathcal{C}$, e.g. life-sequence contains at least one E16 diagnoses over year 2015,
2. Randomly sample 10 000 life-sequences, $\mathbf{s}$, from the *test* dataset: (1) for every life-sequence, $\mathbf{s}$, find a person-summary, $h(\mathbf{s}) = \mathbf{x}$ (where h is the encoder part of the model), and (2) calculate the gradient of output values, $\nabla f(\mathbf{x})$, with respect to the person-summary, $\mathbf{x}$, where f is a decoder part of the model that takes a person-summary, $\mathbf{x}$, as an input and outputs logits,
3. Randomly sample 3 000 life-sequences, $\mathbf{s}$, (from the *validation* dataset) that satisfy the specifications of a concept $\mathcal{C}$ and calculate the person-summary, $\mathbf{x}$ for each sample (we will refer to this set as $\mathcal{D}_{\mathcal{C}}$
4. Randomly sample 5 000 life-sequences (from the *validation* dataset) that do not satisfy the specifications of a concept, $\mathcal{C}$ and calculate the person-summary, $\mathbf{x}$ for each sample (we refer to this set as $\mathcal{D}_{-\mathcal{C}}$)
5. Using stratified 5-fold cross-validation, find the optimal l2-regularisation parameter for the logistic regression on $\mathcal{D}_{\mathcal{C}} \cup \mathcal{D}_{-\mathcal{C}}$ datasets. The task for the logistic regression is to predict whether the life-sequence satisfies the concept $\mathcal{C}$.
6. Train 1 000 logistic regressions (with the optimal l2-regularisation parameter found in previous step) on the bootstrapped od $\mathcal{D}_{\mathcal{C}} \cup \mathcal{D}_{-\mathcal{C}}$ datasets,
7. Find the orthonormal vector to each separating hyperplane found by the logistic re-

gression. These normals are concept activation directions, h_C .

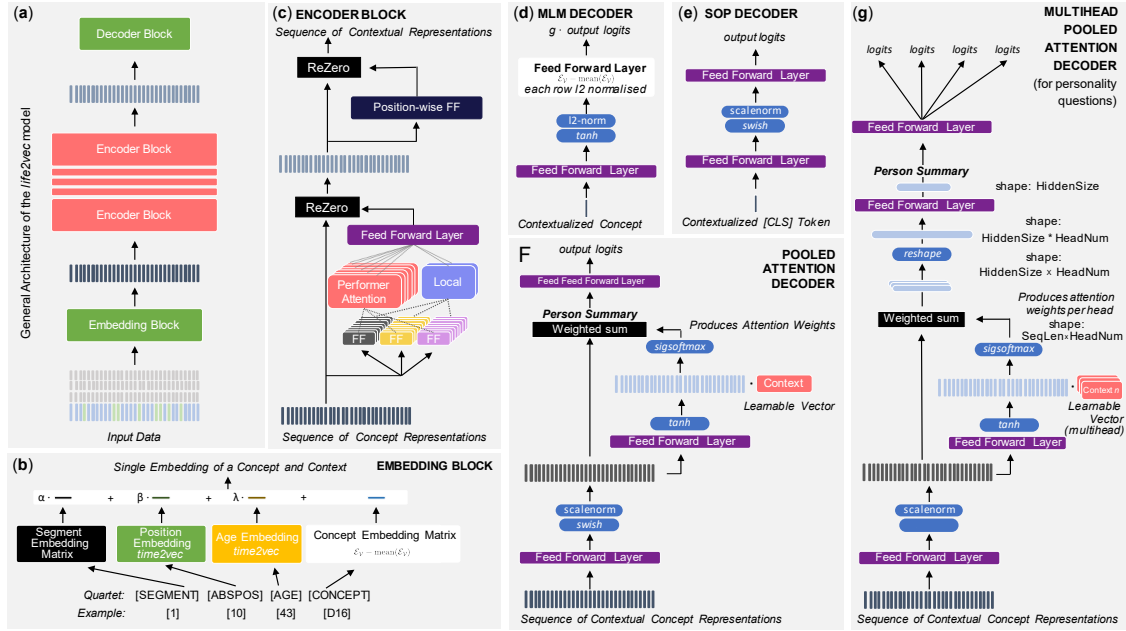
8. For each gradient vector, $\nabla f(\mathbf{x})$, (step 1) find a dot product between every concept activation direction, h_C (step 6). The average of these values is the sensitivity score of a model to a particular concept.

As a baseline, we specify a random concept (i.e., no specifications). Then, during steps 2 and 3, we randomly sample sequences. Then, we use the Wilcoxon test to compare the distribution of the scores of baseline and distribution of the scores of a particular concept, C (refer to Supplementary Tab. 11 and 12).

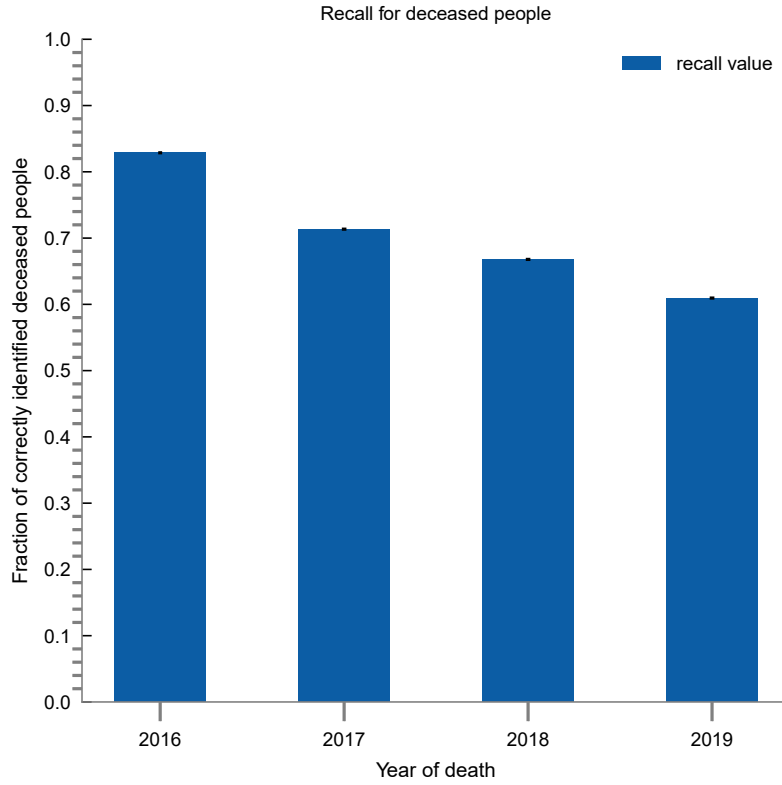
Supplementary Figures



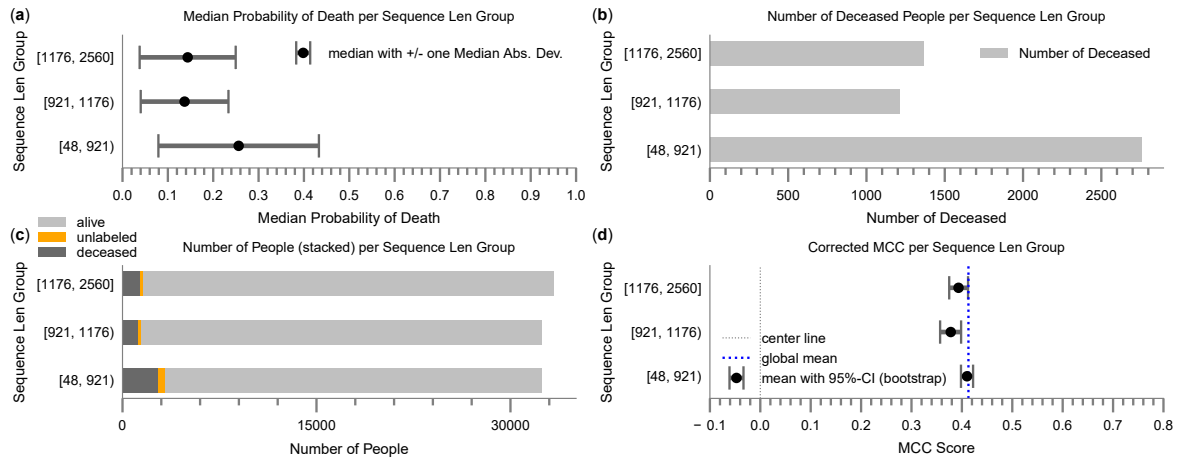
Supplementary Figure 1: The data transformation from the Tabular format to the `life2vec` suitable format. (a) We start by looking up the next (chronologically-wise) life-event in a person's history. (b) We convert relevant features to concept tokens of our vocabulary. (c) We look up relevant positional information such as age, absolute temporal position, and segment of the life-event. (d) We insert the data into the sequence.



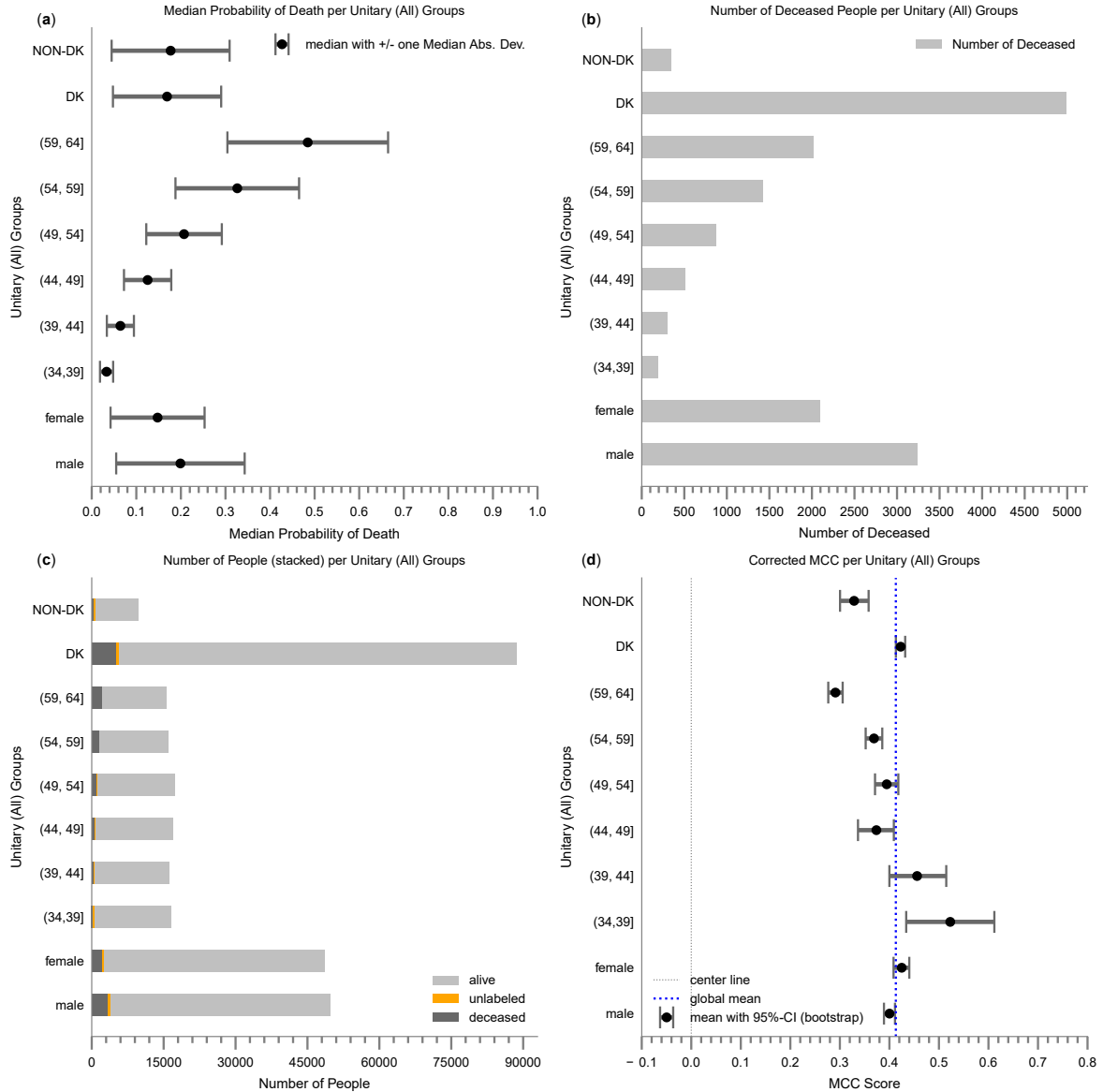
Supplementary Figure 2: Architecture of life2vec. (a) The overall structure of the model. The life-sequence is passed to the embedding block and every encoder block. The implementation of a decoder block depends on the task. Each block before the decoder outputs a contextualized representation of concept tokens. (b) Each concept token in a sequence (together with the positional information) passes through the embedding block. It merges positional and concept information. Age and Absolute Position embeddings are calculated via time2vec [12], while segment and concept embeddings are stored in a lookup matrix. To retrieve the concept embedding, we lookup the corresponding embedding in a matrix $\mathcal{E}_V$ and then remove the mean of $\mathcal{E}_V$ (without changing the representation in an actual $\mathcal{E}_V$). Positional representations are weighted and added to the representation of a concept. (c) The encoder block takes the life-sequence representation from the previous block. The sequence is passed through the attention layer (Multi-head Performer and Local Attention). The result of the attention layer is added to life-sequence representation via ReZero gate. Further, the life-sequence representation is passed through the Position-Wise Feed Forward Layer. The result is added back to the life-sequence representation via another ReZero gate. (d) Masked Language Model Decoder pools (separately) contextual representation of masked concept tokens. MLM Decoder and the Sequence Order Prediction Decoder (E) are used during the pretraining. The last feed-forward layer copies weights from the matrix $\mathcal{E}_V$. We remove the mean of $\mathcal{E}_V$ from each row and then apply the l2-norm. Thus, the output logits are calculated as a dot product between the contextual representation of a concept and each row of standardized $\mathcal{E}_V$ matrix [13]. (e) The Sequence Order Prediction decoder pools the contextual representation of [CLS] concept (which is always placed first in the sequence). (f) The pooled attention decoder is used during the finetuning. It pools the life-sequence from the last encoder block. After passing through the first feed-forward layer, it uses life-sequence representations to compute attention weights (right-hand side). It uses context (a trainable vector) to calculate the importance of a concept at a position i . Attention weights are then used to calculate the weighted average of concepts in the life-sequence, or *person-summary*. (g) A multihead version of the pooled attention decoder. This version has multiple context vectors instead of one. For each context vector, we get a set of attention weights - as we apply attention weights to the sequence, we get n sequence representations corresponding to n context vectors. We concatenate these representations and pass them through a feed-forward layer to get a *person-summary*. We use this decoder in the personality nuances prediction task, where we have ten context vectors (since we have ten questions).



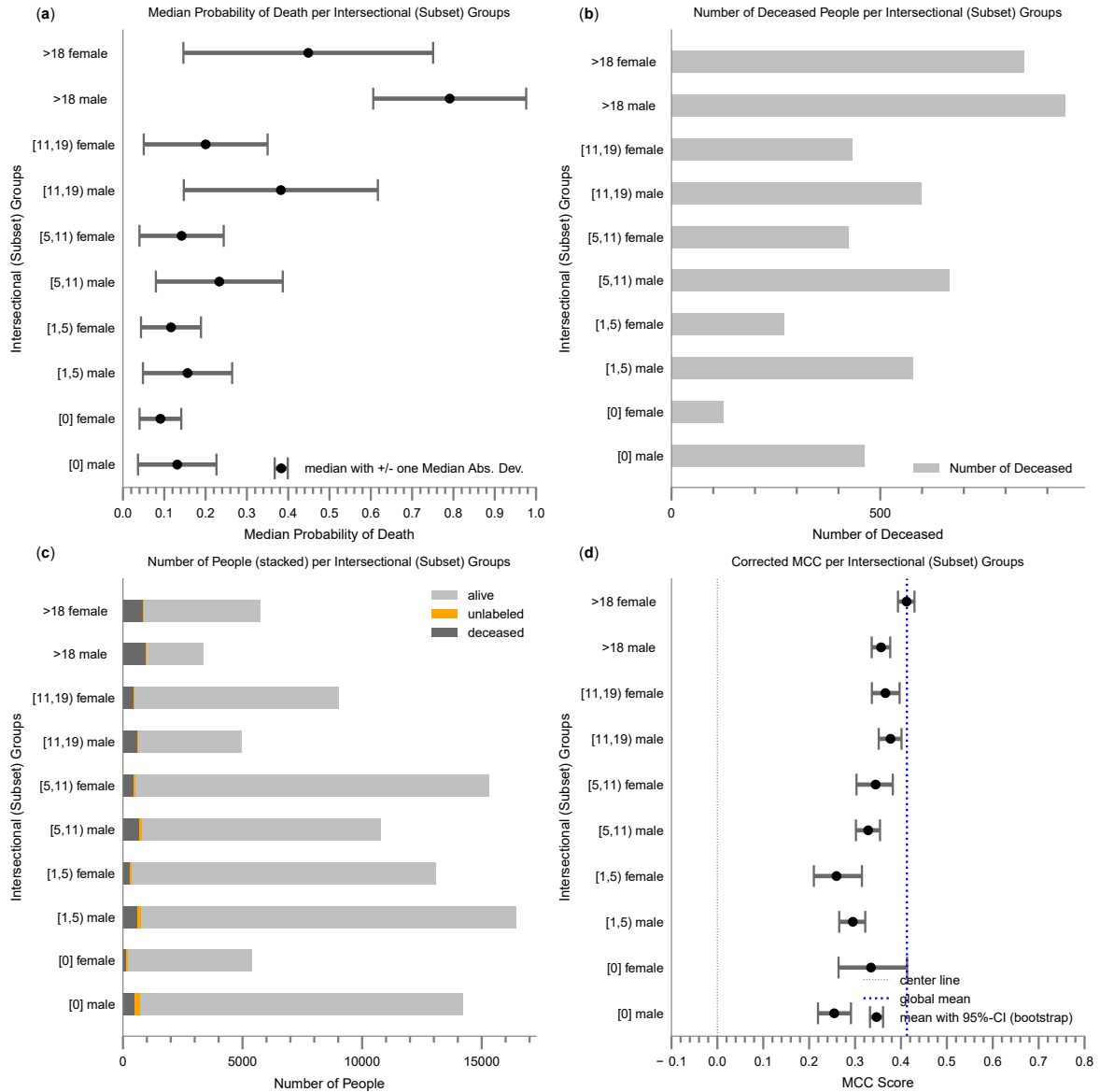
Supplementary Figure 3: The *life2vec*'s recall is based on the period between the day of the prediction and the day of death (yearly bins). The recall is the fraction between two values: the number of (known) true positives (deceased) divided by the total number of (known) deceased. The higher the value, the higher the performance.



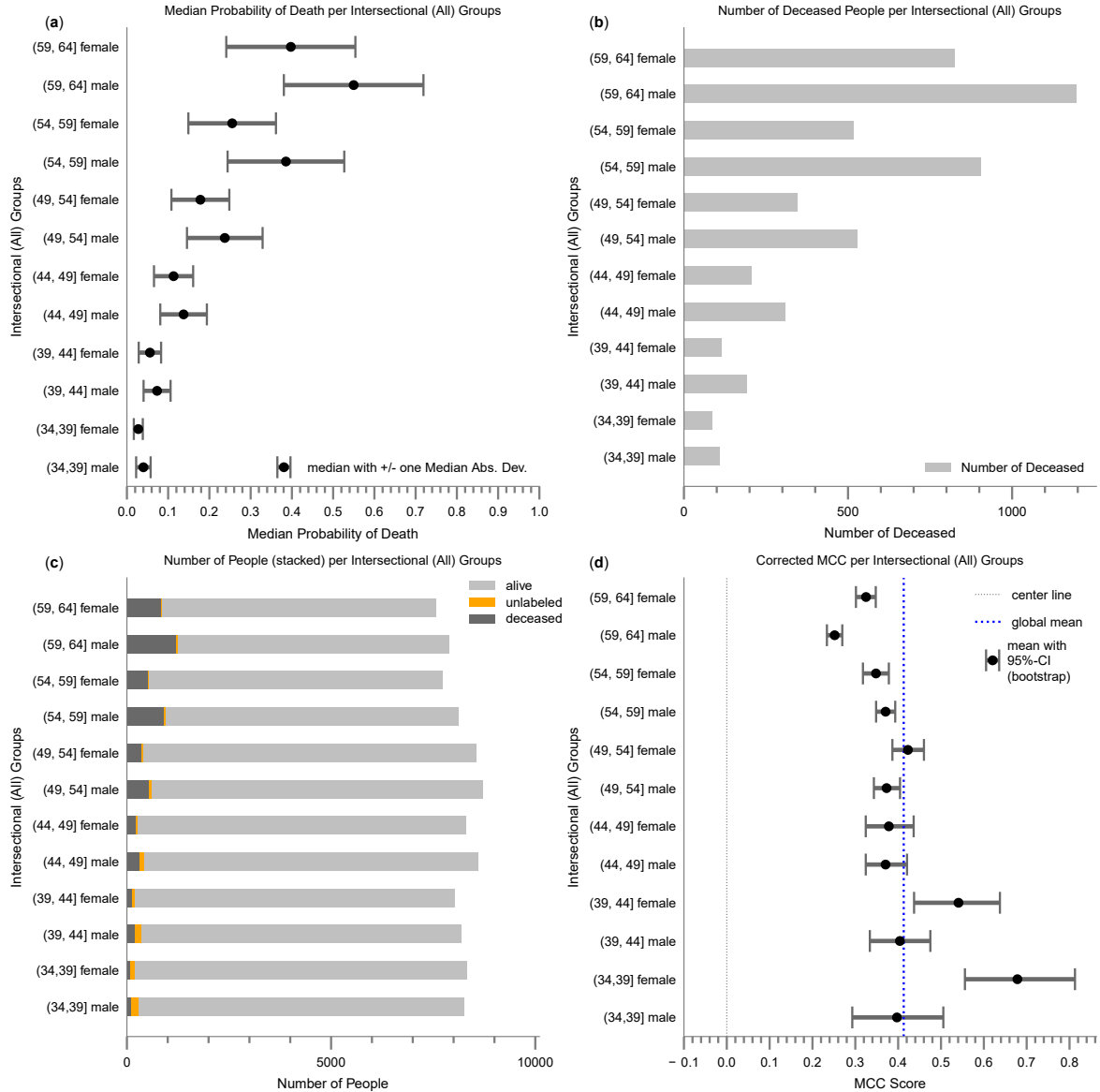
Supplementary Figure 4: Detailed evaluation of the *life2vec* model based on the **sequence length**. (a) The median predicted probability of early mortality per group (along with ± 1 Median Standard Deviation on error bars). (b) Number of known deceased people per group. (c) The overall number of people per group – stratified by the number of alive people, number of people with an unknown label (i.e. unlabeled), and number of known deceased people. (d) Corrected MCC (C-MCC) per each group – the mean value and the 95%-Confidence Interval (estimated by the stratified bootstrapping similarly to the procedure described in the Supplementary Table 3).



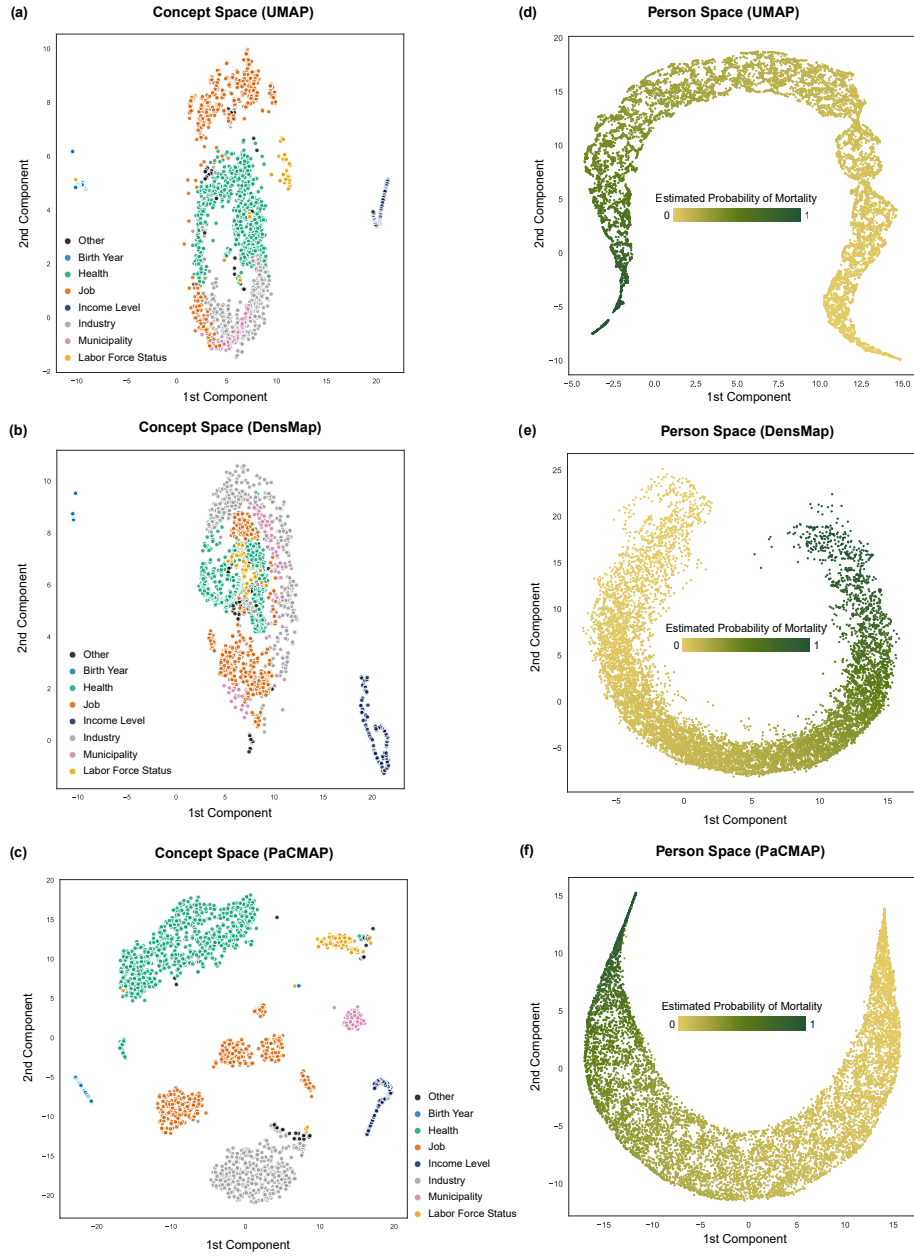
Supplementary Figure 5: Detailed evaluation of the `life2vec` model based on the **socio-demographic** attributes. (a) The median predicted probability of early mortality per group (along with ± 1 Median Standard Deviation on error bars). (b) Number of known deceased people per group. (c) The overall number of people per group – stratified by the number of alive people, number of people with an unknown label (i.e., unlabeled), and number of known deceased people. (d) Corrected MCC (C-MCC) per each group – the mean value and the 95%-Confidence Interval (estimated by the stratified bootstrapping similarly to the procedure described in the Supplementary Table 3).



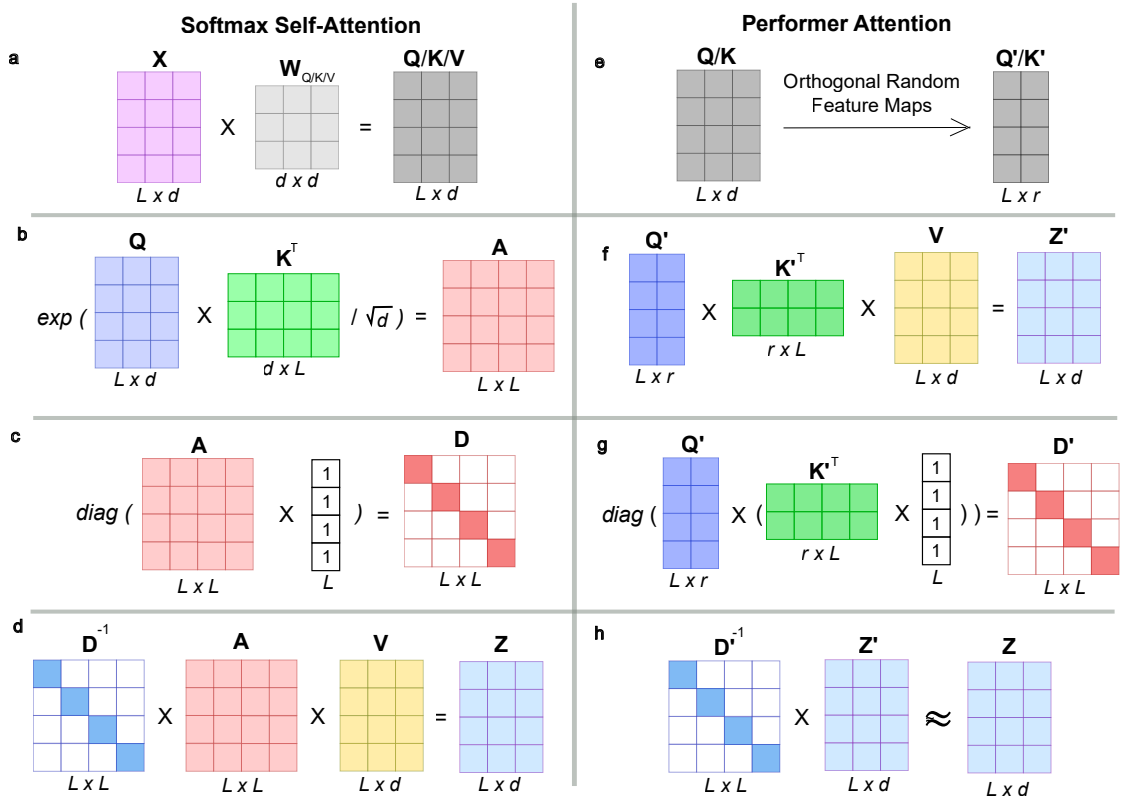
Supplementary Figure 6: Detailed evaluation of the life2vec model based on the intersection of **sex** and the **number of health events**. The label on the y-axis specifies the sex and the number of health events (associated with a particular group). **(a)** The median predicted probability of early mortality per group (along with ± 1 Median Standard Deviation on error bars). **(b)** Number of known deceased people per group. **(c)** The overall number of people per group – stratified by the number of alive people, number of people with an unknown label (i.e., unlabeled), and number of known deceased people. **(d)** Corrected MCC (C-MCC) per each group – the mean value and the 95%-Confidence Interval (estimated by the stratified bootstrapping similarly to the procedure described in the Supplementary Table 3).



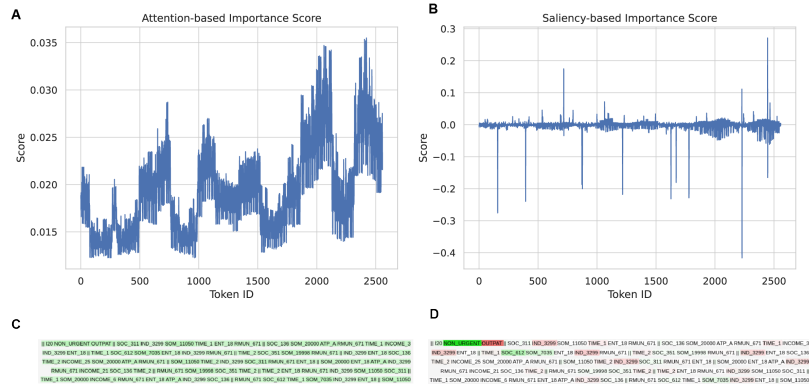
Supplementary Figure 7: Detailed evaluation of the `life2vec` model based on the intersection of **sex** and the **age** groups. The label on the y-axis specifies the sex and the age group (associated with a particular group). **(a)** The median predicted probability of early mortality per group (along with ± 1 Median Standard Deviation on error bars). **(b)** Number of known deceased people per group. **(c)** The overall number of people per group – stratified by the number of alive people, number of people with an unknown label (i.e., unlabeled), and number of known deceased people. **(d)** Corrected MCC (C-MCC) per each group – the mean value and the 95%-Confidence Interval (estimated by the stratified bootstrapping similarly to the procedure described in the Supplementary Table 3).



Supplementary Figure 8: Projections of Concept Embedding Space (a-c) and Person Embedding Space (d-f). We visualize each space with three different projection methods: DensMap, UMAP, and PaCMAP. In (a-c), concept tokens are colored based on the category. In (d-f) points are colored based on the predicted probability of mortality.



Supplementary Figure 9: Visualization of the Attention Mechanism [14]. (a-d) Steps to compute the Softmax Self-Attention. (a and e-h) Steps to compute the FAVOR+ Attention (i.e., Performer-style Attention) [15] that approximates the Softmax Self-Attention output, the visualization omits the details for computing the Orthogonal Random Feature maps (for more details refer to [15]). You can find corresponding equations in the Methods section of the manuscript



Supplementary Figure 10: Visualization of Attention and Saliency Scores for a specific sequence (only a small piece of a sequence is shown). This sequence belongs to a person who survived the four-year interval, and the model assigned the probability of an early mortality of 0.23. **(A - B)** Attention and Saliency score per each token in the sequence. **(C)** Part of a life-sequence, each word is colored based on the assigned Attention score. **(D)** Part of a life-sequence, each word is colored based on the assigned Saliency score. Saliency scores provide a more comprehensive importance score: red stands for negative contribution, and green stands for positive contribution. OUTPAT (outpatient visit) lowers the probability of early mortality, as well as IND_3299 (Manufacturing activities). However, SOC_612 (sick leave) increases the probability of death. The comparison between the Attention and Saliency scores supports the claim that Saliency is a more meaningful importance score [16].

Supplementary Tables

Supplementary Table 1: Table of Notations

PFF	$\triangleq$	Position-wise feed forward module
MLM	$\triangleq$	Masked Language Model
SOP	$\triangleq$	Sequence Order Prediction
S_r	$\triangleq$	a life-sequence of a r -th resident
s_r^i	$\triangleq$	i -th event in a life-sequence of a resident r
L	$\triangleq$	maximum length of a sequence
d	$\triangleq$	number of hidden dimensions
$\mathcal{E}_V$	$\triangleq$	embedding matrix of concepts
$\mathcal{E}_G$	$\triangleq$	embedding matrix of segments
$\mathcal{T}_A$	$\triangleq$	time2vec embedding of the age
$\mathcal{T}_P$	$\triangleq$	time2vec embedding of the absolute position
$\mathbf{ff}_i$	$\triangleq$	i -th fully connected layer
$\mathbf{W}_i$	$\triangleq$	weight matrix of the i -th layer
$\mathbf{b}_i$	$\triangleq$	bias vector of the i -th layer
g	$\triangleq$	trainable parameter
$\mathbf{A}$	$\triangleq$	Attention score matrix
$\mathbf{1}_L$	$\triangleq$	vector of 1s with the length of L

Functions		
$\text{Norm}(x)$	$=$	$\frac{x}{\ x\ }$
$\text{ScaleNorm}(x)$	$=$	$g \cdot \frac{x}{\ x\ }$ [13]
$\text{sigmoid}(x)$	$=$	$\frac{1}{1+e^{-x}}$
$\text{swish}(x)$	$=$	$x \cdot \text{sigmoid}(x)$ [17]
$\text{SigSoftmax}(x_i)$	$=$	$\frac{\exp(x_i) \cdot \text{sigmoid}(x_i)}{\sum \exp(x_j) \cdot \text{sigmoid}(x_j)}$ [18]

Supplementary Table 2: Specification of features and their sources.

Type	Feature	Source	# Categories	Encoding
Background Information	Sex	KOEN	2 binary	Male, Female
	Birth Month	FOED_DAG	12	Jan-Feb
	Birth Year	FOED_DAG	45	1946-1991
	Country of Origin	OPR_LAND	2 binary	National or International
Labour Records	Municipality of Residence	BOPAEL_KOM_KODE	97	Danish municipality codes
	Tax Bracket	ATP_BIDRAG_SATS_KODE	6	based on DST definitions
	Income Level	BREDT_LOEN_BELOEB	100	Quantile-based
	Labour Force Status	SOC_STATUS_KODE	35	based on DST definitions
	Labour Force Status (Modification)	TILSTAND_KODE_AMR	58	based on DST definitions
	Labour-Force-Interval	TILSTAND_LAENGDE_ARR	10	Quantile based
	Industry Area (Company)	ARB_HOVED_BRA_DB07	290	Danish Industry Classification System
	Job type	DISCO_KODE	359	International Standard Classification of Occupations
Health Records	Enterprise Type (Company)	ARB_SEKTORKODE	15	European System of Accounts
	Diagnosis	C_ADIAG	704	ICD-10
	Urgency	C_INDM	3	Urgent, Non-Urgent, Emergency
	Patient Type	C_PATTYPE	2	In-, out- patient

Supplementary Table 3: Corrected Matthew’s Correlation Coefficient (C-MCC) and Area under the Lift (AUL) on the Mortality Prediction Task. 95%-Confidence intervals for the C-MCC, Corrected Accuracy and Corrected F1-Score are based on the stratified bootstrapping – the number of bootstrapped sets is 5,000 (each set has 100,000 samples). In all cases, the higher value is preferred. Model Size specifies the number of trainable parameters, we performed a hyperparameter tuning on RNN, FFNN, and Logistic Regression.

Model	C-MCC, 95%-CI	AUL	C-Accuracy, 95%-CI	C-F1-Score, 95%-CI	Model Size
L2V	0.413 [0.410, 0.422]	0.845	0.788 [0.782, 0.794]	0.443 [0.435, 0.451]	8.4m
RNN-GRU	0.369 [0.361, 0.378]	0.834	0.778 [0.771, 0.783]	0.395 [0.389, 0.402]	1.5m
FFNN	0.340 [0.332, 0.348]	0.822	0.768 [0.762, 0.774]	0.345 [0.339, 0.350]	8.4m
Logistic Reg	0.149 [0.142, 0.155]	0.735	0.639 [0.633, 0.645]	0.201 [0.198, 0.204]	2.0k
Life Tables	0.059 [0.051, 0.066]	0.650	0.555 [0.548, 0.562]	0.161 [0.158, 0.164]	3
Random	-0.005 [-0.011, 0.002]	0.497	0.496 [0.489, 0.503]	0.132 [0.128, 0.135]	-
Majority Class	0.0	0.497	0.5	-	-

Supplementary Table 4: Corrected Matthew’s Correlation Coefficient (C-MCC) and Area under the Lift (AUL) on the Mortality Prediction Task. 95%-Confidence intervals for the C-MCC, Corrected Accuracy and Corrected F1-Score are based on the stratified bootstrapping – the number of bootstrapped sets is 5,000 (each set has 100,000 samples). In all cases, the higher value is preferred. Evaluation of the `life2vec` model on various data. Models are pretrained from scratch using these datasets and then finetuned accordingly. **Full Labour & Health** corresponds to the `life2vec` model described in the main manuscript. **Only Full labor** corresponds to the dataset without any health-related events. **Partial Labor & Health** contains health-related events and partial-labor events (without the specification of Work Industry, Sector, Position and Labour Force). **Partial Labor** consists only of partial-labor events (without the specification of Work Industry, Sector, Position and Labour Force).

Data	C-MCC, 95%-CI	AUL	C-Accuracy, 95%-CI	C-F1-Score, 95%-CI	Vocab Size
Full Labor & Health	0.413 [0.410, 0.422]	0.845	0.788 [0.782, 0.794]	0.443 [0.435, 0.450]	2043
Partial Labor & Health	0.375 [0.367, 0.384]	0.837	0.784 [0.777, 0.790]	0.399 [0.393, 0.406]	1034
Only Full Labor	0.319 [0.312, 0.327]	0.809	0.764 [0.758, 0.770]	0.340 [0.335, 0.345]	1290
Only Partial Labor	0.278 [0.271, 0.285]	0.782	0.737 [0.731, 0.743]	0.305 [0.300, 0.310]	281

Supplementary Table 5: Similarity of the concept space. Comparison of three `life2vec` instances initialized with different random seeds and trained on unique (nonoverlapping) subsets of the original training dataset. Models are trained for ten epochs. We compare the distance matrices generated by each of the models: $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$. We also include a distance matrix, $\mathcal{M}_R$ computed on the random embedding matrix, and a distance matrix computed on the permuted version of the embedding matrix associated with the first model, referred to as $\mathcal{M}_{1p}$. We follow the evaluation procedure described in the Methods Chapter to calculate the significance values using a one-sided permutation test – the number of permutations used is 5,000. To compute each p-value, we apply the continuity correction. Finally, p-values are adjusted with the Benjamini-Hochberg procedure to account for multiple comparisons.

Model Comparison	Spearman's ρ	Corrected p-value
$\mathcal{M}_1$ vs $\mathcal{M}_2$	.668	.00033
$\mathcal{M}_1$ vs $\mathcal{M}_3$	.660	.00033
$\mathcal{M}_2$ vs $\mathcal{M}_3$	.661	.00033
$\mathcal{M}_1$ vs $\mathcal{M}_R$	-.001	.7615
$\mathcal{M}_1$ vs $\mathcal{M}_{1p}$	.000	.3712

Supplementary Table 6: Similarity of the concept space. Comparison of the original concept space (generated from pretraining on the full dataset described in the Methods Chapter) to the concept space generated with different dataset versions (after 30 epochs of pretraining). We follow the evaluation procedure described in the Methods Chapter to calculate the significance values using a one-sided permutation test – the number of permutations used is 5,000. To compute each p-value, we apply the continuity correction. Finally, p-values are adjusted with the Benjamini-Hochberg procedure to account for multiple comparisons.

Data Composition	Spearman's ρ	Corrected p-value
Age: 25-35 Period: 2008-2015	.672	.0002
Age: 45-55 Period: 2008-2015	.691	.0002
Age: 55-70 Period: 2008-2015	.651	.0002
Age: 25-70 Period: 2008-2011	.687	.0002

Supplementary Table 7: Corrected Matthew's Correlation Coefficient (MCC) and Area under the Lift (AUL) on the *Emigration* Prediction Task. 95%-Confidence intervals for the C-MCC, Corrected Accuracy and Corrected F1-Score are based on the stratified bootstrapping – the number of bootstrapped sets is 5,000 (each set has 100,000 samples). In all cases, the higher value is preferred.

Model	C-MCC, 95%-CI	AUL	Accuracy, 95%-CI	F1-Score, 95%-CI	Model Size
L2V	0.168 [0.159, 0.177]	0.802	0.731 [0.719, 0.744]	0.130 [0.125, 0.134]	8.4m
RNN-GRU	0.144 [0.136, 0.151]	0.786	0.714 [0.702, 0.726]	0.106 [0.103, 0.110]	1.5m
Random	0.000 [-0.001, 0.009]	0.504	0.499 [0.486, 0.513]	0.052 [0.049, 0.055]	-
Majority Class	0.0	0.504	0.5	-	-

Supplementary Table 8: Hyperparameter optimization for the `life2vec` model. We use Bayesian search to find the optimal configuration of the parameters. The overall perplexity is calculated as a weighted sum of perplexities generated by the MLM task (0.7) and Sequence Order Prediction task (0.3). We train each model for five epochs and then pick six models with the lowest scores. Lastly, we train these six models for 30 epochs and choose the one with the lowest perplexity score on the validation set. Model Nr. 3 is the final configuration of the `life2vec` model.

ID	Overall Perplexity	Hidden Size	# encoders	# heads	# local heads	FF Hidden Size	# random features	Local Window Size
0	1.870	332	13	4	3	996	242	52
1	1.843	238	13	14	4	1586	263	32
2	1.859	208	13	8	3	2235	413	93
3	1.835	280	5	10	7	2210	436	38
4	1.925	80	11	10	6	1355	153	40
5	1.908	96	12	8	1	1790	360	41
6	1.881	184	6	4	1	709	469	19
7	1.857	196	5	14	11	1124	326	74
8	1.838	228	14	6	1	1605	490	65
9	1.859	208	12	4	1	2135	77	114
10	1.846	210	8	14	5	1615	356	49
11	1.889	312	8	13	12	1991	138	165
12	1.867	216	10	4	2	1839	223	99
13	1.886	154	12	14	1	1532	97	102
14	1.916	70	8	14	5	2074	410	41
15	1.829	270	6	10	1	1964	275	99
16	1.848	242	4	11	10	2386	120	80
17	1.884	168	7	12	7	1702	66	229
18	1.865	336	4	12	10	2432	512	14
19	1.848	336	4	8	4	2560	512	4
20	1.889	162	6	9	4	1214	188	137
21	1.886	220	9	11	8	2482	271	188
22	1.846	336	7	12	4	2560	512	68
23	1.873	336	10	8	7	2560	512	4
24	1.878	310	5	5	2	2049	241	108
25	1.867	322	6	14	9	882	282	150
26	2.036	270	9	10	0	2322	192	208
27	1.857	288	7	6	1	1964	325	154
28	2.014	144	9	6	0	1906	395	198
29	1.867	242	10	11	1	2285	158	140
30	1.925	301	8	7	5	1380	296	183
31	1.944	120	5	12	5	2391	361	130
32	1.851	252	7	9	1	1771	209	254
33	1.878	294	10	7	5	2506	316	167
34	1.873	300	5	10	9	2124	429	82
35	1.857	275	6	11	10	2019	452	118
36	1.946	261	4	9	5	1304	174	230
37	1.878	312	5	13	7	1713	376	169
38	1.827	253	6	11	9	2439	133	120
39	1.916	171	5	9	3	2168	297	254
40	1.857	273	7	7	3	1904	250	89
41	1.835	286	7	13	9	1446	207	138
42	1.859	234	4	9	4	2352	315	25
43	1.838	286	8	11	6	1213	346	55
44	1.848	260	6	13	12	556	114	116
45	1.881	238	9	14	10	2560	106	133
46	2.039	294	4	14	0	2560	236	53
47	2.031	336	6	14	0	1912	180	256
48	1.870	108	13	12	10	1388	98	103
49	1.952	70	7	10	10	1454	334	4
50	1.938	70	4	14	12	1096	497	4
51	1.933	66	4	3	2	877	231	256
52	1.838	294	13	14	8	2412	91	4
53	1.829	336	14	14	2	1785	105	4
54	1.862	252	10	14	2	2312	116	4
55	1.832	336	14	14	2	2096	64	256

Supplementary Table 9: Hyperparameter optimization for the RNN-GRU (Mortality Prediction). We use Bayesian search to find the optimal configuration of the parameters. We optimized parameters that were the most sensitive with respect to the model’s performance (determined manually). We pick the model with the highest AUL score (on the validation set after five epochs). Model 18 has the most optimal configuration of the hyperparameters.

ID	Hidden Size	# layers	Dropout, %	Bidirectional	AUL
1	370	1	0.16	False	0.7898
2	292	7	0.49	False	0.7824
3	155	7	0.02	False	0.7902
4	137	5	0.39	False	0.7898
5	525	5	0.01	False	0.7903
6	361	3	0.48	False	0.7865
7	646	3	0.35	False	0.7865
8	372	2	0.26	True	0.7902
9	260	7	0.09	False	0.7897
10	408	1	0.48	True	0.7890
11	367	2	0.37	False	0.7890
12	415	1	0.37	True	0.7897
13	267	1	0.40	True	0.7910
14	64	4	0.28	True	0.7910
15	64	1	0.50	False	0.7873
16	304	4	0.00	True	0.7893
17	768	8	0.00	True	0.7888
18	256	3	0.27	True	0.7912
19	710	2	0.04	False	0.7888
20	490	8	0.26	False	0.7375
21	268	8	0.21	False	0.7902
22	657	8	0.18	True	0.7728
23	248	8	0.14	False	0.7882
24	520	1	0.10	True	0.7893

Supplementary Table 10: Hyperparameter optimization for the Feedforward neural network (Mortality Prediction). We use Bayesian search to find the optimal configuration of the parameters. We optimized parameters that were the most sensitive with respect to the model’s performance (determined manually). We pick the model with the highest *AUL* score (on the validation set after five epochs). Model 7 has the most optimal configuration of the hyperparameters.

ID	Hidden Size	Layers	Dropout	LR	AUL
1	370	1	0.16	0.00039	0.7260
2	292	7	0.49	0.00043	0.6950
3	155	7	0.02	0.00378	0.7230
4	137	5	0.39	0.00374	0.7199
5	525	5	0.01	0.00303	0.7234
6	361	3	0.48	0.00431	0.7196
7	646	3	0.35	0.00017	0.7290
8	372	2	0.26	0.00503	0.7216
9	260	7	0.09	0.00296	0.7203
10	408	1	0.48	0.00595	0.7223
11	64	5	0.19	0.01000	0.7193
12	64	1	0.39	0.01000	0.7239
13	768	3	0.25	0.00001	0.6914
14	768	2	0.00	0.00171	0.7250
15	768	3	0.00	0.00109	0.7258
16	599	7	0.00	0.00718	0.7196
17	665	1	0.47	0.00874	0.7171
18	768	2	0.24	0.00041	0.7246
19	363	3	0.35	0.00736	0.7185
20	462	2	0.24	0.00065	0.7255
21	110	2	0.21	0.00700	0.7232
22	592	3	0.38	0.00866	0.7190
23	511	2	0.20	0.00273	0.7236
24	155	2	0.21	0.00244	0.7250

Supplementary Table 11: Concept sensitivity towards the "deceased" prediction. The table provides the median and the biweight midvariance (uncertainty) of the TCAV scores for eight different concept directions. The Wilcoxon signed-rank test compares each concept score distribution with the random concept score distribution (one-sided). The Wilcoxon test statistics is normalized by removing the mean and standard deviation of the Wilcoxon statistics under the null-hypothesis ($n = 10,000$). To compute p -values, we apply the continuity correction and use the Benjamini-Hochberg procedure to adjust p -values for multiple comparisons.

Concept	Z score	p value	Median	Biweight midvariance
Skilled Workers	83.68	.0	.9458	$\pm 3.09\text{e-}4$
High Income	-22.79	1.	.4072	$\pm .0108$
Infectious and Parasitic Diagnosis	40.55	.0	.4994	$\pm .0229$
Managerial Position	-42.02	1.	.3195	$\pm .0081$
Mental and Behavioral Diagnosis	65.61	.0	.6203	$\pm .0372$
Machine Operators	84.18	.0	.9549	$\pm 1.21\text{e-}4$
Male	86.56	.0	.9993	$\pm 1.87\text{e-}07$
Random	-	-	.3750	$\pm .1225$

Supplementary Table 12: Concept sensitivity towards the "alive" prediction. The table provides the median and the biweight midvariance (uncertainty) of the TCAV scores for eight different concept directions. The Wilcoxon signed-rank test compares each concept score distribution with the random concept score distribution (one-sided). The Wilcoxon test statistics is normalized by removing the mean and standard deviation of the Wilcoxon statistics under the null-hypothesis ($n = 10,000$). To compute p -values, we apply the continuity correction and use the Benjamini-Hochberg procedure to adjust p -values for multiple comparisons.

Concept	Z score	p value	Median	Biweight midvariance
Skilled Workers	-73.44	1.	.1711	$\pm .0031$
High Income	69.4	.0	.8376	$\pm .007$
Infectious and Parasitic Diagnosis	-61.49	1.	.3284	$\pm .0241$
Managerial Position	64.32	.0	.7951	$\pm .0062$
Mental and Behavioral Diagnosis	-70.22	1.	.2709	$\pm .2079$
Machine Operators	-77.9	1.	.1283	$\pm 9.05e-4$
Male	-86.45	1.	.0010	$\pm 1.05e-6$
Random	-	-	.5666	$\pm .1283$

References

- [1] Savcisens, G. Socialcomplexitylab/life2vec: Doi for publication. <https://doi.org/10.5281/zenodo.10118621> (2023). Zenodo.
- [2] Statistik, D. Arbejdsmarkedsregnskab (2022). URL <https://www.dst.dk/da/Statistik/emner/arbejde-og-indkomst/befolkningens-arbejdsmarkedsstatus/arbejdsmarkedsregnskab>.
- [3] Lynge, E., Sandegaard, J. L. & Rebolj, M. The danish national patient register. *Scand. J. Public Health* **39**, 30–33 (2011).
- [4] World Health Organization. *International Classification of Diseases, 10th Revision (ICD-10)* (World Health Organization, Geneva, 1994).
- [5] Müller, R., Kornblith, S. & Hinton, G. E. When does label smoothing help? *Proc. NeurIPS* **32** (2019).
- [6] Wang, C., Pu, J., Xu, Z. & Zhang, J. Asymmetric loss for positive-unlabeled learning. *2021 IEEE Int. Conf. Multim. Expo (ICME)* 1–6 (2021). URL <https://doi.org/10.1109/ICME51207.2021.9428350>.
- [7] Polat, G. *et al.* Class distance weighted cross-entropy loss for ulcerative colitis severity estimation. *Proc. MIUA* **13413**, 157–171 (2022). URL https://doi.org/10.1007/978-3-031-12053-4_12.
- [8] Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. *Proc. IEEE PAMI* **2**, 318–327 (2018). URL <https://doi.org/10.1109/TPAMI.2018.2858826>.
- [9] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. *Proc. IEEE CVPR* 2818–2826 (2016). URL <https://doi.org/10.1109/CVPR.2016.308>.
- [10] Groenendijk, R., Karaoglu, S., Gevers, T. & Mensink, T. Multi-loss weighting with coefficient of variations. *Proc. WACV* 1469–1478 (2021). URL <https://doi.org/10.1109/WACV48630.2021.00151>.
- [11] Yu, S. *et al.* A re-balancing strategy for class-imbalanced classification based on instance difficulty. *Proc. CVPR* 70–79 (2022). URL <https://doi.org/10.1109/CVPR52688.2022.00017>.
- [12] Kazemi, S. M. *et al.* Time2vec: learning a vector representation of time (2019). URL <https://doi.org/10.48550/arXiv.1907.05321>.

- [13] Nguyen, T. Q. & Salazar, J. Transformers without tears: improving the normalization of self-attention. *Proc. IWSLT* (2019). URL <https://aclanthology.org/2019.iwslt-1.17>.
- [14] Vaswani, A. *et al.* Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 5999–6009 (2017).
- [15] Choromanski, K. M. *et al.* Rethinking attention with performers. *Proc. ICLR* (2021). URL <https://openreview.net/forum?id=Ua6zuk0WRH>.
- [16] Atanasova, P., Simonsen, J. G., Lioma, C. & Augenstein, I. A diagnostic study of explainability techniques for text classification. *Proc. EMNLP* 3256–3274 (2020).
- [17] Ramachandran, P., Zoph, B. & Le, Q. V. Searching for activation functions. *Workshop ICLR* (2018). URL <https://openreview.net/forum?id=SkBYyZRZ>.
- [18] Kanai, S., Fujiwara, Y., Yamanaka, Y. & Adachi, S. Sigsoftmax: Reanalysis of the softmax bottleneck. *Proc. NeurIPS* **31** (2018).