universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Interpreting and Imitating Human Perception in Dimensionality Reduction via Surrogate Models“

verfasst von / submitted by

Raphael Mitsch, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2020 / Vienna, 2020

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von / Supervisor:	Univ. Prof. Torsten Möller, PhD

Erklärung zur Verfassung der Arbeit

Raphael Mitsch, BSc
Mollardgasse 33/23, 1060 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 16.10.2020

Raphael Mitsch

Danksagung

Mein Dank gilt meiner Familie, die mich immer unterstützt hat. Diese Arbeit wäre nicht zustande gekommen ohne Torsten Möllers Ratschläge und Ideen. Diskussionen mit Christoph Kralj lieferten wertvolles Feedback und Inspiration. Ich schulde allen Studienteilnehmern Dank für ihre Zeit und ihr Engagement. Meine spezielle Wertschätzung gilt all jenen, die die Produkte ihrer Arbeit und Recherche frei zur Verfügung stellen - sei es als wissenschaftliches Paper, offene Software oder in anderer Form. Ohne sie wäre dieses Unterfangen nicht möglich gewesen.

Acknowledgements

I would like to express my gratitude to my family for their unwavering support. This thesis would have not come about without Torsten Möller's ideas and guidance. Discussions with Christoph Kralj provided valuable feedback and inspiration. I am thankful to all study participants for their time and commitment. I owe a special debt of gratitude to all those making their work and knowledge available to the public, be it as scientific paper, open source software or in any other shape or form. This endeavour would not have been possible without them.

Kurzfassung

Dimensionalitätsreduktion wird im akademischen sowie im industriellen Umfeld häufig verwendet, um niedrigdimensionale Projektionen hochdimensionaler Daten - sogenannte Embeddings - zu erstellen. Deren Visualisierung kann jedoch schwer zu interpretieren und eventuell irreführend sein, besonders ohne eine klar definierte Grundwahrheit. Die von Benutzern gezogenen Schlüsse über Muster in den zugrunde liegenden Daten können stark darauf beruhen, wie die korrespondierende Visualisierung wahrgenommen wird. Eine Untersuchung dessen, wie dieser perzeptionelle Prozess abläuft und wie akkurat er ist, ist daher wichtig. Die vorliegende Masterarbeit widmet sich dieser Frage.
Wir stellen eine Applikation für die Interpretation, das Verständnis und die Bewertung solcher niedrigdimensionaler Embeddings vor. Diese erhielt in einer Benutzerstudie überwiegend positive Rückmeldungen und wurde von uns eingesetzt, um Bewertungen der wahrgenommenen Qualität von Embeddings zu sammeln.
Wir analysieren den Grad der Korrelation zwischen wahrgenommener und objektiver Embeddingqualität. Wir zeigen, dass erstere nicht ausschließlich durch zweitere erklärt wird und dass die verbleibende Lücke durch die Einbeziehung visueller Qualitätskriterien für Cluster- und Klassen-Separabilität reduziert werden kann. Der Einfluss dieser Kriterien auf die wahrgenommene Embeddingqualität wird untersucht. Unser Ansatz setzt nicht den überwachten Fall voraus und kann mit Embeddings von Entitäten ohne Klassenzuordnung umgehen.
Abschließend diskutieren wir sowohl Einschränkungen und Vorteile unserer Applikation und Analyse als auch Implikationen unserer Ergebnisse, mögliche Anwendungsfälle und zukünftige Verbesserungen.

Abstract

Dimensionality reduction is widely used in both academia and industry to generate low-dimensional projections of high-dimensional datasets. Visualizations thereof can be notoriously hard to interpret and even misleading, particularly lacking a clearly defined ground truth. Users' interpretation of and conclusions on patterns in the underlying data can be rooted in their perception of the corresponding low-dimensional embedding visualization. It is hence important to investigate how and how accurately this perceptual process operates, which is addressed in this thesis.
We introduce a tool designed for the task of interpreting, understanding and rating low-dimensional embeddings, which received favourable feedback in a user study. We employed this tool to obtain perceived embedding quality (PEQ) scores.
We analyze the degree to which objective quality and PEQ correlate. We show that the former is not fully explained by the latter and that the remaining gap can be narrowed by incorporating a set of visual quality criteria designed for cluster and class separation. The influence of these criteria on PEQ is examined. Our methodology does not presuppose the supervised case and handles embeddings of entities without class labels.
Finally, we discuss limitations and merits of our annotation tool and analysis methodology as well as implications of our results, potential applications and future improvements.

Contents

List of Figures

List of Tables

1 | Motivation

Dimensionality reduction offers a convenient and generic way to reduce complexity in datasets by projecting high-dimensional datasets to lower-dimensional spaces. It is commonly applied in visualization-centric use cases, since the lower number of dimensions allows for human-interpretable visual representations of the original dataset. Increasing success in mapping high-dimensional to low-dimensional spaces, e. g. with t-SNE [81], have propelled the popularity of those methods and led to a wider acceptance and broader usage of dimensionality reduction for visualization purposes amongst practitioners in different fields, both academia and industry. A good deal of literature on the improvement of dimensionality reduction methods as well as on dimensionality reduction in the context of visualization is available (see chapter 2 for more information).

We distinguish the task of employing dimensionality reduction for generating visual representations of high-dimensional datasets as an aid for human understanding from it serving as an intermediate step in a data processing or machine learning pipeline. Treating the former necessitates taking human perception and behaviour into account. Due to dimensionality reduction being a necessarily lossy process given sufficiently complex datasets, an evident instance of a perception-centered concern is visual error identification: Errors in the reduction process are indicated in the embedding's[1] visualization. We elaborate on this in chapter 5. Using dimensionality reduction without considering deviations from the high-dimensional dataset can lead to a loss of interpretative power and, in further consequence, corrode trust in the result. The pitfalls of interpreting embedding visualizations in general and t-SNE in particular are mentioned in Kobak and Berens [40], Wattenberg et al. [87], Nguyen and Holmes [57] and others.

In this thesis we focus on an issue concisely summarized in one question: How and how accurately do humans perceive and judge the quality of embeddings? Answering this question requires an understanding on the degree of alignment between embedding quality as perceived by humans and dimensionality reduction metrics as more objective measures thereof. There is a sizable body of

[1] We are using the terms low-dimensional *projection*, *embedding* and low-dimensional *mapping* synonymously - as in the result of a dimensionality reduction process yielding a low-dimensional representation of the original, high-dimensional dataset.

work introducing and discussing metrics attempting to capture an embedding's inherent faithfulness - i.e. how well properties are preserved throughout the dimensionality reduction process - that is discussed further in section 2.2. The majority of metrics rely in some fashion on measuring the preservation of one either distances between pairs of points or topological structure, i. e. some properties of the neighbourhood graph. Subsection 4.1.2 discusses this in-depth.

While these metrics' approaches to measuring faithfulness are usually reasonably intuitive, perceptual aspects are often neglected in the relevant literature or only considered in the supervised case, i.e. the availability of class labels is assumed. We argue that given the widespread usage of dimensionality reduction for the visualization of high-dimensional datasets and the inherent complexity of evaluating its results[2], investigating how users perceive embedding quality warrants more attention. We address the following research questions in the context of dimensionality reduction for visualization:

- **(RQ1)** To which degree does embedding quality as perceived by users correlate with embedding quality as measured by preservation metrics? In other words - if users believe an embedding to represent the underlying data well, how faithful is the embedding?
- **(RQ2)** How well can we model human judgement of embedding quality?
- **(RQ3)** Which factors wield which influence on how embedding quality is perceived?

1.1 Approach

Answering **(RQ1)** necessarily involves measuring the correlation of metrics with users' perceptual alignment. This entails gathering users' judgments of embeddings' quality, since such data in a sufficient scope, diversity and quality is as of the time of writing not publicly available. **(RQ2)** requires building a surrogate model aiming to imitate human rating behaviour from a set of embedding properties. While **(RQ3)** might be partially answered by inquiring users about their reasoning, an alternative and automated approach is to infer their motivations based on embeddings' properties. One way to achieve this is to extract feature importance and instance-based explanations from the model built to answer RQ2.

Due to the need of a sufficiently large and diverse set of human ratings of embeddings without the assumption of class labels, we introduce a *T*ool to *A*nnotate *L*ow-dimensional *E*mbeddings (*TALE*). To allow the development of a robust

[2]It is not always (1) trivial to interpret embeddings due to e.g. a lack of ground truth to compare the embedding to and (2) intuitive due to e.g. dimensionality reduction algorithms not taking spatial properties into account that users might expect, for instance inter-cluster distances.

model, we incorporated multiple sources of variance: different datasets, algorithms and hyperparameter configurations[3]. TALE serves to navigate the space spanned by these variables and to annotate embeddings in correspondence with the user's subjective judgement.

By leveraging the gathered data, correlations between perceived embedding quality (PEQ) and a set of dimensionality reduction metrics are computed. We show to which degree PEQ correlates with established dimensionality reduction metrics measuring embeddings' faithfulness. This allows for tangible recommendations for the visualization of embeddings, aiming to mitigate the risk of misinterpretation and to establish trust in the result.

Furthermore a surrogate model is trained on the same data to estimate PEQ by utilizing several existing dimensionality reduction metrics and other embedding properties as input features. We assume that those features capture different facets of the low-dimensional structure and contain enough latent information to allow the model to predict human rating patterns with a sufficient amount of accuracy. This model could e. g. serve as an approximate surrogate for human users during the evaluation of dimensionality reduction algorithms or explain why users prefer some embeddings over others, even if those preferences are not rooted in the embeddings' respective faithfulness. When combined with dimensionality reduction metrics it could also be employed in hyperparameter optimization workflows in order to generate embeddings that are still faithful, but score higher PEQ values and are thus more apprehensible and easier to interpret by users.

Explainability, i. e. the ability to describe *why* algorithms and self-learning models are inferring the way they do, was a crucial point of consideration throughout the entire process. This is reflected by our design decisions, from designing TALE's user interface to the choice of model architecture for the final surrogate model.

This thesis' contributions are threefold:

1. We review the process of designing an annotation tool for low-dimensional embeddings under consideration of different datasets, algorithms and hyperparameter configurations. We include feedback gathered in multiple rounds during the testing process and derive insights for future work in this direction. We implement the tool and provide it without restrictions on https://github.com/rmitsch/TALE.

2. We employ the tool to collect data on the perception of several both well- and lesser-known datasets. We publish the gathered data without restrictions on https://github.com/rmitsch/TALE.

3. We analyse the gathered data and discuss the underlying methodology. We investigate the degree of alignment between established preservation

[3]For a more detailed treatment of the requirement see 3.

metrics and PEQ scores and show that the maximal correlation is just below 0.7. We select, modify and implement various visual quality criteria intending to capture properties influencing PEQ. We train a surrogate model narrowing the gap in alignment between existing dimensionality reduction metrics and human perception by leveraging preservation metrics and visual quality criteria as features. We analyze the models accuracy as well as the contributions and influence of each feature.

The research process and the creation of this thesis were supervised by Torsten Möller. I contributed by (1) co-devising the research questions, (2) designing and implementing the embedding annotation tool, (3) planning, executing and evaluating usability interviews and tests and finally (4) selecting and training surrogate models for improved perception alignment.

2 | Related work

A vast corpus of research on the topics of dimensionality reduction and visualization is available. We restrict our deliberations here only to those contributions that are directly relevant due to their proximity to this work in regard of either one of those two fields.

For better readability we categorize this thesis' relation to prior work into six aspects:

1. Dimensionality reduction algorithms.
2. Dimensionality reduction metrics, as they are employed to appropriately evaluate low-dimensional projection quality.
3. User interactions with visualizations in general and dimensionality reduction visualizations in particular.
4. Visual parameter space analysis, i. e. how to enable users to explore parameter spaces spanned by a multiple variables.
5. Visual quality measures, i. e. measures designed for taking human perception into account in some manner.
6. Perception and interpretability of low-dimensional embeddings.

As of the time of writing there is, to the best of our knowledge, no other tool or paper aiming at either (1) building a tool enabling annotation of low-dimensional projections while supporting (multi-objective) PSA or at (2) training machine learning models to serve as perception-aligned proxy objectives for dimensionality reduction.

2.1 Dimensionality Reduction Algorithms

Van der Maaten et al. [82] compile an extensive and insightful structured analysis of dimensionality reduction techniques. Amongst other findings they present a taxonomy of the principal approaches in this field - see figure 2.1. Since then a number of new algorithms have emerged, amongst them the highly popular t-SNE [81] and UMAP [54]. Sarveniazi, [65], Cunningham and Ghahrami [25] as

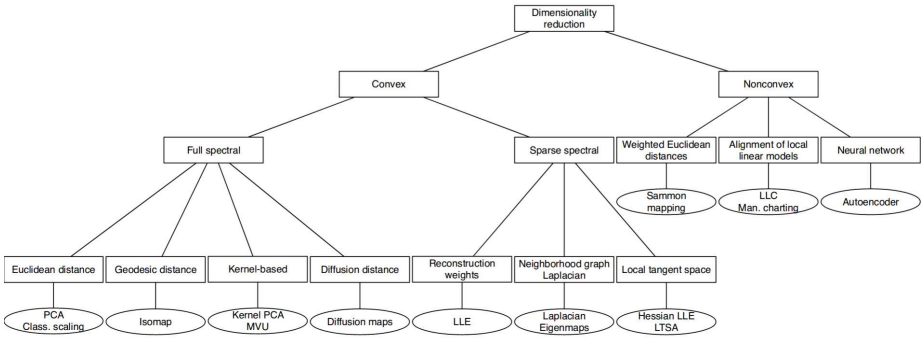


Figure 2.1: Taxonomy of dimensionality reduction techniques. Taken from van der Maaten et al. [82].

well as Wang and Sun [84] summarize newer developments[1] after the publication of the survey by van der Maaten et al. [82].

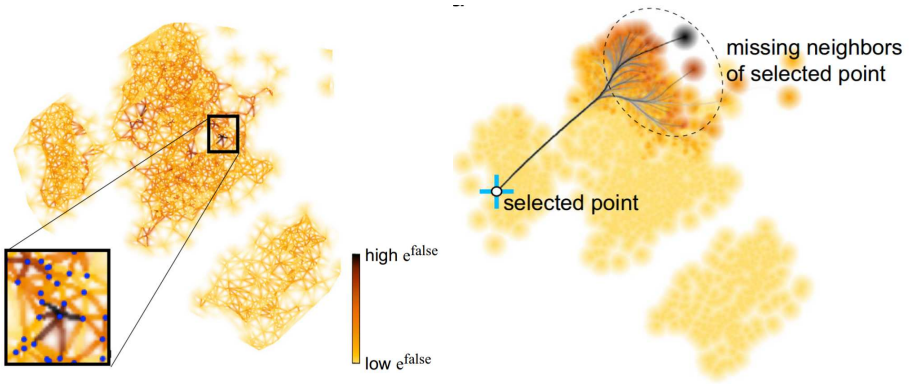
2.2 Dimensionality Reduction Metrics & Evaluation

Gracia et al. [30] provide a well written summary of the different types of evaluation metrics used in dimensionality reduction. It includes an introduction to the mathematical ideas behind those metrics, analyzes their usage in literature and proposes a methodology to compare the performance and stability of different dimensionality reduction algorithms.

Lee and Verleysen [43] as well as Lueks et al. [49] treat topology-based metrics in greater detail, particularly their integration into a common framework, the co-ranking matrix. The practice of combining several established quality measures for practical purposes like communicating an embedding's faithfulness is common and discussed e. g. by Johansson Fernstad et al. [27].

Beyond the most common metrics discussed in the surveys mentioned above there are also approaches deviating from the prevalent paradigm of pairwise absolute or rank distances. One example is the application of methods rooted in persistent homology as shown by Rieck and Leitte [62].

Lewis et al. [46] contributed a behavioral investigation of the evaluation of embeddings by dimensionality reduction novices and experts, which informed our test design discussed in chapter 7.



(a) False/intruding neighbours.

(b) Missing/extruding neighbours.

Figure 2.2: Pointwise indication displaced neighbours. Taken from Martins et al. [51].

2.3 Visualizing Low-Dimensional Embeddings

Sacha et al. [64] present a structured overview of user interactions with low-dimensional projection data. Liu et al. [47] reflect on progress in the visualization of high-dimensional data by identifying and categorizing approaches. Bertini et al. [17] are concerned with visualizing quality metrics for high-dimensional data. An introduction into visualization of low-dimensional projection data and its linkage with original data is given by Silva et al. [71].

Note that while only a few of the papers mentioned in those surveys explicitly concern themselves with highlighting errors in embeddings, the task of comparing multiple embeddings can be considered as strongly similar under some circumstances, as it also involves enabling the user to find divergences between an embedding and a reference structure[2].

2.4 Visual Parameter Space Analysis

According to Sedlmair et al., parameter space analysis (PSA) [66] is the *"...systematic variation of model input parameters, generating outputs for each combination of parameters, and investigating the relation between parameter settings and corresponding outputs."* Visual PSA is consequently the facilitation of (interactive) visualization to support this process.

Why do we care about PSA in this context? As argued in chapters 1 and chapter 5, in order to gather a sufficiently diverse dataset, we expect the user to

[1]We couldn't find any subsequent survey papers on dimensionality reduction methods.

[2]The 'reference structure' here being either the original high-dimensional data or an alternative embedding.

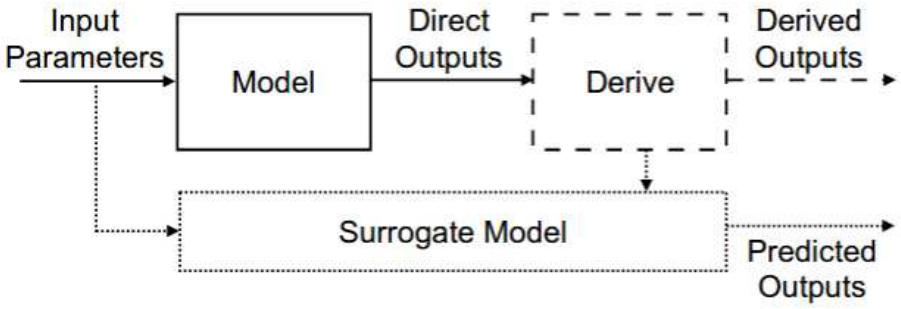


Figure 2.3: Abstract illustration of the data flow in a visual parameter space analysis. Taken from Sedlmair et al. [66].

navigate a multitude of embeddings characterized by their dataset, dimensionality reduction algorithm, hyperparameter configuration, and finally a subset of metrics estimating their quality. Even with our simplifications of exploring only one dataset and algorithm at a time[3], efficient navigation of the space of hyperparameters and metrics is crucial to allow users select informative embeddings. It follows that we built our design upon the work researched in this field.

The aforementioned seminal paper by Sedlmaier et al. [66] is a thorough introduction into this topic. Mühlbacher et al. [56] discuss strategies for enabling users to explore black box algorithms[4] utilizing VPSA. Further examples showcasing the merits of (V)PSA are Wang et al.'s [85] introduction of a special type of parallel coordinates plot for the visualization of parameters in climate simulations, and El-Assady et al.'s [26] visual analytics application for topic models.

2.5 Visual Quality Measures

The essential task of a visual (or perceptual) quality measure is to reflect human judgement in one or more tasks related to pattern recognition, e. g. cluster separability or outlier identification. Behrisch et al. [16] offer a comprehensive review on visual quality measures and the corresponding literature. According to their definition, visual quality measures consist of an optimization algorithm and the objective to optimize, which is referred to as visual quality criterion. Visual quality measures or criteria have been used to support finding accurate low-dimensional embeddings, e. g. by Tatu et al. [74], Wilkinson et al. [88] and Sips et al. [72].

[3]The reasons for this are discussed in detail in chapter 5.

[4]I. e. algorithms that do not inherently allow to retrace and understand *why* an algorithm or machine learning model behaves the way it does. Although explainability exists on a spectrum, the term 'black box (model)' is often used for model architectures that are more difficult to interpret, like deep neural nets.

Sedlmair and Aupetit [67] introduce a framework for the data-driven evaluation of visual quality, motivated by the need to overcome the quantitative limitation of having to use human judges. The fundamental idea is to train a machine learning model to learn the correlation between metrics and human judgement so that the alignment between those two can be inferred by the model and thus scaled to an arbitrary number of unseen datasets. They demonstrate their approach for the case of class separability.

Wang et al. [86] propose a perception-driven dimensionality reduction approach. They introduce two measures reflecting the visual separation between classes and describe the process of maximizing one of those measures using simulated annealing.

2.6 Perceptual Quality & Interpretability of Embeddings

Prior work on the perceived quality and interpretability of embeddings works almost exclusively within the confines of the supervised use case, i.e. the records to be embedded have at least one class label. This simplifies the analysis substantially, since established class separation measures can be used directly.

Bibal and Frénay [18] collect user preferences between pairs of low-dimensional labelled embeddings and fit a linear model to compute preference scores reflecting embeddings' perceived quality. They compare their model's accuracy with those achieved by two class separation measures, one cluster separation measure and one preservation metric. We integrated all four measures as features into our PEQ prediction model[5].

In a later paper Bibal and Frénay [19] propose to assess embedding interpretability by fitting a linear combination of weights for an undefined series of preservation metrics and visual quality criteria, which they refer to correspondingly as accuracy measures and interpretability measures. They recommend conducting an experiment in which users should rate embeddings according to their interpretability, i.e. how easy it is for them to recognize meaningful patterns in it. They note that users should be aware of the embeddings' preservation metrics while doing so. This methodology is virtually identical to the one we are following. Beyond that however we (1) design, implement and supply a tool for conducting the suggested experiment, (2) conduct the experiment and obtain data, (3) engineer features and fit a model to PEQ scores and (4) analyze the contribution of each feature.

We are not aware of any literature concerning the design and implementation of tools like TALE for the interactive interpretation, analysis and rating of embeddings.

[5]The class separation measures were modified to handle the unsupervised case. This is described in detail in table 4.1.

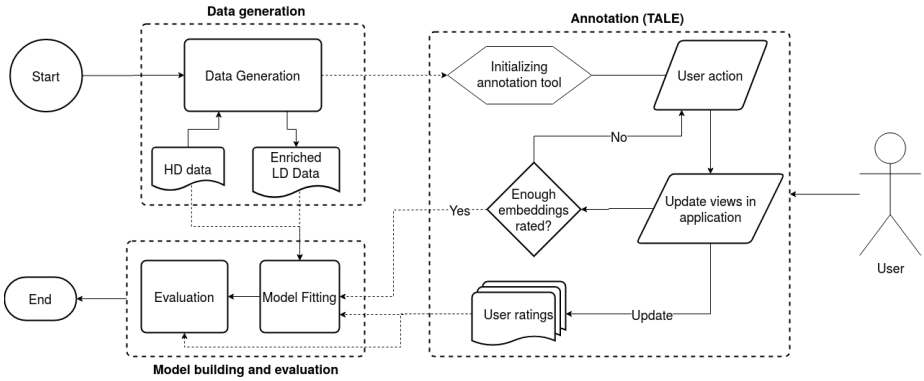


Figure 3.1: Data-flow in our solution from data generation to evaluation. Dashed lines mark interactions or usage between components.

3 | Data-Flow & Requirement Analysis

We establish and discuss the requirements to be met for our approach as described in section 1.1. This involves a high-level description of the necessary input data and user tasks in the annotation tool as well as a specification of the final surrogate machine learning model. We depict our solution's data-flow and contextualize our requirements by anchoring them to individual steps in this process.

The complete data-flow can be seen in figure 3.1. Note that there are three distinct phases: Data generation, annotation[1] and model building. Only the annotation process involves users - data generation happens prior, model building subsequent to any user involvement. A description of the data-flow follows in the next sections.

[1] *User action* represents any kind of user interaction with the UI, culminating at some point in the manipulation of an embedding's rating.

3.1 Data Generation

3.1.1 Functional Requirements

The original input dataset has to specify the raw dataset and some task that reasonably approximates the predictive power of a model in the target domain - such as a classifier or a regressor[2].

The expected output data consists of (1) a set of embeddings described by a set of supported dimensionality reduction algorithms and the corresponding hyperparameters and (2) the computed values for an array of dimensionality reduction metrics applied on the generated embeddings. The exact parameters for the embedding process that generated the dataset used during evaluation is specified in chapter 7. The set of applied dimensionality reduction metrics is described in 4.1.2.

3.1.2 Constraints

Since embeddings are generated before the annotation phase (see figure 3.1), there are no stringent performance requirements. Measures nonetheless undertaken to increase performance are discussed in 6.

3.2 Annotation (TALE)

The annotation component pictured in figure 3.1 is equivalent to TALE. Hence this section details requirements related to user interaction with a graphical interface in order to solve the problems imposed by the goal of manually evaluating embeddings in the context of the parameter space.

3.2.1 Functional Requirements

TALE has to support a bundle of user tasks in order to enable users to accurately form opinions on the faithfulness of embeddings and navigate the parameter space. We refer to their specification in chapter 5, since these tasks can be considered equivalent to the functional requirements of the *Annotation* component.

3.2.2 Constraints

Our approach requiring embeddings for multiple datasets being annotated by different users implies several critical non-functional requirements to be met by TALE:

[2]The reasons for this are elaborated upon in section 5

- *Responsiveness*[3]. As to not diminish the user's motivation and ability to navigate through the parameter space and evaluate embeddings, feedback to user actions should happen as close to instantaneous as possible. This applies regardless of dataset size (although realistically there is obviously a limit to what can be supported), meaning that ideally, feedback time should scale sub-linearly with the number of records in a dataset.
- *Portability.* Due to the need of gathering data from as many users as possible, the application should be runnable regardless of the user's underlying operating system.
- *Compactness.* Since we do not expect standardized equipment for user evaluations, we have to account for varying monitor sizes. Consequently we have to anticipate usage on small devices and factor that into our design considerations.
- *Low dimensionality.* The complexity of embedding visualizations increases with the number of involved dimensions. More complex visualizations are harder to examine and interpret. We hence restrict the maximum number of embedding dimensions to three.

Chapter 6 expands on how those requirements are being fulfilled from a technical perspective.

3.3 Model Building and Evaluation

3.3.1 Functional Requirements

To achieve learning a perception-aligned dimensionality reduction metric based on the data gathered and interpreting it, the corresponding model has to fulfill three key requirements.

- Accept and process data reflecting users' ratings of embeddings and values for several dimensionality reduction metrics.
- Infer the estimated perceived quality of an embedding based on the same feature set the model was trained on.
- Offer the possibility to interpret results and explain the contributing factors in the model's decision making process.

The first two items are fundamental to every machine learning model. Not all model architectures inherently implement the third item, but various methods explaining model inference while assuming the underlying models to act as black boxes exist (see e. g. *LIME* by Ribeiro et al. [61] or *SHAP* by Lundberg & Lee [50]).

[3]Note that we use this term to describe the application's latency: The time between users' actions and the application's responses.

3.3.2 Constraints

Our approach is only viable if it works reliably and accurately enough. This can be verified by comparing predictions on the quality of embeddings in a held-out dataset to the judgement of expert testers. Deliberations on what makes an acceptable level of accuracy is subject to personal opinions. Given that there is no clear threshold acting as demarcation between a good or bad model, we consider metric performance as a soft target, hoping to achieve mean absolute percentage errors of of 10% or less.

3.4 Evaluation

The conducted evaluations take two factors into account: Feedback on the usability of TALE and performance measures for the machine learning model(s) predicting perceived embedding quality.

3.4.1 Usability

To obtain a somewhat robust estimation of TALE's usability, all participants are queried about the simplicity, effectiveness and efficiency of the tool's interface. The participants' responses with respect to these facets are captured using a Likert scale-based questionnaire. The emanating feedback is measured quantitatively in form of SUS scores as introduced by Brooke [21] and qualitatively by analyzing free-form textual comments. The SUS questionnaire consists of the following ten questions:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

3.4.2 Model Performance

We aim to determine the quality of our machine learning models' fit to human perception. As mentioned in subsection 3.3.2, the metrics applied to assess this quality of fit are common when evaluating the performance of regression models. The specifics of this process are outlined in chapter 7.

4 | Algorithmic Components

A significant part of our contribution involves the application of existing algorithms and machine learning models. We hence expound on their characteristics and relevance in the context of our contribution, involving both the user-facing tool and the data analysis and model building part.

4.1 Dimensionality Reduction

Dimensionality reduction is the process of mapping high-dimensional data to a lower-dimensional space while minimizing some error criterion (see section 4.1.2 for an overview of used error criteria). A common problem in real-world applications is that the choice of algorithm and parametrization might influence the result dramatically - see Wattenberg and Viégas [87] for a rather influential, non-academic treatise on the volatility of t-SNE w.r.t. its parametrization. The selection of a metric to assess the resulting embedding's quality adds another degree of freedom.

4.1.1 Embedding Approaches

A classification and description of the different techniques used for dimensionality reduction is not in the scope of this paper. We refer to dedicated surveys on this topic such as one contributed by van der Maaten et al. [82]. Some of the more prevalent dimensionality reduction algorithms are MDS [77, 41], PCA [58], t-SNE [81] and UMAP [54].

Applications The wide array of fields in which dimensionality reduction-based visualizations are used further substantiates the impact of enabling users to judge faithfulness of low-dimensional projections. Some examples are:

- Law enforcement interpreting crime data, [36],
- clustering single-cell DNA data stemming from high-resolution dissection of tissue composition [14] and
- identifying tumors utilizing mass spectrometry data mapped to a low-dimensional space with t-SNE [9].

Embedding artefacts can have consequences changing the outcome of user-guided decision processes. Especially in sensitive domains like these it is therefore paramount to inform users about inaccuracies in the embedding process' outcome.

4.1.2 Metrics

A number of quality metrics assessing the quality of low-dimensional embeddings have been proposed over the years since the introduction of Kruskal's stress in 1964 [41]. They can be clustered in three groups[1]:

- Distance-based, i. e. the absolute point-wise distances in the original space in proportion to the absolute point-wise distances in the low-dimensional space. Distances are measured by a continuous metric, e. g. a p-norm.
- Topology-based, i. e. some measure of retention of the local or global topological properties of the k-neighbourhood graph.
- Based on the target domain, i. e. any kind of metric that captures performance on a task in the original as well as the reduced space. Examples are regression or classification tasks.

Distance-based

Distance-based metrics in the context of dimensionality reduction measure the proportion of pair-wise distances in the original space to those in the low-dimensional one. Perhaps the most well known is Kruskal's stress [41], introduced in 1964. Some of the more popular measures are (based on) the Euclidean, Cosine, Mahalanobis distances and the Kullback-Leibler divergence. More generally, any type of distance metric[2] between sets might be applied. This type of dimensionality reduction metric is often visualized in a Shepard diagram, shown in figure 4.1.

Note that some metrics compute the geodesic instead of common spatial distances, that is they compose neighbourhood graphs and measure distances between points via the connecting edges in this graph instead of utilizing datapoints' properties directly[3]. These metrics could be categorized as both topology- and distance-based.

Topology-based

Many topology-based metrics focus on neighbourhood retention, i. e. how stable any given point's neighbourhood is after the reduction process. Fundamental concepts in this context are *intrusions* and *extrusions*, sometimes also referred

[1] See the papers presented in section 2.2 for an in-depth treatment of this topic.

[2] *Distance metric*, like the L2 distance, as opposed to *distance-based* dimensionality reduction metric, like Kruskal's stress.

[3] Isomap [75] is an example of a dimensionality reduction technique that bases its projections on geodesic distances.

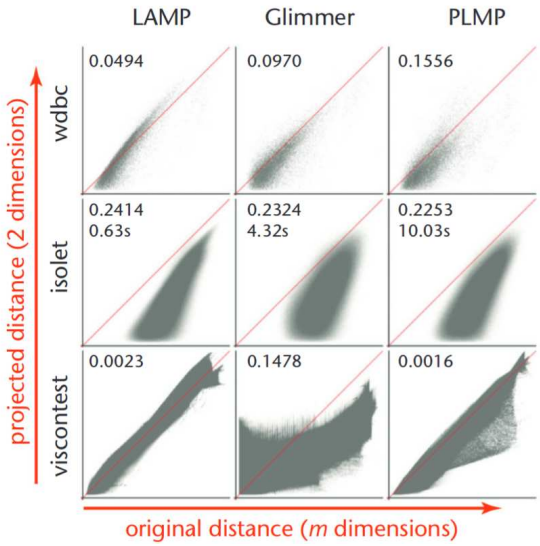


Figure 4.1: Several Shepard diagrams illustrating the relation between pairwise point distances in the high-dimensional (x-axis) and in the low-dimensional (y-axis) space for multiple datasets and dimensionality reduction algorithms. Taken from Silva et al. [71].

to as *false neighbours* and *missing neighbours*. The former describes the number of neighbours in a point's low-dimensional k-ary neighbourhood that aren't its neighbours in the high-dimensional one, the latter the number of neighbours in its high-dimensional k-ary neighbourhood not present in the low-dimensional one. This concept has been generalized to the co-ranking matrix [43] by Lee et. al, which describes a family of related measures for neighbourhood retention. Subsequent work done by, amongst others, Lee and Verleysen as well as Lueks et al. [44, 49] aims to further improve the generalized co-ranking framework. An illustration of a co-ranking matrix can be seen in figure 4.2.

The core of all measures based on the co-ranking matrix is a rank-wise comparison of distance between any pair of points: Let $\rho_{ij} = l \in \mathbb{N}$ define that record j is the l-th closest neighbour to record i in the high-dimensional space and let $r_{ij} \in \mathbb{N}$ represent the same concept in low-dimensional space. Then $\rho_{ij} > r_{ij}$ is referred to as *intrusion* (points are closer in rank than they should be) and $\rho_{ij} < r_{ij}$ as *extrusion* (points are farther apart in rank than they should be).

Since this family of metrics is at its core point-based, aggregation is necessary for it to describe an embedding's global quality. One way to achieve this is by averaging the co-ranking matrix-based values for each point in the dataset over varying values of k, with weights decreasing with k. This allows for discounting the impact of neighbourhood distortions with increasing topological distance.

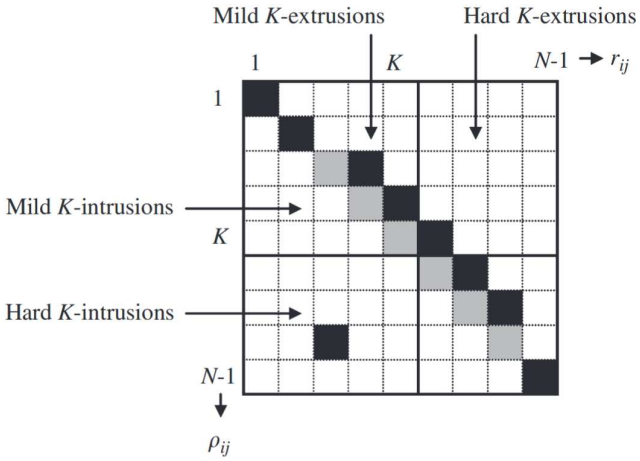


Figure 4.2: Illustration of an exemplary co-ranking matrix for a dataset with 10 points and a neighbourhood size of $k = 5$. The diagonal contains the numbers of all neighbourhood relations that retained their rank throughout the dimensionality reduction process. Intrusions are counted to the left of the diagonal, extrusions to its right. Taken from Lee and Verleysen [43].

Based On Target Domain

Utilizing the target domain as a mean to capture the accuracy of an embedding is fundamentally different from the two approaches mentioned before, as it is only indirectly concerned with the faithfulness of the embedding. Instead, accuracy is measured by applying the same evaluation procedure to the dataset in the high-dimensional and in the low-dimensional space. This evaluation procedure can be any applicable target domain performance measure, such as R^2 for regression tasks, precision or recall for classification tasks, clustering-based metrics like Silhouette or intra-cluster consistency for unsupervised tasks, and many others. The resulting performance in the low-dimensional space relative to the one in the high-dimensional space can be used as a proxy benchmark for accuracy loss (or, less likely, gain) after projection.

4.1.3 Discussion

Topology-based metrics have gained a lot of traction in the last decade, partly due to the rise of manifold-based dimensionality reduction techniques such as t-SNE [81], for which absolute inter-cluster distance might be drastically less informative than intra-cluster. Neighbourhood-based approaches measure local and not global faithfulness though, so in order to reach a verdict on an embedding's global quality, local measures have to be aggregated.

This requires design decisions, such as up to which k to sample and how to weight the individual measures. Target domain-based metrics on the other

hand are potentially useful if well defined model evaluation metrics are available, which is not always the case. They also do not consider faithfulness, which renders them a useful dimensionality reduction metrics measure by proxy only.

No single metric discussed here covers all angles, so for different use cases different metrics or combinations thereof might be beneficial.

4.2 Probing Black Boxes: Explainers

Explainers, also sometimes referred to XAI as in "eXplainable AI", are techniques designed to explain machine learning models' predictions. This can support human reasoning on the relevance of features for the model and allow the user to form an approximation of the model's mode of operation. This field of research has gained more traction in the last couple of years due to the rising influence of machine learning in everyday life and increasingly complex models that are thus more difficult to interpret and understand. The LIME explainer by Ribeiro et al. [61] in 2016 was of unprecedented popularity in the machine learning community for an XAI method, indicating the demand for these methods.

Such techniques are critical when building trust is required and helpful to assess the validity of the model's inference process. Some use cases for machine learning are subject to an especially high degree of scrutiny. This is exemplified by the medical field, where erroneous predictions might have potentially huge consequences such as the loss of human life. Under such circumstances predictions alone might not be sufficient - end users have to understand the model's reasoning for any given prediction and to assess whether they are trustworthy. Explainability techniques can aid this process. Adadi and Berrada [10] provide a thorough walk-through and an ontology (see figure 4.3) of the state of XAI up to the date of publication in 2018.

The relevant literature [10, 61] often distinguishes two definitions of trust in the context of the interaction of the end user and a machine learning model: "...(1) trusting a prediction, i.e. whether a user trusts an individual prediction sufficiently to take some action based on it, and (2) trusting a model, i.e. whether the user trusts a model to behave in reasonable ways if deployed.". Developers can derive information from global quality metrics to address the second issue. Machine learning explanation techniques aim to fill the gap for definition (1) and for end users not directly involved in the development of machine learning models by providing explanations for predictions in the hope of building trust by doing so.

Explainers can be categorized as model-agnostic or model-specific. Instances of the latter expose the decision making process for a particular type of model. They take explicit advantage of the model architecture, e.g. layers and nodes in neural networks or the tree structure in random forests. DeepLIFT by Shrikumar et al. [70] is an example for an explainer that is specifically designed for deep neural networks (DNNs), which have been the subject of model-specific explainers particularly often due to their widespread adoption, success and complexity

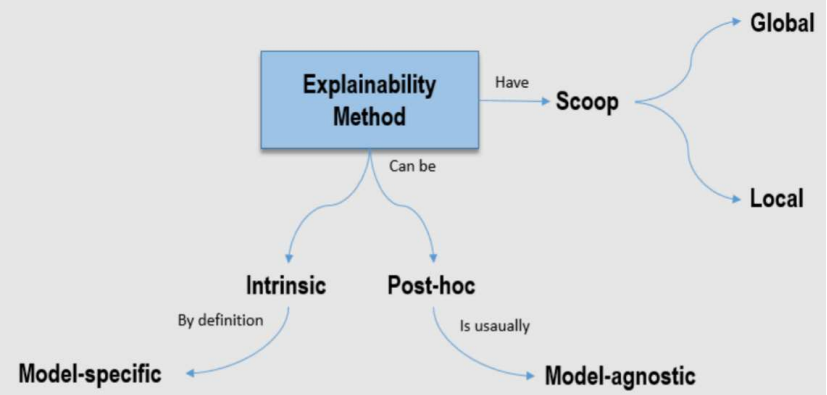


Figure 4.3: Adadi and Berrada's [10] ontology of XAI methods. Note: "Scoop" should be "Scope".
.

arising from up to billions of parameters.

Model-agnostic explainers on the other hand can be applied to any form of model. They often rely on surrogate models, for example by approximating a prediction locally by either training a surrogate model that acts locally congruent to the machine learning model, but is easier to interpret than the original model. This local model could be a decision tree, linear regression or any other machine learning model lending itself to accessible, intuitive human interpretation. Their simplicity and/or ability to derive simple explanations is key. The result of applying a generic explainer (in this case: LIME [61]) is illustrated in figure 4.4.

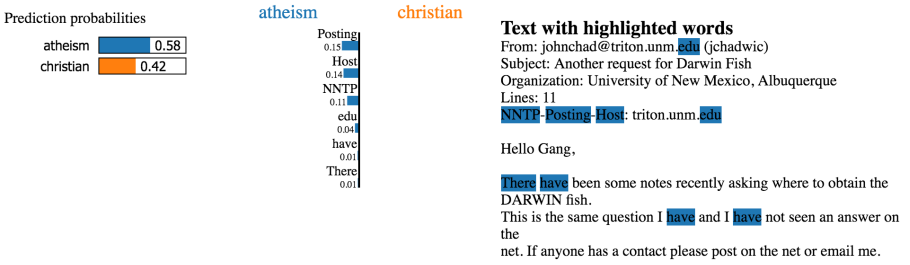
Lundberg and Li [50] presented SHAP, which comprises a set of different explanation methods based on Shapley values[4] [68]. We employ the author's official implementation[5] to explain the influence of dimensionality reduction hyperparameters on metrics to facilitate better understanding of the parameter space and algorithms' sensitivity towards hyperparameter configurations.

4.3 Modeling Human Perception of Embedding Quality

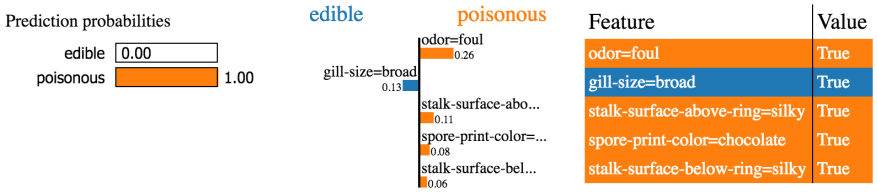
As a mean to understanding the process of perceiving the quality of low-dimensional embedding visualizations we train a machine learning model aiming to predict perceptual quality from a series of features. We employ a tree-based gradient boosting model due to this model type's trade-off of learning capabilities,

[4]Shapley values are rooted in game theory and were originally developed to measure player's payouts in a game based on their contributions. In the context of machine learning models, Shapley values measure features' contribution to a prediction for any given record in a dataset.

[5]https://github.com/slundberg/shap



(a) Binary text classification.



(b) Multi-class classification for tabular data.

(c) Image classification (green: cat, red: not cat).

Figure 4.4: Demonstration of the applicability of generic explainers (here: LIME by Ribeiro et al. [61]) on different types of models and data. Taken from the official github repository.

even with smaller datasets, and their popularity and hence compatibility and straightforward integration with explainers like SHAP. Alternative model types like Tibshirani's linear regression with LASSO [76] or deep learning methods for tabular data like Arik and Pfister's TabNet [12] are possible alternatives with their own drawbacks: Linear regression doesn't account for nonlinearity, whereas deep learning requires larger datasets to converge. For comparison both a boosting and a LASSO model were trained, with little difference in the accuracy of the respective predictions. See chapter 7 for a discussion of the results.

As discussed in sections 1.1 and 2.5, our initial hypothesis states that visual quality criteria might capture some aspects of perceptual embedding quality not reflected in preservation metrics. Hence some visual quality criteria shown to be effective in literature - as well as some other features derived or inspired from these criteria - were computed for each low-dimensional embedding and utilized

as features by the predictive model. They are listed in table 4.1. While some features are computed solely on properties of the low-dimensional embedding data, others also take the original high-dimensional space into account. The motivation behind the latter is to incorporate properties of the high-dimensional data - such as attributes or distances between points - into these features in lieu of a defined ground truth.

Feature	Description	Source
ABW	*Average between-within* is defined as the ratio of the average distance between points in the same cluster and the average distance between points in different clusters.	Lewis et al. [45].
Number of detected clusters	Number of detected cluster	-
$ABW_{high-dim}$	Like ABW, but distances are computed in the original high-dimensional space. Cluster membership is still inferred in the embedding space.	Inspired by Lewis et al. [45].
Share of detected outliers	Percentage of how many records are identified as outliers.	-
Density distribution	Distribution of records in the low-dimensional space. Record coordinates are binned in a two-dimensional grid. The numbers of points in each grid cell are computed. The resulting quantiles describing their distribution are used as features.	-
Unsupervised *DSC*	*Distance consistency* describes the (normalized) number of records who are closer to the centroid of another class than to their own. We apply an unsupervised variant in which class membership is replaced by membership in clusters in the high-dimensional space.	Inspired by Sips et al. [72].
Unsupervised *HM*	*Hypothesis margin* is computed as the average difference between the distance of each point to its closest neighbour in its own class and to its closest neighbour in a different class. As with unsupervised DSC, class membership here is exchanged with membership in high-dimensional clusters.	Inspired by Gilad-Bachrach et al. [29].
Number of dimensions	The number of dimensions in the low-dimensional space.	-

Table 4.1: Features used in the perceptual quality model in addition to preservation metrics.

Figure 5.1: TALE: Initial global view with referenced components. *HE*: Histograms over embedding hyperparameters; *SSP*: scattered scree plots indicating impact of individual hyperparameters on metrics; *HH*: hexagonal heatmaps representing correlations between metrics; *RP*: ratings panel with histogram over user ratings; *GT*: table with values for hyperparameters and metrics for all embeddings; *HHIG*: influence of hyperparameters on metrics in current set of embeddings. A more thorough description of these components can be found in table 5.1.

5 | User Interface Design

As depicted in the figure 2.3, the only component in our workflow that users directly interact with is TALE, our embedding labeling tool. In the following we outline its user interface's components and the rationale behind them. Furthermore we highlight some of the more central and impactful individual design choices made.

Figure 5.2: TALE: Initial local view for an embedding with referenced components. *HMI* displays hyperparameters and metrics. *HHIL* indicates which hyperparameters had which influence on which metrics. *SP* represents all records in the low-dimensional space as scatterplot matrix. *RT* lists all records with all of their attributes. *SD* and *CRM* correspondingly show the Shepard diagram and the coranking matrix, which represent how well neighbourhood distances and ranks have been preserved. *RC* allows to assign a rating to the currently shown embedding. A more thorough description of these components can be found in table 5.1.

5.1 General Design Choices

Some design choices are applied consistently throughout the entire application.

- **Overview first, zoom and filter, then details on demand.** Shneiderman's mantra *overview first, zoom and filter, then details on demand* [69], postulating that an overview of the information to be presented always precedes details and that details should only be shown if so selected and requested by the user. Following this guideline has proven effective in reducing users' cognitive stress by limiting the amount of information that has to be processed simultaneously.

  We incorporate this by splitting our interface in two major components: The *global view* and the *local view*. The former presents all generated embeddings and emphasizes their position in the parameter space. It includes correlations with hyperparameters and metrics, explanations of hyperparameters' influence on dimensionality reduction metrics and a list of all currently selected embeddings. The latter shows information on record-level granularity for one selected embedding.

We also offer details on visualizations' configuration, e.g. color encoding or glyph density, in a dedicated settings menu that encapsulates all those options that are not immediately essential, but nonetheless offer value for fine-tuning visualizations to improve the user experience.

- **Brushing and linking**. According to Hearst [31], brushing and linking can be understood as *"...the connecting of two or more views of the same data, such that a change to the representation in one view affects the representation in the other views as well."* Introduced by Becker and Cleveland [15], it enforces consistency in applications with multiple visualizations and facilitates a better understanding of the data by concurrently showing the same token of information from multiple perspectives.

  We adhere to this principle by synchronizing all visualizations in the global and local view respectively: (1) Hovered over data is highlighted in the other charts in this view and (2) a modification of the set of selected items is immediately reflected by all other charts.

- **User guidance**. The interaction between several levels of granularity with multiple orthogonal layers (e.g. embedding parametrization, the local explainer's estimation of the influence of hyperparameters on dimensionality reduction metrics in any given subset, etc.) brings about a degree of complexity that might negatively affect user experience and overwhelm users. To mitigate this, additionally to splitting the UI in a global and a local view, we attempt to guide the user through the experience. This is done by (1) giving an automated tour[1], illustrated in figure 5.3, of all individual UI elements and (2) hiding or fading them out if the information they represent is not relevant enough[2].

5.2 User tasks

In earlier stages TALE's design and functionality followed a predominantly experimental, iterative life cycle: Observations in feedback meetings often caused propositions of adjustments and new features, usually implemented until the next round of meetings. This allowed for a great degree of flexibility and a dynamic design process anchored in user feedback. It also abet feature creep to a certain extent however. To mitigate this, we specified a list of user tasks. This simplified the identification of unmet requirements and served as the basis for decisions regarding whether to add new or discard existing features.

All user tasks are considered as means to the end of supporting users to effectively and efficiently rate low-dimensional embeddings in accordance with their subjective estimation of its quality. Building on this premise, we identified three sets of user tasks[3]. We motivate each task with our design considerations

[1] This tour is started on application launch and can be skipped at any point.
[2] The relevancy threshold can be manipulated by the user.
[3] Task set X is denoted as TX, a user task Y as T$X.Y$.

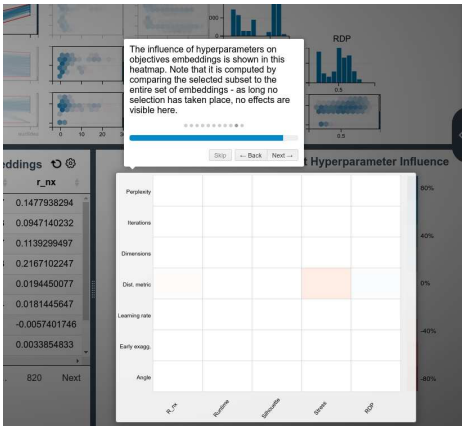


Figure 5.3: An exemplary step of the automated guided tour offered at application launch.

and/or feedback received during early feedback rounds. While there is some overlap in user tasks between sets, they can be considered mostly orthogonal and represent topically cohesive groups of tasks aiming at a common goal. Table 5.1 lists all UI components and their relation to the defined user tasks, figure 5.4 visualizes them as a matrix.

T_{INSP} | **Inspection, rating and assessment of individual embeddings.** This cluster involves all activities necessary for users to acquire enough information to accurately rate an embedding.

- $T_{INSP}.1$ | Filter embeddings based on their user-assigned quality ratings. Motivation: Users want to look up how many and which embeddings they have rated with which scores.
- $T_{INSP}.2$ | Locate a selected embedding in the global parameter space and relate its position to the subset of the other, non-selected embeddings. Motivation: Users want contextual information on whether the selected embedding is "special" in any way, e.g. whether it has extreme values compared to all other embeddings.
- $T_{INSP}.3$ | Obtain information on an embedding's quality that goes beyond a single scalar for the entire embedding, but doesn't require users to examine individual records. Motivation: When users select an embedding, they want a quick overview of the embedding's quality that is more informative than a global measure.
- $T_{INSP}.4$ | Access the dataset's records. Motivation: Users want to relate records' positions in the low-dimensional space to their properties, e.g. in order to find out how any given attribute aligns with axes.

- $T_{INSP}.5$ | Filter and sort records based on any of their attributes, including their position in the selected embedding's low-dimensional space. Motivation: Users want to (1) limit records to cohesive subsets, e.g. to assert that records in those subsets are close to each other in the low-dimensional space; (2) examine records sorted by any given property to see whether records similar w.r.t. this property are also close to each other in the low-dimensional space.

- $T_{INSP}.6$ | Explore an embedding's low-dimensional space and locate the dataset's records in it. Motivation: Users want to explore the embedding and locate known records in it - e.g. to use them as "anchors" to compare the position of other records with.

- $T_{INSP}.7$ | Gather the influences of each hyperparameter on each dimensionality reduction metric for an individual embedding. Motivation: Explaining the influence of hyperparameters should give users context, thus engaging them more strongly, and convey some insight on how the dimensionality reduction algorithm operates.

- $T_{INSP}.8$ | Rate an embedding after assessing its quality. Motivation: In order to extract a quantified measure of users' impression of an embedding, they are encouraged to score embeddings based on their perceived quality.

$T_{NAV\text{-}PS}$ **| Parameter space navigation and selection of interesting subsets of embeddings.** $T_{NAV\text{-}PS}$ includes all tasks required to select subsets of embeddings. This necessitates navigating the parameter space - e.g. filtering or selecting embeddings from different strata.

- $T_{NAV\text{-}PS}.1$ | Filter embeddings by any of their hyperparameters or dimensionality reduction metrics. Filters for different attributes should be combinable. Motivation: Users want to reduce the number of shown embeddings by limiting them based on any given property, e.g. to show only those embeddings scoring highest w.r.t. a particular metric.

- $T_{NAV\text{-}PS}.2$ | Obtain a sortable overview of all embeddings with their corresponding metadata. Motivation: Users want to quickly access the highest- and lowest-ranking embeddings w.r.t. any given property.

- $T_{NAV\text{-}PS}.3$ | Update subset of selected embeddings by deriving embeddings of interest from set of filtered embeddings. Motivation: Selecting subsets of embeddings can yield significantly different patterns, e.g. with regard to the correlation of hyperparameters to metrics. Users want to be able to drill down and discover interesting patterns, select the involved embeddings, discover more patterns etc.

- $T_{NAV\text{-}PS}.4$ | Reset all filters to their default state. Motivation: Users want to be able to remove all filters in order to revert the UI to its initial state.

$T_{INT\text{-}PS}$ | **Interpreting the parameter space for the currently selected subset.** All tasks related with understanding the parameter space, i.e. the connection between hyperparameters and dimensionality reduction metrics, are grouped together in this set.

- $T_{INT\text{-}PS}.1$ | Identify patterns of interaction and correlations between pairs of hyperparameters and dimensionality reduction metrics, including hyperparameter/metric and metric/metric pairs. This should yield insight into whether there is a causal relationship between the pair of examined attributes. Motivation: Users want to explore the parameter space and understand which hyperparameter influences which metric in which way to which degree in order to better understand how the dimensionality reduction algorithm operates.

- $T_{INT\text{-}PS}.2$ | Understand the influence of every hyperparameter on every metric in the selected set of embeddings. Note that this task is related to $T_{INT\text{-}PS}.1$, yet differs from it in terms of emphasis: The focus is not on gauging the nature of the relationship between two attributes, but rather to offer a concise synopsis of how positive or negative each hyperparameter affects each measured metric. Motivation: Users want a concise overview of the effects of hyperparameters on metrics that doesn't require them to analyse visual patterns.

- $T_{INT\text{-}PS}.3$ | Link individual embeddings to their position in parameter space. Note that this is akin to $T_{INSP}.2$.

A note on $T_{NAV\text{-}PS}$ - it can be argued that in lieu of offering functionality to interpret and understand the parameter space we might algorithmically select the next embedding to rate instead, in effect rendering the entirety of $T_{NAV\text{-}PS}$ irrelevant. We offer Amershi et al.'s [11] findings[4] as counterarguments, specifically:

(1) *"Users are people, not oracles."* Users tend to cope badly with an immutable stream of questions generated by a non-human agent. It influences both their mood and the resulting label quality negatively.

(2) *"Transparency can help people provide better labels."* Providing explanations, context and agent-generated estimations can reduce human error when labeling, thus increasing label accuracy. Being subjected to contextual information and transparency also seems to impact users' mood positively.

We address these issues raised by Amershi et al. by enabling users to carry out the tasks in $T_{NAV\text{-}PS}$, which grant agency to the user and offer context, explanations and machine-generated estimations of embeddings' quality. While we hence consider a purely algorithm-based selection of embeddings as detrimental, supporting users in their execution of the tasks in $T_{NAV\text{-}PS}$ with active learning might increase labeling efficiency. We touch on this in section 8.3.

[4] Their work is on active learning. While our use case does not make use of active learning, the general setup - users are required to label datasets one record at a time - is similar enough to infer that these findings are likely to apply here as well.

5.3 UI Components

As mentioned in section 5.1, TALE's interface is split in the global and the local view. They incorporate two different sets of user interface components, addressing different segments of the user tasks introduced in section 5.2. The relation between UI components and user tasks is summarized in table 5.1 and visualized in figure 5.4.

ID	Description	View	Related Task
HE	Histograms for embedding attributes: Distribution of attributes' values. See figure 5.5.	Global	T_{INSP}.2, $T_{NAV\text{-}PS}$.1, $T_{NAV\text{-}PS}$.3, $T_{INT\text{-}PS}$.3
SSP	Scattered Scree plots: Effect of change in one hyperparameter on one metric (while all other hyperparameters are fixed). See figure 5.6.	Global	T_{INSP}.2, $T_{NAV\text{-}PS}$.1, $T_{NAV\text{-}PS}$.3, $T_{INT\text{-}PS}$.1, $T_{INT\text{-}PS}$.3
HH	Hexagonal heatmaps: Correlations between metrics. See figure 5.8.	Global	T_{INSP}.2, $T_{NAV\text{-}PS}$.1, $T_{NAV\text{-}PS}$.3, $T_{INT\text{-}PS}$.1, $T_{INT\text{-}PS}$.3
RP	Ratings panel: Selection of embeddings by their rating (respectively lack of rating). See figure 5.9.	Global	T_{INSP}.1, T_{INSP}.2, $T_{INT\text{-}PS}$.3
GT	Global table: Lists all embeddings with their metadata. See figure 5.10.	Global	$T_{NAV\text{-}PS}$.2, $T_{NAV\text{-}PS}$.3, $T_{NAV\text{-}PS}$.4
HHIG	Heatmap for hyperparameter influences, global: Influence of each hyperparameter on each metric for selected subset of embeddings in global view. See figure 5.11.	Global	$T_{INT\text{-}PS}$.2
HMI	Indicator of hyperparameter and metric values for selected embedding. See figure 5.14.	Local	T_{INSP}.2
HHIL	Heatmap for hyperparameter influences, local: Influence of each hyperparameter on each metric for selected embedding in local view. See figure 5.12.	Local	T_{INSP}.7

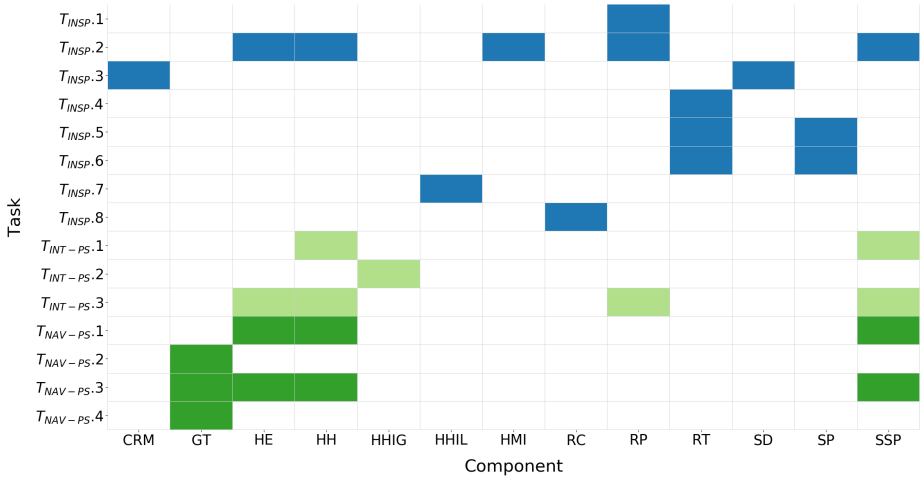


Figure 5.4: Associations between user tasks and UI components. Cell color is representative of the corresponding task group (T_{INSP}, $T_{INT\text{-}PS}$ or $T_{NAV\text{-}PS}$).

SP	Scatter plots for records in low-dimensional space: Each record's position in the low-dimensional space according to selected embedding. See figure 5.13.	Local	$T_{INSP}.5$, $T_{INSP}.6$
RT	Record data table: List of all records in dataset. Includes all attributes in dataset. See figure 5.15.	Local	$T_{INSP}.4$, $T_{INSP}.5$, $T_{INSP}.6$
SD	Shepard diagram: Selected embedding's faithfulness in terms of consistency of pair-wise spatial distances. See figure 5.16.	Local	$T_{INSP}.3$
CRM	Co-ranking matrix: Selected embedding's faithfulness in terms of consistency of pair-wise topological distances. See figure 5.17.	Local	$T_{INSP}.3$
RC	Rating component: Assignment of user rating to selected embedding. See figure 5.18.	Local	$T_{INSP}.8$

Table 5.1: UI components and their relations to user tasks.

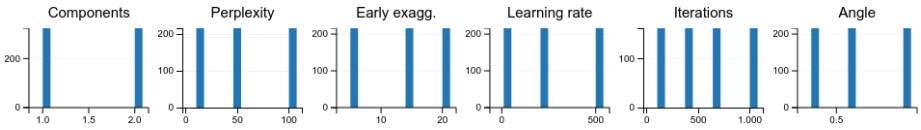


Figure 5.5: Component HE: Histogram of hyperparameters for selected embeddings.

.

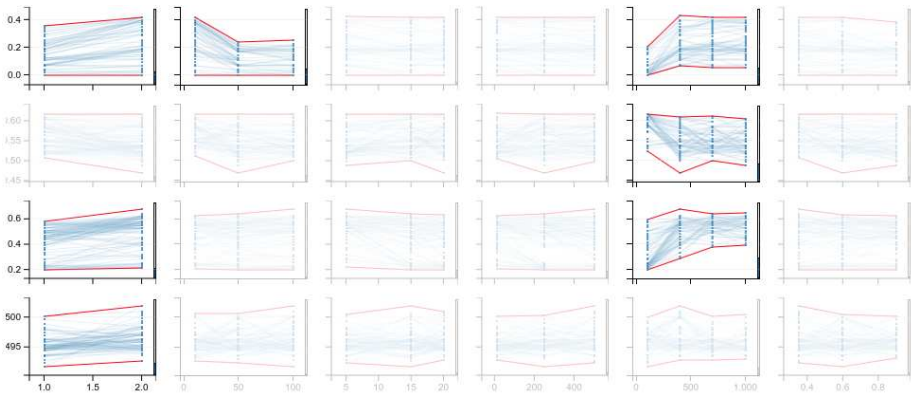


Figure 5.6: Component SSP: Scattered scree plots for parameter space analysis.

.

5.3.1 Global View

The global view deals mostly with tasks from clusters $T_{NAV\text{-}PS}$ and $T_{INT\text{-}PS}$ (figure 5.4 allows for a direct mapping of tasks to individual UI components). Consequently most of its components aim to convey an overview of the parameter space and to assist the user in narrowing down his selection of embeddings.

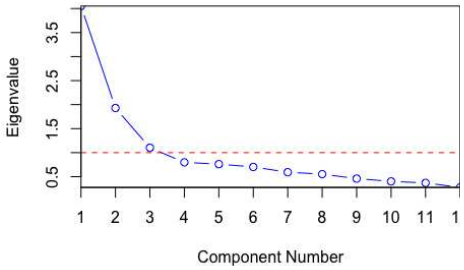
HE: Histograms

HE uses histograms - for each embedding hyperparameter and metric - as an established form to inform the user about the distribution of data over the corresponding attribute. Each histogram also serves as range filter: The user can select a range of values. See figure 5.5.

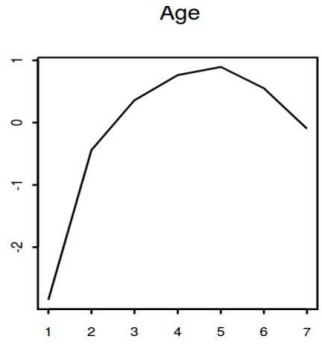
SSP: Scattered Scree Plots[5]

SSP (see figure 5.6) expresses the change in behaviour, as measured by a metric, where all hyperparameters except the one in question are considered fixed. In other words, it represents a form of sensitivity analysis: If a new embedding with exactly the same configuration for all but one hyperparameter were generated,

[5] As set forth in the text below, the name is derived from the plot's affinity to scatter and scree plots.



(a) Scree plot showing eigenvalues w.r.t. the number of retained components after PCA.



(b) Partial dependence plot showing how age affects income (y-axis).

(c) Scattered scree plot for influence of the number of dimensions (x-axis) on runtime (y-axis).

Figure 5.7: Juxtaposition of a scree plot, a partial dependence plot and a scattered scree plot. Sources: (a) Wikipedia article on scree plots, [7], (b) Friedman [28], (c) TALE.

how would the resulting metric be affected? A single scattered scree plot (SSP) renders single embeddings as points and then connects series of embeddings with a line. A series comprises all embeddings with identical parametrizations except for the hyperparameter in question. The ensemble of all series forms a Pareto frontier reflecting the highest achieved metric values for every variation of the corresponding hyperparameter.

This approach is an extension of Cattell's scree plots [24], which are often used in statistics and machine learning to estimate the effect of the number of dimensions on the dimensionality-reduced dataset's information content. It is also closely related to Friedman's partial dependence plots [28], which depict the marginal effect of one or two attributes on a machine learning model's inferred output. figure 5.7 juxtaposes scree plots, partial dependence plots and SSPs.

The distance correlation, introduced by Székely et al. [73], between hyperparameter and metric is displayed in a single bar to the right of a SSP. In contrast to the more commonly used Pearson correlation coefficient it also detects nonlinear correlation. Its visualization concisely represents the correlation strength. The complete matrix of SSPs covers every combination of hyperparameters and metrics, hence their number s is $s = h \cdot m$, where h is the number of hyperparameters and m the number of metrics. Due to the potentially large number

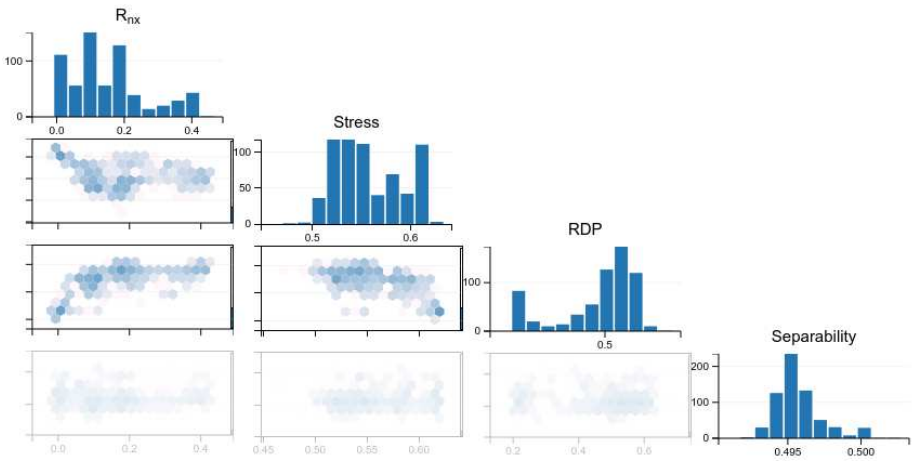


Figure 5.8: Component HH: Hexagonal heatmaps showing correlations between metrics.

of resulting SSPs those with a degree of correlation below a set threshold are faded out to diminish the user's cognitive stress. This is justified with their assumed irrelevance based on the low correlation score between their pairings of hyperparameters and metrics. The threshold for this is adjustable by the user.

Our implementation allows to select embeddings or series of embeddings with an area brush.

HH: Hexagonal Heat Maps

We present data resulting from pairings of two metrics in hexagonal heatmaps as introduced by Carr et al. [22]. We chose hexagonal over square heatmaps due to hexagons being closer to circles, which translates into a more faithful data aggregation around the center of the cell than with squares. Embeddings can be selected by selecting an area enclosing their containing hexagon. See figure 5.8.

RP: Ratings Histogram

Embeddings' ratings are shown as a histogram. All embeddings start without a rating. The ratings histogram is designed to allow insights into the distribution of user-assigned ratings and allow selection and filtering by rating. Unrated embeddings are hidden by default. This can be toggled with a control next to the histogram. See figure 5.9.

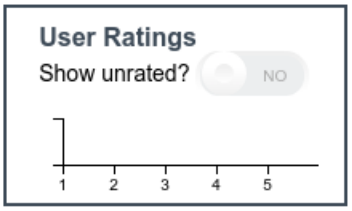


Figure 5.9: Component RP: Ratings pane with histogram of all ratings assigned by user.

Show entries Search: All embeddings

ID	Rating	n_components	perplexity	early_exaggeration	learning_rate	n_iter	angle	r_nx	stress	rdp	separability
0	0	1	10	5	250	100	0.6	0.0503444523	0.5762340426	0.493488729	0.4969854057
1	0	1	50	20	250	700	0.35	0.0912751183	0.553526938	0.5255864263	0.4941666722
2	0	2	50	20	10	100	0.35	-0.0014780943	0.6200031042	0.2306342125	0.4975020885
3	0	2	100	15	10	400	0.35	0.1907415688	0.533033669	0.5689522624	0.4953543842
4	0	2	100	20	500	100	0.6	0.0058960766	0.6200789213	0.2267445624	0.4952603281
5	0	2	50	15	250	1000	0.9	0.1684928685	0.5096985102	0.5656601191	0.4956123829
6	0	1	50	20	250	100	0.35	-0.0007968433	0.6186413765	0.2033361197	0.4951344132
7	0	2	50	20	10	400	0.35	0.1878339946	0.5455796719	0.5837818384	0.5001708865
8	0	1	100	5	10	400	0.9	0.1044132859	0.5246009231	0.5379235744	0.4950955808
9	0	1	10	5	10	700	0.35	0.3014246523	0.5841159821	0.5692326427	0.4947369397
10	0	2	50	5	10	700	0.6	0.1949918121	0.5336705446	0.528398037	0.4950903654
11	0	2	50	15	10	400	0.35	0.1682862639	0.5279732943	0.572724402	0.4950588644

Showing 1 to 15 of 648 entries Previous 1 2 3 4 5 ... 44 Next

Figure 5.10: Component GT: Table listing embeddings with all of their attributes.

.

GT: Embedding Table

The embedding table lists all embeddings with their attributes, that is hyperparameters and metrics. It allows for sorting by columns and offers a free text search in all embedding properties. Hovering over a row highlights the corresponding embedding in other charts. An arbitrary number of rows can be selected in the table, thereby reducing the set of embeddings to the selection. A reset button is offered to set the filter state for the entire application back to default, i.e. with all embeddings being available. See figure 5.10.

HHIG: Hyperparameter Influence Heat Map

SHAP - see section 4.2 - is employed to compute the contributions of each hyperparameter for each embedding. All contributions are averaged and displayed as a heat map where each cell marks the influence, positive or negative, of a particular hyperparameter on a particular metric. The size of a cell represents the magnitude of the hyperparameter's effect on the metric, the color indicates whether its influence is positive or negative. SHAP contributions over an entire dataset average to zero, consequentially this visualization will show a mostly

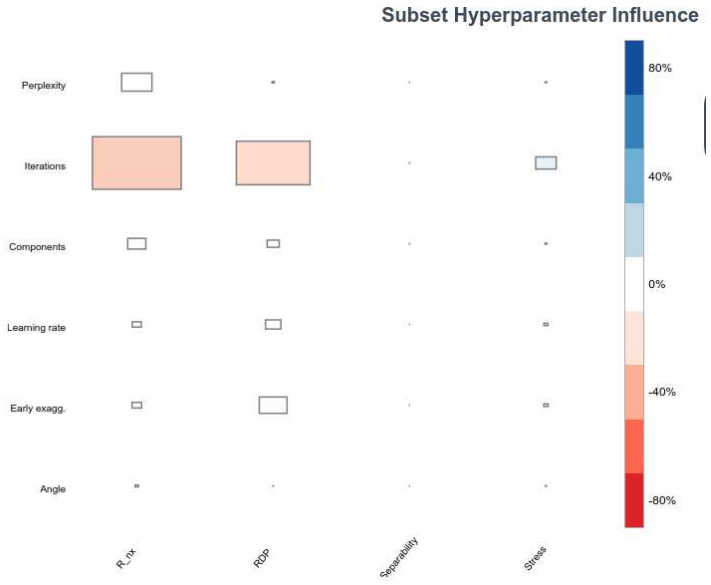


Figure 5.11: Component HHIG: Heatmap reflecting the influence of hyperparameters on metrics for selected set of embeddings.

.

neutral map before subsets of embeddings are selected. While this can be considered a shortcoming[6], the combination of HE, SSP and this heat map - see figure 5.11 - allow to counterbalance each component's weakness: HE and SSP might be used to identify regions of interest in the parameter space, but fail to consider multivariate effects. SSP can lose explanatory power after a selection of a relevant subset. HH, in contrast, yields more informative statements with smaller datasets and does not depend on correlations in the selected subset to explain hyperparameter influence. It furthermore compiles multivariate effects into a single visualization.

5.3.2 Local View

Components SP to RC are part of the local view. It covers exclusively tasks from cluster T_{INSP}, i.e. it provides functionality for the assessment and rating of individual embeddings.

[6]From a usability perspective there is no sound reason why the user shouldn't be given information on hyperparameter influence on the entire dataset. SHAP explanations don't lend themselves to this kind of aggregation though due to their properties - computing non-neutral statements about a set of data always requires a separate dataset used as reference. Discarding some of the generated embedding data just for this reason does not seem reasonable in our case however.

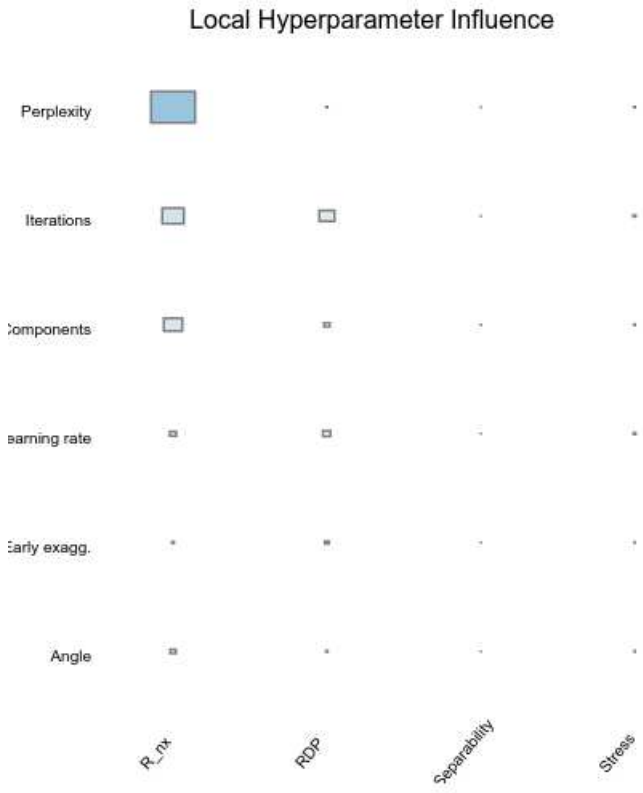


Figure 5.12: Component HHIL: Heatmap reflecting the influence of hyperparameters on metrics for selected embedding.

.

HHIL: Hyperparameter Influence Heat Map

Identical to the component characterized in subsubsection 5.3.1, but showing hyperparameter influences only for the selected embedding. See figure 5.12.

SP: Embedding Scatterplot Matrix

This scatterplot matrix shows the "actual" embedding, i.e. all record coordinates in the lower-dimensional space - see figure 5.13. $\binom{d}{2}$ scatterplots are generated to visualize every pair of d dimensions in the lower-dimensional space. Records can be selected with a rectangular selection of their containing area. A dropdown allows to color records by any of their attributes to aid the user in the construction of a semantic interpretation of the embedding's fidelity.

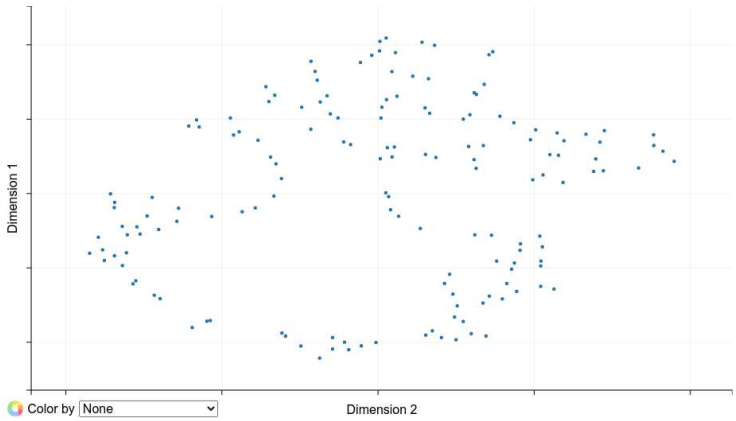


Figure 5.13: Component SP: Scatterplot or scatterplot matrix visualizing records in the low-dimensional space.

.

HMI: Metadata Attribute Indicators

A table with textual information on the precise attribute values in combination with histograms show the embedding's position in the examined parameter space. See figure 5.14.

RT: Record Table

All records in the original dataset are listed in the RT component. Similar to GT this table allows for free text search in all record properties seen in the table and sorting by any attribute. Hovered-over records are highlighted in component SP. Additionally it offers histograms over every non-categorical column[7] that provides range filters to limit the selection of records shown. See figure 5.15.

SD: Shepard Diagram

The Shepard diagram visualizes the consistency of the pairwise spatial distances[8] between high- and low-dimensional space. Due to the large number of combinations of records results are binned and displayed as an hexagonal heat map. See figure 5.16.

[7]Including histograms for categorical data disrupts the UI layout for some datasets, since they necessarily have to display all unique values and can't be arbitrarily binned like numerical data. Additionally, since range filters are not useful for categorical data, users would filter by clicking individual bars. This proved challenging due to the small size of these charts and was criticized in early user tests.

[8]See subsubsection 4.1.2 for more information.

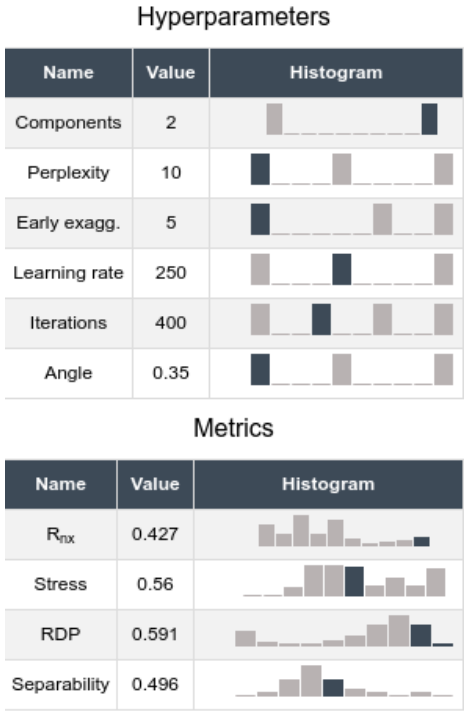


Figure 5.14: Component HMI: Table with histograms, reflecting properties of the currently viewed embedding. The column *value* contains the selected embedding's value for the corresponding attribute.

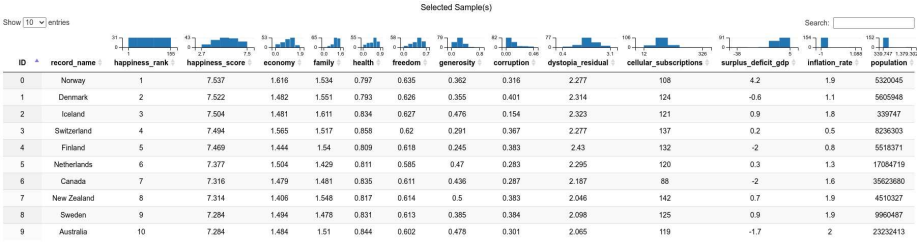


Figure 5.15: Component RT: Table showing all records with all of their attributes in the dataset.

CRM: Co-Ranking Matrix

The co-ranking matrix is closely related to SD. Other than the Shepard diagram however it is concerned with topological distances instead of spatial ones. See figure 5.17.

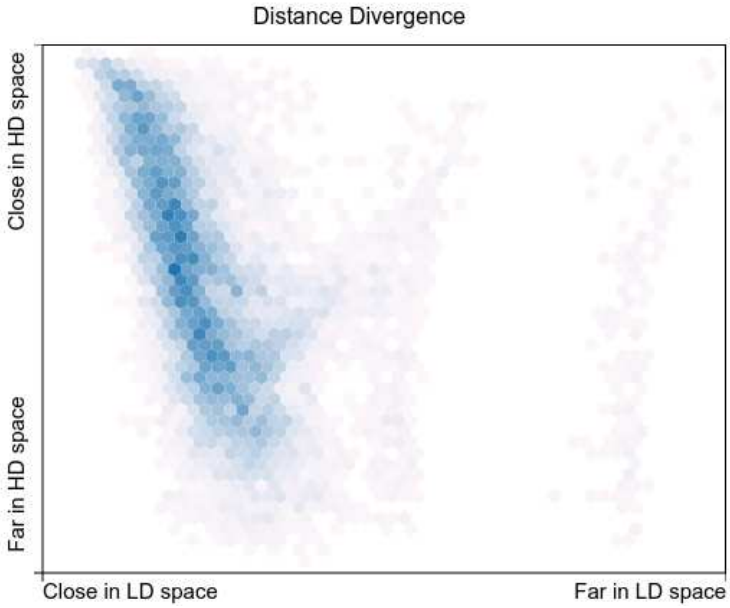


Figure 5.16: Component SD: A Shepard diagram representing the proportions between distances in the high- and those in the low-dimensional space.

RC: Title & Rating Indicator

A bar at the very top right allows rating the selected embedding on a scale from one to five, the latter being the best possible rating. Embeddings start without ratings - this is reflected by a bar with zero selected stars. Once an embedding has been rated, the rating can be modified in the permitted range at any point. It is not possible to revoke a rating and restore the embedding to unrated. See figure 5.18.

5.4 Interaction Flow and Usage Patterns

Our UI design aims to reflect the requirements specified in section 5.2. When devising the interface we intended to facilitate certain patterns of user interactions with the UI components as listed in table 5.1 to address all defined user tasks satisfactorily. We outline the conceived interaction flow as a sequence of grouped usage patterns here. We contextualize it with respect to the defined user tasks and discuss limitations subsequently.

1. **Parameter space exploration.** Initially the user strives to obtain an overview of the presented data. She garners information on the distributions of the embeddings' attributes, particularly metrics. She identifies interesting patterns w.r.t. hyperparameters and/or metrics.

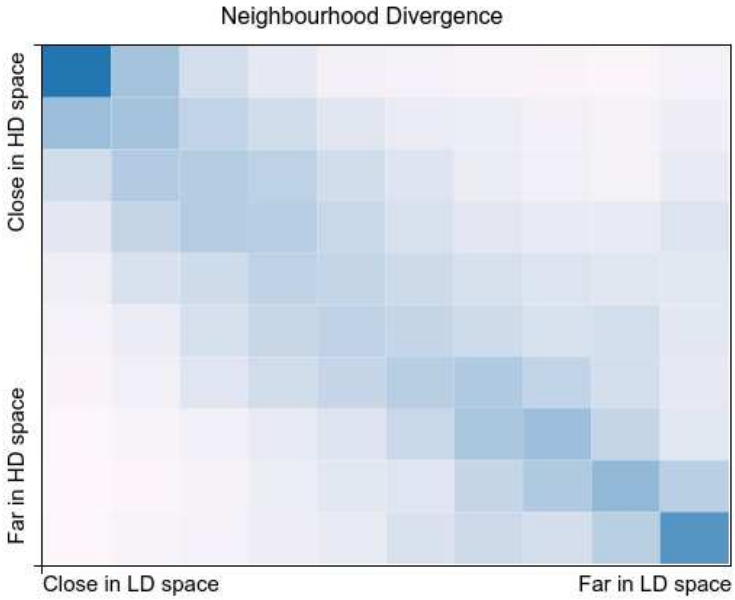


Figure 5.17: Component CRM: A co-ranking matrix representing how much neighbourhoods have changed throughout the dimensionality reduction process.

Figure 5.18: Component RC: The rating component allowing to assign a rating between 1 and 5 to a single embedding.

2. **Narrowing down selection.** Based on the insights gathered in step 1, the user decides which regions of the parameter space to limit her selection to. Various filters are employed to narrow down the set of embeddings to those that exhibit characteristics interesting to the user. Such characteristics might relate to the embeddings' metric values, e.g. the best or worst embeddings with regard to any given metric, or be part of a more complex pattern.

3. **Selecting an embedding to rate.** As a further step of filtering, a single embedding is chosen as the next embedding to inspect.

4. **Assess embedding quality and rate embedding.** The user inspects the selected embedding and relates it to her notion of a good representation of the data. She transforms her subjective judgment into a numerical rating assigns it to this embedding.

5. **Repeat.** Steps 1 to 4 are repeated as necessary or possible.

These five steps can be mapped directly to the clusters of user tasks introduced in the beginning of section 5.2 - steps 1 to 3 are covered by cluster $T_{NAV\text{-}PS}$, step 4 to T_{INSP}. $T_{INT\text{-}PS}$ is integrated orthogonally to support transparency and provide explanations.

It is to be expected that the actual usage of applications differs from user to user and might not match the designer's anticipations, since they are rooted in the limited experience gathered in earlier feedback rounds. Users' approach to using TALE can hence deviate considerably from the sequence laid out above. We report on the observed user behaviour and differences to the conceived flow of interactions in section 7.2.

5.5 Design as a Function of Evolving Requirements

The final user interface design for TALE is a product of multiple iterations of feedback rounds, spanning around two years and including a substantial deviation from the application's goals. The original intention was to devise a tool whose primary purpose was the interactive exploration of the hyperparameter space for dimensionality reduction algorithms. While this functionality remains an essential part of the application, the initial design emphasized the parameter space exploration aspect more strongly than the latest version, which prioritizes the usage of TALE for the rating of low-dimensional embeddings in the context of a multi-dimensional parameter space.

It also dedicated more screen space to XAI methods encapsulating the relationships between hyperparameters and metrics. This was achieved by fitting and visualizing interpretable surrogate models, specifically decision trees[9] and explanation rules[10]. The latter turned out to be be more practical for the purpose of demonstrating why embeddings were good or bad (i.e. exhibited high or low values for their metrics), since they emphasized the result, e.g. good or bad, through displaying it exactly once and concatenating it with the extracted explanation rules. Visualizing a decision tree however tends to emphasize the forking decision paths composing its structure due to the results being displayed as an arbitrary number of leafs and hence glyphs.

After the change in orientation some components, such as the approaches to global explanation introduced above, were discarded, whereas supporting users in estimating embeddings' fidelity was assigned a higher priority. Shifting the viewpoint to aiding users in the assessment of embeddings also meant subordinating the previously primary functions of open-end, global parameter space exploration to local, comparative embedding evaluation.

[9] A screenshot of an earlier prototype visualizing a decision tree as surrogate model can be seen in figure 7.3.

[10] We employed `skope-rules` [8] to extract the most influential de-duplicated decision rules from tree-based bagging ensembles.

6 | Implementation

In the following we present the implementation concisely from a high-level technical point of view. The codebase can be best understood as three separate components: (1) The process generating low-dimensional embeddings including all required metadata, (2) the user-facing application TALE and finally (3) the analysis of the data obtained from the user study. This chapter is structured accordingly. Note that this coincides with chapter 3 - hence the information offered for software components here can be cross-referenced with scope, purpose and restrictions of the workflow steps there.

All non-user facing parts are implemented in Python 3, everything else in Javascript. All non-standard software packages used are mentioned in the respective sections wherever appropriate.

6.1 Data Generating Process

We process raw datasets to generate enriched embedding information, which is detailed in the enumeration below. While producing this information on the fly - i.e. during the initialization and/or usage of TALE - would enable a considerable higher degree of flexibility, it is infeasible to do so without causing unacceptable long delays for the user. This applies even to use cases with small datasets due to the large number of sampled hyperparameter configurations.

- The actual embedding information, i.e. coordinates $C \in \mathbb{R}^m$ where $m < n$ is the dimension of the low-dimensional space compared to the original dimension m.
- A set of metrics measuring the quality of the embedding. While there is an abundance of other metrics available, the goal of visualizing them all in TALE sets a practical limit on the number of metrics we can reasonably show. We opted for three, each illuminating a different aspect of embedding quality.
  - NH_{AUC}, a topological quality measure based on the coranking matrix, introduced by Lee et al. [42]. We base our computation of this measure on a freely available Python package for the extraction of the coranking matrix from a distance matrix [37]. It is computed

by averaging R_{nx} over several sizes of neighbourhood k. R_{nx} is a normalized version of Q_{nx}, which measures the number of intruding and missing neighbourhood members. Intruding records are part of the k-neighbourhood only in the low-dimensional space; missing or extruding records only in the high-dimensional space. Both represent a deviation from the desired state, in which a record either is or is not a k-neighbour in both the high- and the low-dimensional space.

  - Kruskal's stress, a distance-based coranking measure first proposed by Kruskal [41].
  - Relative (target) domain performance (RDP) as measured by a task chosen specifically for each dataset. While this does not constitute a direct test for an embedding's faithfulness, it can serve as a proxy measure to capture the degree of predictive power a dimensionality-reduced dataset holds compared to its original state. It is computed as $RDP = DP_{low-dimensional}/DP_{high-dimensional}$[1], where DP denotes the domain performance, i.e. the performance of a machine learning model trained on this dataset for a task in the dataset's application domain.

- Explanatory data generated with SHAP. We train random forests as surrogate models to emulate the behaviour of the dimensionality reduction algorithm and apply SHAP's tree-based explainer to elicit instance-specific feature weights.

- Static metadata used in TALE, such as the dataset's number of records, attributes' data types, which attributes are targets for an (un-)supervised domain task, etc. This data is not generated anew, but merely packaged with the other, generated data.

The process yields this information for a multitude of hyperparameter configurations for different dimensionality reduction algorithms and different datasets. The exact specifications applied during evaluation are given in chapter 7.

We compute embeddings with three different algorithms, chosen for both their properties and popularity:

- Principal Component Analysis (PCA) by Pearson [58] is one of the most frequently used dimensionality reduction algorithms. It relies on linear, global assumptions and therefore assumes the same projections to be applied on every record in the dataset regardless of their neighbourhood graph. We incorporate the implementation in Pedregosa et al.'s `scikit-learn` package [59].

[1] $RDP < 1$ is to be expected in most cases due to the large number of dimensions being discarded of during dimensionality reduction for visualization purposes. $RDP > 1$ is also a possible case though - dimensionality reduction can act as noise filter and thus improve predictive power.

- t-distributed Stochastic Neighbourhood Embedding (t-SNE) by van der Maaten[79]. t-SNE is a manifold learning technique, i.e. it assumes that all points lie on a higher-dimensional manifold and approximating this manifold allows for finding faithful projections into lower-dimensional spaces. Unlike PCA it is not a linear projection. We integrate Ulyanov's parallelized implementation [78] of van der Maaten's Barnes-Hut-t-SNE [80].

- Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) by Leeland and Healy [54]. It is rather similar to t-SNE as it is a stochastic manifold learning technique as well. It places a stronger emphasis on performance, the preservation of global - i.e. inter-cluster - distances and its utilization as a preprocessing step instead of exclusively addressing visualization use cases. The authors' official implementation [52] is utilized.

6.2 TALE

TALE is realized with a server-client architecture to satisfy the constraints listed in subsection 3.2.2. The server backend is written in Flask [4], a Python-based REST framework. Most information served is pre-computed and merely loaded and rearranged by the backend to allow for a responsive usage of the tool with low latency between user actions. The server provides an initial template to the client, which proceeds by constructing structure and content of the application based on the loaded data, i.e. the charts and UI elements are generated partially dynamically with respect to the dataset's features. Data assumed to be static with respect to a single session is cached if beneficial to the performance.

The backend's functionality is detailed in table 6.1. All routes return data in the JSON format.

The frontend leverages the visualization framework `dc.js` [3] that emphasizes building sets of visualizations that are linked by Brushing and Linking: All manipulations of data filters in any chart are reflected in all other charts so that the individual views are always consistent with each other. `dc.js` is built on Bostock et al.'s `d3.js` [20]. `crossfilter.js` [2] acts as the linking mechanism and offers an assortment of charts with integrated support for linked views. The bulk of DOM creation and manipulation is handled by JQuery [6], supplemented by vanilla ES6 JavaScript[2]. Several other JavaScript libraries were taken advantage of for implementing various functionality and UI elements [63, 5, 35, 48, 1].

Since we strive for TALE to be easily portable, it is provided as a public Docker (see Merkel, [55]) image. This bypasses any need for the installation of required software on the user's side except the Docker software itself.

[2] In hindsight modern frontend JavaScript frameworks like `React.js` or `Angular` in lieu of `JQuery` would have been more suitable. `JQuery` was chosen purely for the sake of convenience due to the author's familiarity with it.

Name	Task
`/get_metadadata`	Reads data and metadata with embedding-level granularity for the specified combination of dataset and dimensionality reduction algorithm from disk. This includes embeddings' hyperparameter and dimensionality reduction metric values, number of features, target labels, results from explainer models etc. The loaded data is cached for later calls.
`/get_metadadata_template`	Loads a specification of the current dimensionality reduction algorithm's hyperparameters and the set of applied dimensionality reduction metrics.
`/get_dr_model_details`	Gathers data with record-level granularity for a single embedding: Record-wise embedding quality metrics, original and embedded coordinates, distances between records in the high-dimensional and low-dimensional space, explanatory SHAP values w.r.t. hyperparameters and dimensionality reduction metrics and the co-ranking matrix.
`/get_explainer_values`	Pulls up explanatory SHAP values with record-level granularity.
`/get_binned_pointwise_quality`	Retrieves embedding quality measure values with record-level granularity.
`/compute_correlation_strength`	Calculates Székely et al.'s [73] distance correlation between each combination of hyperparameter and dimensionality reduction metric with Carreño's implementation [23].

Table 6.1: Backend routes provided by the server and utilized by the frontend.

6.3 Data Analysis

Most functionality applied for the analysis of the obtained data is bundled in scikit-learn [59], e.g. its `svm.OneClassSVM` for outlier identification or `linear_model.LASSO` for LASSO regression. We also utilize various functionality from scipy [83], e.g. for the computation of distance matrices, and numpy. Clusters are detected using McInnes et al.'s HDBScan [53]. Ke et al.'s LightGBM [39] serves as the boosting model trained to predict perceived embedding quality from preservation metrics and visual quality criteria.

7 | Evaluation and Analysis

Two datasets were utilized for the user study and analysis phase:

1. The UN World Happiness Dataset from 2017 [32], which contains data on the satisfaction with a number of different factors such as overall happiness, the state of their health system, economy, freedom, etc. in 155 countries. These scores were assembled by polling a representative number of inhabitants in the corresponding countries. They are confined to an interval from 1 to 10, the latter representing the highest satisfaction.

2. A self-curated dataset collecting the 300 most popular movies, pulled from the IMDB API. Each movie is described by a set of attributes, such as popularity, budget, production country, tags describing the movies' contents, genres etc.

The usability and usefulness of TALE for the purpose of assessing embeddings was evaluated by conducting a user study with 15 participants, two of which were participating in the pilot study. Two participants were female, the remainder male. The average experience with dimensionality reduction was 1.46 on a scale from 0 (no prior knowledge) to 3, i.e. the average tester had a decent idea of dimensionality reduction and what it is used for. From the set of 15 participants three had an experience score between 0 and 1, four a score between 1 and 2, the remaining testers rated two to three - see figure 7.1.

This section additionally presents our findings on the alignment between perception-driven embedding ratings and existing dimensionality reduction metrics as well as an evaluation of the model trained to imitate human embedding rating behaviour (see section 6.3).

7.1 TALE - Formative Evaluation

Before as well as during the implementation phase, a series of feedback rounds and discussions amongst a small circle consisting of the author and two other contributors was essential in establishing and refining the fundamental requirements, tasks and design consideration for TALE. The original concept aimed at the parameter space analysis of word embeddings - the first sketch is shown

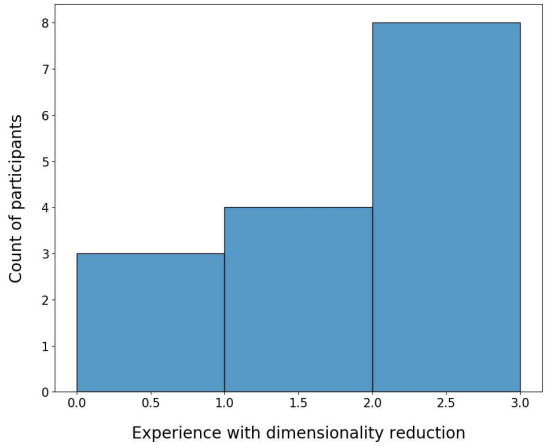


Figure 7.1: Experience with dimensionality reduction amongst study participants.

in 7.2. A discussion during one of these feedback rounds resulted in the conclusion that an effective visualization of word embeddings would require the application of dimensionality reduction algorithms, hence it would be more reasonable to tackle the parameter space analysis of those first. We consequently shifted and adapted the tool design for the exploration of low-dimensional, data type-agnostic (as opposed to high-dimensional word) embeddings.

Such early conversations and brainstorming sessions also had a significant impact on which components were cut and which ones remained. The limitation of available screen space was frequently topic of feedback rounds and design meetings. Figure 7.3 shows an early prototype, containing two visualizations not contained in the final version: An explanatory surrogate model based on decision trees and a heatmap representing the faithfulness of embeddings per record. Both of them were found to be unhelpful, confusing or less insightful than potential alternatives and thus dropped.

Later feedback rounds and informal discussions helped to align the tool's functionality with user needs and practical concerns, e.g. the capability to adjust settings or the lack of features necessary to render available visualizations useful to the user.

7.2 TALE - User Study

All 15 participants were given an introduction to the topic of dimensionality reduction and this thesis' goal, followed by a tour of TALE. After the tour and choosing one of the provided datasets, users were asked to select embeddings from a set of several hundred available ones, assess and rate them. A test lasted roughly one hour, with some cases exceeding this time span by up to 30

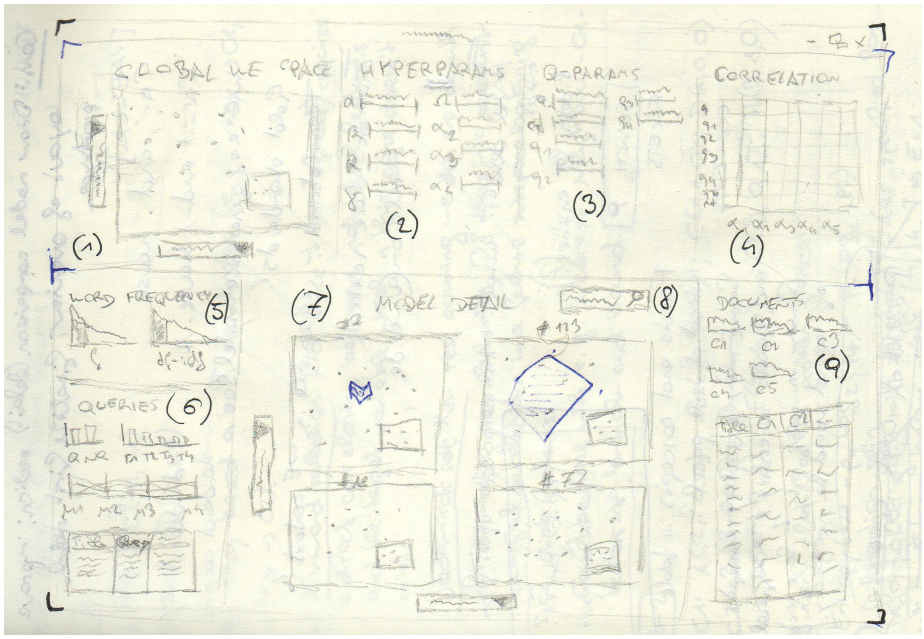


Figure 7.2: One of the first sketches, which was geared towards an application for the parameter space analysis of word embeddings.

Figure 7.3: An earlier prototype, including a decision-tree based surrogate model and a record-based embedding quality heatmap.

minutes, primarily due to technical issues[1]. In a single case the evaluation had to be postponed to the next day due to connectivity issues on the tester's side. After the introduction to the motivation behind the tool and the tour of the tool itself users rated on average four embeddings while thinking aloud.

Following the rating process the testers were asked to fill out the SUS questionnaire and optionally supply free form feedback. The cumulative SUS score averages to 75.63 with a median of 78.75 and a standard deviation of 9.05. This is considered a good score in the literature (compared to a literature-wide average of 70) according to Bangor et al. [13], reflecting a generally positive response from the study participants. figure 7.4 shows the distribution of ratings as box plot.

The most common free-form feedback was that (1) the tool offers useful components and visualizations to aid the user in interpreting and assessing embeddings and (2) the total set of functionality, some of which is purely explanatory and not essential to the actual task[2], can be overwhelming and too much to process in a single hour. Other commonly mentioned points of criticism and feature requests:

- Using scented histograms as for filtering was occasionally considered uncomfortable due to their small size.
- Multiple embeddings should be comparable at once.
- More information on the provided data and components should be displayed on hover.
- It should be possible to define certain rules that automatically check whether conditions are met in any given embedding (e.g. whether Denmark and Sweden are to be found in the same neighbourhood in the Happiness dataset).
- Ability to show similar/dissimilar embeddings w.r.t. their clustering structure or metric values.
- The scatterplot visualization should allow zooming, selection of (multiple) individual points and show record information on hover. Its color scales should be customizable.
- Some testers mentioned occasional lagging, flickering or screen freezes that could be resolved by a Teamviewer restart. No tester reported substantial negative impacts of connectivity issues on their experience though.
- The lack of a dark mode, which two testers mentioned to be desirable for prolonged tool usage.

[1] The user study took place during the Corona pandemic in spring 2020, so we were forced to conduct it remotely.

[2] I. e. the assessment of embeddings would be possible without those components. They were integrated to allow an interpretation and deeper understanding of the underlying models and algorithms. See chapter 5 for more information on this and the rationale behind it.

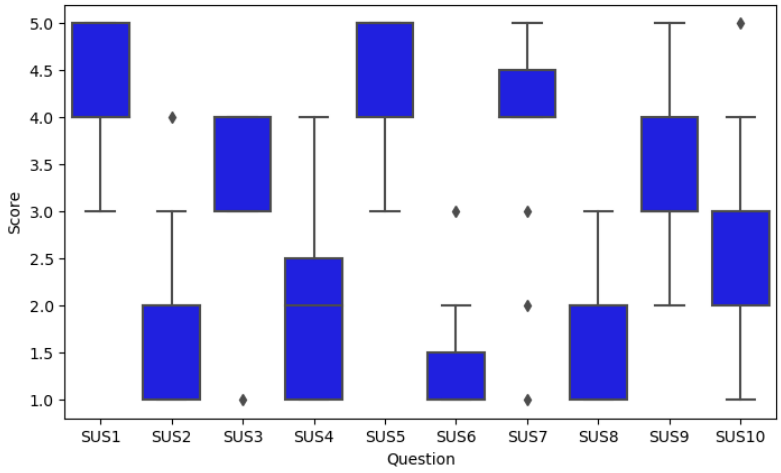


Figure 7.4: Distribution of SUS scores per question. See section 3.4.1 for a list of all questions.

Usage patterns Testers developed an unexpected diversity of usage patterns, i.e. selections of components and techniques to use them, to build up estimations of the embeddings' quality. A discernible trend however was that two tools - namely filtering in the scatterplot to select clusters and hovering over points in the record table to highlight them in the scatter plot - were almost always used. Other components, e.g. selecting records via the scented histogram filters, were less commonly used. When discussing the reasons for this, participants adhering to these patterns usually cited that they either forgot those other, less commonly used components because they felt overloaded by information due to the number of offered components or dismissed them because they didn't see their value since they could obtain the knowledge they were looking for by other means.

The overall feedback by testers was positive and encouraging - users appreciated the consistent brushing and linking and the tool enabled them to approach the problem from different angles by combining different techniques and components. A common sentiment was that this tool is a good expert tool - lots of well integrated functionality, but at times overloaded and requiring some practice in order to understand all of its functionality and figuring out strategies. The set time limit of roughly one hour per tests thus proved challenging. Lessons and remaining questions are discussed in section 8.2.

7.3 Analysis of Gathered Embedding Ratings

This section discusses the outcome of our analysis and its implications. It has to be stressed that these results are of limited significance due to the small number of embedding rating in our datasets (79 ratings by 16 users). To obtain results that are robust towards personal and topological variance would likely require a number of ratings at least one order of magnitude higher as well as a larger number of datasets to be embedded. The evaluation and interpretation done in this section however is still of value, as it demonstrates a methodology to do so and highlights patterns worth investigating in a more comprehensive study.

7.3.1 Correlation between Perceived Embedding Quality and Preservation Metrics

The obtained PEQ scores were correlated with the computed preservation metrics using Pearson's, Kendall's and Spearman's correlation. Figure 7.5 displays embeddings with a preservation metric on the x- and the PEQ score on the y-axis. The results show that PEQ scores are most strongly correlated with NH_{AUC}, followed by Kruskal's stress and relative predictive power. The strongest relationship with PEQ is a Spearman correlation value of 0.7 with NH_{AUC}. Interestingly, this is very close to the agreement score of 0.715 that Bibal and Frénay [18] report in their investigation of interpretability of labeled embeddings[3]. An informal rule of thumb prevalent in statistics is that correlations above 0.7 are considered strong[4]. Thus it can be concluded, with the caveat of the limited sample size, that users are able to identify quality in embeddings somewhat accurately within the confines of our experimental setup. When considering the implications, we have to take into account that (1) our tool provides functionality to analyse embedding visualizations that common (oftentimes static) embedding visualizations don't and (2) even with all the analytical functionality in our tool, these results show a considerable gap that is not explained by any preservation metric.

7.3.2 Predicting PEQ

Preceding the actual model training and interpretation, the features introduced in 4.3 were computed for each generated and rated embedding. The feature set was complemented by some of the preservation metrics discussed in subsection 4.1.2, specifically NH_{AUC}, B_{nx}[5], Kruskal's stress and relative target domain

[3] Their approach differs from ours, hence these two numbers must not be compared without taking these differences into account. In our approach, the correlation between PEQ score and NH_{AUC} is measured; in Bibal and Frénay's case it is how often NH_{AUC} correctly indicated a users preference of one embedding over the other. In a more abstract sense they both measure how strongly NH_{AUC} agrees with the subjective, perceived quality of embeddings.

[4] Sources for the exact values are unclear, but see for example Ratner [60] for a confirmation.

[5] B_{nx} is a topological measure describing an embedding's tendency to pull non-members into neighbourhoods or to push members from their original neighbourhood. It represents which kind of error occurs more often in this embedding and is derived from the same topo-

performance[6]. Subsequent to the feature engineering a LightGBM and a LASSO regression model were employed, with rather similar results. The data set was split into a training and test set, with a validation set split from the training set when applying hyperparameter optimization. The resulting metrics were averaged over 20 runs with random seeds. We trained several models with the following composition of features: (1) All features, (2) all preservation metrics, (3) all visual quality criteria and (4) each attribute on its own. Model (1) is analyzed further in regards of feature importance and other factors below.

Figure 7.6 shows the results as measured by the mean absolute error on all of the feature groups desribed above. Our results indicate that NH_{AUC} is the strongest predictor of perceived embedding quality in the set of preservation metrics we investigated. This confirms Bibal and Frénay's results [18], for which they compared NH_{AUC} with three of the best-performing - as shown by Sedlmair and Aupetit [67] - visual quality criteria used for gauging the perceived quality of embeddings: Average between-within (ABW), distance consistency (DSC) and hypothesis margin (HM). Since the latter two assume the supervised case of each point being assigned exactly one class and we don't, we implemented unsupervised variants (see table 4.1). Unsupervised HM by itself is the strongest predictor overall, while ABW and unsupervised DSC do not score remarkably well. Notably, some models with only one feature perform better than some models with several or even all features, which suggests that the number of features compared with the number of datapoints is too high. The true efficacy of models with higher number of features remains to be verified with larger datasets.

Overall, the visual quality criteria computed to serve as features for each embedding narrow the gap, as evident by unsupervised HM performing best and the model with all features performing second-best, ahead of the model based only on preservation metrics. While our engineered features seem to contribute, the difference is too marginal to claim statistical significance. The preliminary results w.r.t. using visual quality criteria as features to model perceptual embedding quality thus show promise, but don't seem to improve predictions dramatically. More definite conclusions can only be drawn with a larger number of embedding ratings.

Interpreting the PEQ Model

The feature importance in the model operating on the entire feature set is shown in figure 7.7. Feature importance here is computed utilizing SHAP by adding all absolute SHAP values w.r.t. records' PEQ scores. Notably, preservation metrics are the strongest features - however, some of the engineered visual quality criteria contribute significantly as well. The SHAP summary plot 7.8 reveals mostly expected behaviour, such as high values for stress or low values for ABW

logical framework as NH_{AUC}.

[6] I.e. the ratio of approximative predictive power in the low-dimensional and the high-dimensional space. See sub-sub-section 4.1.2 for more detailed information.

being detrimental to the prediction accuracy.

The ordering of features by importance measured by SHAP does not match the ordering of features based on the error rate of their respective single-feature models exhibited in 7.6. This is most likely due to some degree of multicollinearity in the feature set. Recommendations to tackle multicollinearity usually involve obtaining more data or discarding features with a high variance inflation factor (VIF) ≥ 5. Since the former was not practically feasible due to resource constraints, we dropped all columns with $VIF \geq 5$.

This improved the median absolute error by about 5% from 0.63 to 0.6 and, more importantly, yielded feature importance scores that were almost perfectly in line with the ordering obtained from single-feature models. See figure 7.9 for a plot of feature importance for a model including all features with a qualified VIF score. It cannot be claimed that there is a direct causal relationship between these features and PEQ. However, these features form the subset found to be most informative and concise for the task of predicting PEQ scores, probably due to them covering different semantic and topological aspects relevant to the PEQ. Remarkably, all of the preservation metrics except for b_{nx} were dropped in the course of the collinearity reduction, despite having been assigned the highest feature importance prior to it. This indicates a high degree of correlation between those preservation metrics and some of the other, preserved features.

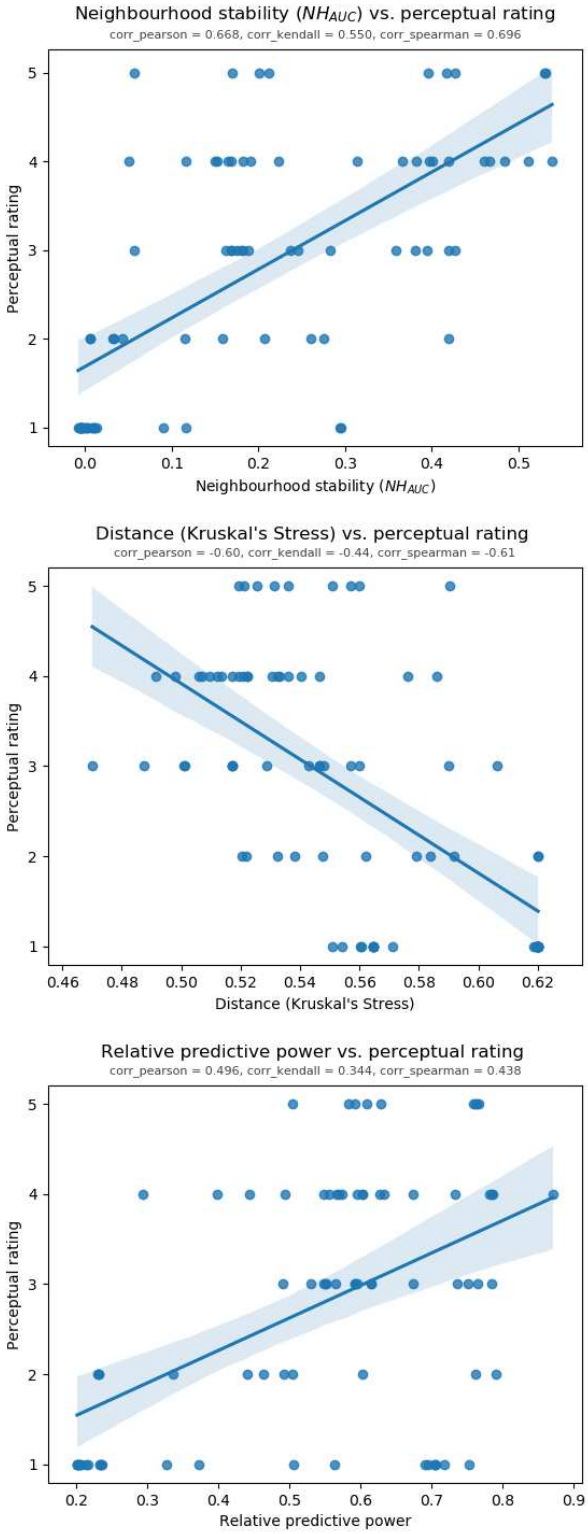


Figure 7.5: Correlation between PEQ scores and preservation metrics.

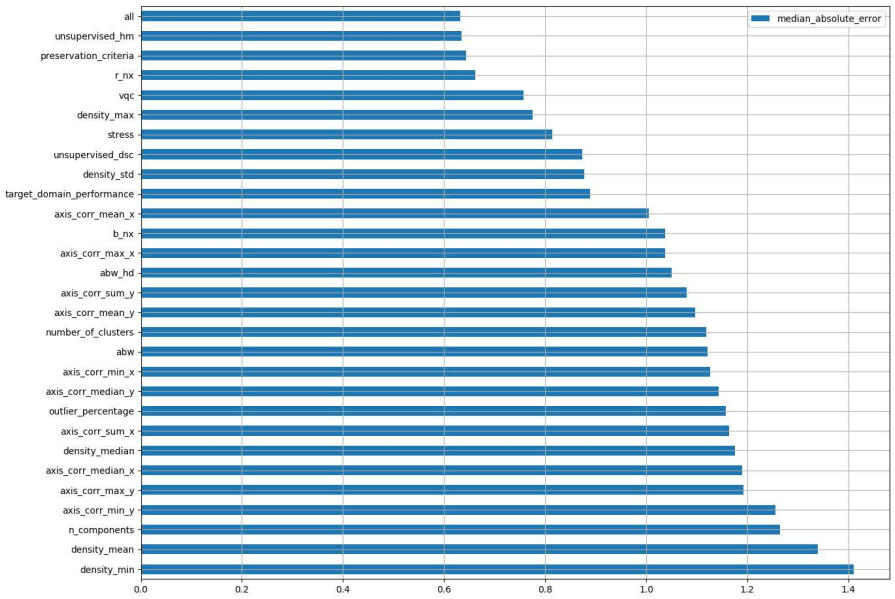


Figure 7.6: Median absolute error scores for LightGBM boosting models with varying feature sets.

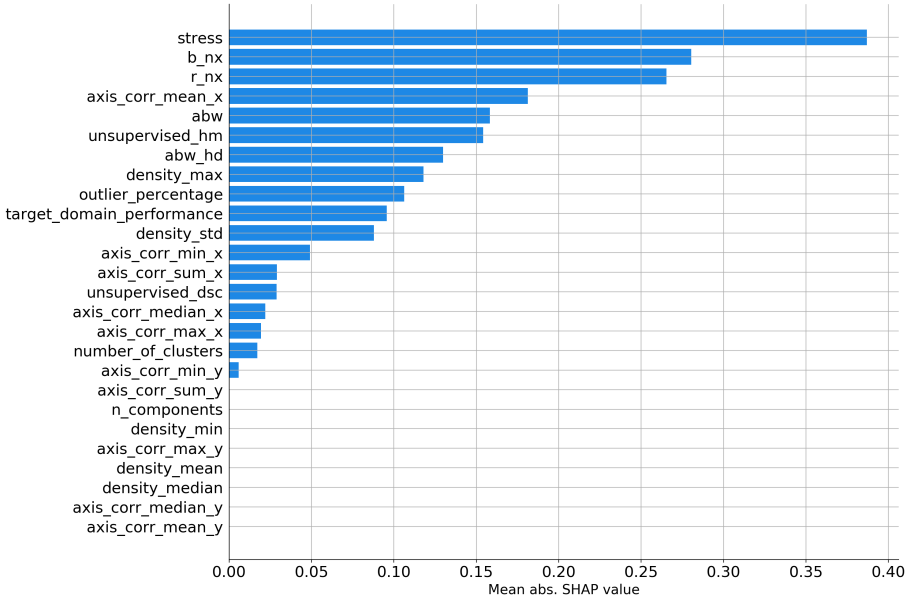


Figure 7.7: SHAP-based feature importance for a LightGBM model with all features.

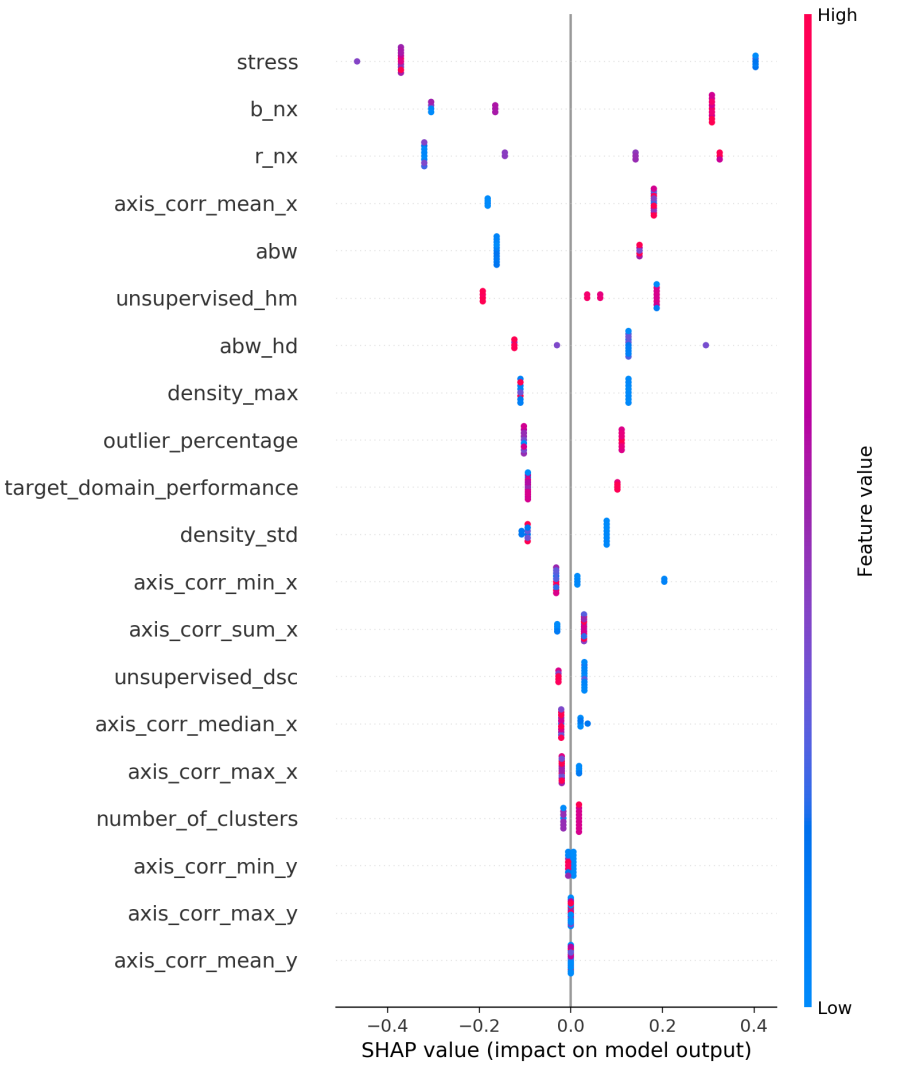


Figure 7.8: Summary of SHAP values per record and feature for a LightGBM model with all features. Each point represents one feature value for one record, i.e. an embedding with an user-assigned rating. Points on the left indicate that the corresponding, color-coded feature value contributed negatively to the embedding's rating, points on the right the opposite. E.g.: Most embeddings with high values (colored red) for stress are placed on the left - it can be concluded that a high value for stress contributed negatively to an embedding's perceived quality in most cases.

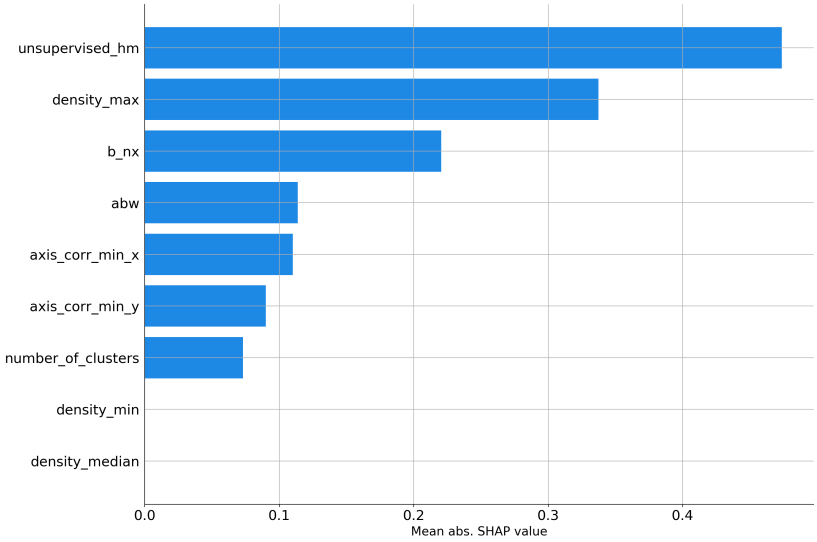


Figure 7.9: SHAP-based feature importance for a LightGBM model with all features with $VIF < 5$ to reduce multicollinearity.

8 Discussion

This project's primary goal was to investigate how accurately users estimate the quality of low-dimensional embedding's through interaction with their visualizations. The secondary goal was to determine which factors are how influential in the users' perceptual process when doing so. Achieving these goals entailed several steps:

- Design and build a tool (TALE) that provides functionality for the exploration, interpretation and rating of embeddings. Conduct a user study and evaluate the results.
- Obtain embedding ratings (using TALE).
- Select or design, implement and compute visual quality criteria for all rated embeddings.
- Compute the degree of correlation between PEQ scores and computed preservation metrics and visual quality criteria.
- Build a machine learning model predicting PEQ scores from computed preservation metrics and visual quality criteria.

Approaches for a study with these or highly similar goals have been proposed in the literature before (see e.g. Bibal and Frénay [19]). However, for the unsupervised case[1] neither applications designed to obtain embedding rating data nor such data itself have been made available yet.

For the sake of clarity we discuss our results separately in three parts: First, the implications of our analysis; second, design lessons w.r.t. TALE and the user feedback we received; third, limitations of our work and how to improve it.

8.1 Implications of the Conducted Analysis

First and foremost it has to be restated that due to the low number of embedding ratings our analysis results have to be understood as preliminary. A conclusive

[1] I.e. embeddings in which records do not have at least one categorical label. Assuming that records have a unique class mapping simplifies the analysis process significantly, but many use cases for embedding visualizations don't involve class labels.

analysis necessitates obtaining considerably more ratings from more users on more datasets. Nonetheless some conclusions potentially interesting for follow-up analyses can be drawn.

One of these is that preservation metrics, which can be regarded as measures of an embedding's faithfulness and hence indicators of "objective" quality, and PEQ have a correlation coefficient of roughly 0.7. As discussed in section 7.3, this implies that a substantial part of PEQ scores is not explained by an embedding's faithfulness. This applies even with the battery of analytical and interpretative functionality provided by TALE, which facilitates understanding and assessing embedding visualizations. We hypothesize that the correlation between PEQ and preservation metrics for the common use case of a static two-dimensional embedding visualization without any this supporting functionality is lower, i.e. that users' ability to estimate an embedding's objective quality is diminished. We therefore, based on the conducted analysis and the prior hypothesis, consider it appropriate to recommend the integration of some kind of visual error indication and/or statistical measures that highlight the global and local quality of embeddings. This might prevent or at least mitigate erroneous conclusions drawn by users after seeing visualizations of inaccurate embeddings.

A second conclusion is that an analysis of the trained PEQ-predicting machine learning models yields that some visual quality criteria, such as the unsupervised hypothesis margin, improve the prediction of PEQ scores. They thus narrow the explainability gap, i.e. the degree to which PEQ cannot be explained yet. None of the introduced visual quality criteria however drastically improved the model's predictive abilities though. While, again, our conclusions are not particularly robust and findings might differ with more and more diverse embedding and ratings, this might indicate the need for other visual quality criteria.

8.2 User Interface Design Lessons

While the feedback received by testers for TALE was generally favourable, there were some features and design decisions that quite clearly didn't work the way we intended them to. The most prominent point of criticism related to the amount of functionality and components - many users felt initially overwhelmed, especially given the limited time frame of one hour. In particular the parameter space exploration (PSE) features were often dismissed, since they were not central to the core task of understanding and rating embeddings. Originally these components were designed to keep users engaged, offer context and help understanding the relationships between hyperparameters and targets, i.e. preservation metrics. Users, except for one or two, hardly ever engaged with these components, which allows several possible - not necessarily exclusive - conclusions:

- The time frame was too short. It is possible that understanding the black-box activity of dimensionality reduction algorithms tends to become more interesting when involved with the assessment process a prolonged amount of time.

- The design of the user interface and/or the introduction were not clear and intuitive enough for users to engage with this functionality. About half of the testers, in fact, asked for clarification regarding the PSE functionality.
- While offering context might actually be helpful and interesting to users in this context also in the short term, the subject of this context - i.e. PSE - might have been too far removed from the central task, which was understanding and assessing individual embeddings.

A future iteration of the user interface would most likely drop or at least reduce support for PSE activities and focus more strongly on the core task.

The second central point of criticism concerned the lack of possibility to compare embeddings with each other. Closely related, testers brought up the lack of a baseline based on which to rate embeddings, especially when starting their rating process. This resulted in a lower degree of confidence for the rating of the first few embeddings rated. While this lack of a baseline was resolved after they inspected a few embeddings, it would still be preferable to eliminate this issue completely.

Both of these issues - no possibility for a direct comparison and the lack of a baseline when starting to rate - are rooted in a cornerstone of our UI design, which is the detail view representing exactly one embedding and tasking users to assess it. The received feedback clarifies that this is, if not insufficient, certainly unsatisfying. A redesign would hence integrate the possibility to compare at least two embeddings side-by-side.

In summary, user feedback has demonstrated that while the tool overall serves it purpose well, some central elements are not ideally designed, selected and combined. This section discussed the principal flaws and the components in need of being reworked in order to resolve the flaws. Further, more radical ideas and approaches for a more effective and efficient UI are discussed in section 8.3.

8.3 Limitations and Future Work

8.3.1 Limitations

The major limitation of this work, as pointed out repeatedly, is the small size of our evaluation dataset with 79 embedding ratings from 16 testers on two datasets. While sufficient to demonstrate the feasibility of the analysis methodology and process and extract some interesting patterns, it does not allow to make unconstrained statements about user behaviour beyond the boundaries of our experimental setup.

A second shortcoming is that our current approach assumes that all user behaviour w.r.t. the perception of embedding quality can be satisfyingly be described by a single model. The veracity of this assumption is questionable and would have to be investigated in a larger experimental setting. A model might also be capable of differentiating between types of users and thus infer their

behaviour, if it is able to generalize well enough and trained on an adequately sized dataset.

8.3.2 Future Work

Feature Extraction and Engineering

We deliberated on the magnitude of improvements w.r.t. the prediction of PEQ scores achieved by the incorporation of visual quality criteria in section 8.1. Further improvements might be made by selecting or designing more visual quality criteria and integrating them into the model. Alternative approaches might circumvent such proxy measures of an embedding's topological structure and instead try to learn from the raw topological structure directly, e.g. by training graph neural networks to infer PEQ values. This might render further feature engineering obsolete. However, leveraging such approaches requires markedly more data and would most likely be more difficult to interpret. In a similar vein, established graph features can be extracted automatically and used as features[2].

Active Learning for more Efficient Labeling

Active learning models should be able to select those embeddings whose rating would yield the highest benefit in terms of the efficacy and efficiency in the prediction of perceived embedding quality. As brought up in 5.2, users must have agency however - so the embeddings selected as next embedding to be rated by an active learning model would remain a recommendation, not an obligation. This might help to improve the accuracy in predicting perceived embedding quality without diminishing user engagement.

Individualization by User Profiles

Differences in the perception of embedding quality between users might be large enough as to warrant adjusting predictive models to individual users or groups of users. Subsection 8.3.1 raises the issue that this is not taken into account by the current single-model approach. This might deteriorate prediction quality and impact the fidelity of the corresponding model explanations, i.e. which factors play which role in the perceptual process. If the model is not able to generalize well enough due to large behavioural differences between users, a potential remedy might be to create different user profiles - one for novice users, one for experts, etc. Each of these profiles would be trained on the subset of ratings assigned by the corresponding set of users. This was suggested by Bibal and Frénay [19].

[2]Various software packages and libraries exist for this purpose, e.g. GraphRole by Kaslovsky et al. [38], which implements graph mining algorithms by Henderson et al. [33, 34]

UI Design

Shortcomings of our UI design are examined in sections 7.2 and 8.2. It is evident that TALE would benefit from a redesign of some central components, chief amongst them that the lack of direct comparison between embeddings and the strong emphasis on parameter space exploration. A thorough redesign might abandon the current global-to-local view paradigm and eliminate the global view entirely. A comparison between at least two embeddings should always be possible so that users can select one embedding over the other, thus expressing their preferences and not assigning a numeric rating. This might solve the problem of an initial baseline. Drawing on our experience, we are thus inclined to favor a preference-based rating system as used by Bibal and Frénay [18] over one working with numerical scales like ours for future work. However, selecting preferences should be integrated into a UI that offers interpretative and analytical functionality like TALE.

Gather More Embedding Ratings

The most obvious and urgent point to address is the lack of a sufficiently sized set of embedding ratings. Therefore a larger experimental setup with more users, more datasets and possible more embeddings produced by other dimensionality reduction algorithms should be conducted. While the possible improvements elaborated upon above would likely improve user experience and reduce noise in the obtained embedding ratings, we consider the tool in its current state to be sufficiently mature and useful. Therefore we consider it a valid course of action to execute this larger study even prior to the realization of any of these improvements.

9 | Conclusion

This thesis addresses the question to which degree perceived and objective embedding quality correlate and, as a secondary objective, which factors impact users' perception of embedding quality in which way. No class labels were presupposed in this work, which sets it apart from preceding literature. The necessary steps and contributions are summarized in the following.

Utilizing a tool we designed and developed specifically for this task (*T*ool for the *A*nnotation of *L*ow-dimensional *E*mbeddings, TALE), we obtained the perceived quality of low-dimensional embeddings from 16 participants. TALE combines interpretative and analytical functionality for the assessment of unsupervised[1] embeddings' quality on a global and local level with components for parameter space exploration to investigate the relationships between hyperparameters and embedding quality. The user study yielded generally favourable feedback with a median SUS score of 78.75.

We investigated to which degree perceived embedding quality (PEQ) correlates with objective embedding quality, measured by three preservation metrics commonly used to capture an embedding's faithfulness and capability for information preservation. Correlation was measured to be about 0.7, indicating that a substantial part of PEQ cannot be explained by objective embedding quality. Leaning on prior work like Bibal and Frénay [18, 19], we hypothesized that certain topological properties can contribute to the explanation of embeddings' PEQ scores, since they partially represent their attractiveness to the user and/or degree of interpretability.

In order to capture those properties, we integrated a series of well-performing visual quality criteria for cluster and class separation, modifying them for the unsupervised case where necessary. We trained a machine learning model to predict PEQ with varying feature sets. In analyzing the trained models, we found that leveraging visual quality criteria together with preservation metrics beat models learning only from the latter in terms of prediction accuracy, confirming our hypothesis. However, due to the small difference of about 6% and the small sample size of 79 embedding ratings, these findings have to be considered preliminary and inconclusive.

The contributions of this work are as follows:

[1] I.e. an embedding of entities without class labels.

1. TALE, including its design, implementation, user study and a discussion of design lessons and potential improvements.

2. The data on perceived embedding quality we obtained and made available.

3. Our analysis and the underlying methodology. This includes the selection, modification and implementation of visual quality criteria for the estimation of various visual and semantic aspects of unlabeled embeddings. Albeit the outcome of our analysis is of limited robustness due to the amount of available rating data and hence to be considered preliminary, it provides insights into which factors wield which influence w.r.t. perceived embedding quality.

Bibliography

[1] Split: Unopinionated utilities for resizeable split views, https://github.com/nathancahill/split, last accessed on: 2020-02-03

[2] Crossfilter: Crossfilter is a javascript library for exploring large multivariate datasets in the browser (Feb 2020), https://github.com/crossfilter/crossfilter, last accessed on: 2015-06-02

[3] dc.js: Dimensional charting built to work natively with crossfilter rendered using d3.js (Feb 2020), https://github.com/dc-js/dc.js, last accessed on: 2012-06-19

[4] Flask: The python micro framework for building web applications (Feb 2020), https://github.com/pallets/flask, last accessed on: 2010-04-06

[5] intro.js: A better way for new feature introduction and step-by-step users guide for your website and project (Feb 2020), https://github.com/usablica/intro.js, last accessed on: 2013-03-10

[6] jQuery: jQuery JavaScript Library (Feb 2020), https://github.com/jquery/jquery, last accessed on: 2009-04-03

[7] Scree plot (Jan 2020), https://en.wikipedia.org/w/index.php?title=Scree_plot&oldid=934407889, page Version ID: 934407889

[8] skope-rules: Machine learning with logical rules in Python (Jan 2020), https://github.com/scikit-learn-contrib/skope-rules, last accessed on: 2018-02-18

[9] Abdelmoula, W.M., Balluff, B., Englert, S., Dijkstra, J., Reinders, M.J.T., Walch, A., McDonnell, L.A., Lelieveldt, B.P.F.: Data-driven identification of prognostic tumor subpopulations using spatially mapped t-SNE of mass spectrometry imaging data. Proceedings of the National Academy of Sciences of the United States of America **113**(43), 12244–12249 (Oct 2016). https://doi.org/10.1073/pnas.1510227113, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5087072/

[10] Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)

[11] Amershi, S., Cakmak, M., Knox, W.B., Kulesza, T.: Power to the people: The role of humans in interactive machine learning. AI Magazine **35**(4), 105–120 (2014)

[12] Arik, S.O., Pfister, T.: Tabnet: Attentive interpretable tabular learning. arXiv preprint arXiv:1908.07442 (2019)

[13] Bangor, A., Kortum, P., Miller, J.: Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of Usability Studies **4**(3), 114–123 (2009)

[14] Becht, E., McInnes, L., Healy, J., Dutertre, C.A., Kwok, I.W.H., Ng, L.G., Ginhoux, F., Newell, E.W.: Dimensionality reduction for visualizing single-cell data using UMAP. Nature Biotechnology **37**(1), 38–44 (Jan 2019). https://doi.org/10.1038/nbt.4314, https://www.nature.com/articles/nbt.4314

[15] Becker, R.A., Cleveland, W.S.: Brushing scatterplots. Technometrics **29**(2), 127–142 (1987)

[16] Behrisch, M., Blumenschein, M., Kim, N.W., Shao, L., El-Assady, M., Fuchs, J., Seebacher, D., Diehl, A., Brandes, U., Pfister, H., et al.: Quality metrics for Information Visualization. In: Computer Graphics Forum. vol. 37, pp. 625–662. Wiley Online Library (2018)

[17] Bertini, E., Tatu, A., Keim, D.: Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization. IEEE Transactions on Visualization and Computer Graphics **17**(12), 2203–2212 (Dec 2011). https://doi.org/10.1109/TVCG.2011.229

[18] Bibal, A., Frénay, B.: Learning interpretability for visualizations using adapted Cox models through a user experiment. arXiv preprint arXiv:1611.06175 (2016)

[19] Bibal, A., Frénay, B.: Measuring quality and interpretability of dimensionality reduction visualizations. In: Safe Machine Learning Workshop at ICLR (2019)

[20] Bostock, M., Ogievetsky, V., Heer, J.: D^3 data-driven documents. IEEE transactions on visualization and computer graphics **17**(12), 2301–2309 (2011)

[21] Brooke, J., et al.: SUS - A quick and dirty usability scale. Usability Evaluation in Industry **189**(194), 4–7 (1996)

[22] Carr, D.B., Littlefield, R.J., Nicholson, W., Littlefield, J.: Scatterplot matrix techniques for large n. Journal of the American Statistical Association **82**(398), 424–436 (1987)

[23] Carreño, C.R.: dcor (Jan 2020), https://github.com/vnmabus/dcor, last accessed on: 2017-09-13

[24] Cattell, R.B.: The scree test for the number of factors. Multivariate behavioral research **1**(2), 245–276 (1966)

[25] Cunningham, J.P., Ghahramani, Z.: Linear dimensionality reduction: survey, insights, and generalizations. The Journal of Machine Learning Research **16**(1), 2859–2900 (Jan 2015), http://dl.acm.org/citation.cfm?id=2789272.2912091

[26] El-Assady, M., Sevastjanova, R., Sperrle, F., Keim, D., Collins, C.: Progressive Learning of Topic Modeling Parameters: A Visual Analytics framework. IEEE Transactions on Visualization and Computer Graphics **24**(1), 382–391 (2017)

[27] Fernstad, S.J., Shaw, J., Johansson, J.: Quality-based guidance for exploratory dimensionality reduction. Information Visualization **12**(1), 44–64 (2013)

[28] Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of Statistics pp. 1189–1232 (2001)

[29] Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin based feature selection-theory and algorithms. In: Proceedings of the twenty-first international conference on Machine learning. p. 43 (2004)

[30] Gracia, A., González, S., Robles, V., Menasalvas, E.: A methodology to compare Dimensionality Reduction algorithms in terms of loss of quality. Information Sciences **270**(Supplement C), 1–27 (Jun 2014). https://doi.org/10.1016/j.ins.2014.02.068, http://www.sciencedirect.com/science/article/pii/S0020025514001741

[31] Hearst, M.A., Baeza-Yates, R., Ribeiro-Neto, B.: User interfaces and visualization. Modern information retrieval pp. 257–323 (1999)

[32] Helliwell, J.F., Huang, H., Wang, S.: The social foundations of world happiness. World happiness report **8** (2017)

[33] Henderson, K., Gallagher, B., Eliassi-Rad, T., Tong, H., Basu, S., Akoglu, L., Koutra, D., Faloutsos, C., Li, L.: RolX: Structural role extraction & mining in large graphs. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1231–1239 (2012)

[34] Henderson, K., Gallagher, B., Li, L., Akoglu, L., Eliassi-Rad, T., Tong, H., Faloutsos, C.: It's who you know: graph mining using recursive structural features. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 663–671 (2011)

[35] Ineshin, D.: IonDen/ion.rangeSlider (Feb 2020), https://github.com/IonDen/ion.rangeSlider, last accessed on: 2013-01-22

[36] Jäckle, D., Stoffel, F., Mittelstädt, S., Keim, D.A., Reiterer, H.: Interpretation of Dimensionally-reduced Crime Data : A Study with Untrained Domain Experts. pp. 164–175 (Feb 2017). https://doi.org/10.5220/0006265101640175, https://kops.uni-konstanz.de/handle/123456789/39720

[37] Jackson, S.: Co-ranking: Python implementation of a co-ranking matrix and various metrics derived from it (May 2017), https://github.com/samueljackson92/coranking, last accessed on: 2015-05-12

[38] Kaslovsky, D.: GraphRole: Automatic feature extraction and node role assignment for transfer learning on graphs (Jul 2020), https://github.com/dkaslovsky/GraphRole, last accessed on: 2018-10-24

[39] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems. pp. 3146–3154 (2017)

[40] Kobak, D., Berens, P.: The art of using t-SNE for single-cell transcriptomics. Nature communications **10**(1), 1–14 (2019)

[41] Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika **29**(1), 1–27 (Mar 1964). https://doi.org/10.1007/BF02289565, https://doi.org/10.1007/BF02289565

[42] Lee, J.A., Renard, E., Bernard, G., Dupont, P., Verleysen, M.: Type 1 and 2 mixtures of Kullback–Leibler divergences as cost functions in dimensionality reduction based on similarity preservation. Neurocomputing **112**, 92–108 (2013)

[43] Lee, J.A., Verleysen, M.: Quality Assessment of Dimensionality Reduction: Rank-based Criteria. Neurocomputing **72**(7-9), 1431–1443 (Mar 2009). https://doi.org/10.1016/j.neucom.2008.12.017, http://dx.doi.org/10.1016/j.neucom.2008.12.017

[44] Lee, J.A., Verleysen, M.: Scale-independent quality criteria for dimensionality reduction. Pattern Recognition Letters **31**(14), 2248–2257 (Oct 2010). https://doi.org/10.1016/j.patrec.2010.04.013, http://www.sciencedirect.com/science/article/pii/S0167865510001364

[45] Lewis, J., Ackerman, M., de Sa, V.: Human cluster evaluation and formal quality measures: A comparative study. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 34 (2012)

[46] Lewis, J., Van der Maaten, L., de Sa, V.: A behavioral investigation of dimensionality reduction. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 34 (2012)

[47] Liu, S., Maljovec, D., Wang, B., Bremer, P., Pascucci, V.: Visualizing High-Dimensional Data: Advances in the Past Decade. IEEE Transactions on Visualization and Computer Graphics **23**(3), 1249–1268 (Mar 2017). https://doi.org/10.1109/TVCG.2016.2640960

[48] Luca: LCweb-ita/LC-switch (Jan 2020), https://github.com/LCweb-ita/LC-switch, last accessed on: 2015-01-24

[49] Lueks, W., Mokbel, B., Biehl, M., Hammer, B.: How to Evaluate Dimensionality Reduction? - Improving the Co-ranking Matrix. arXiv:1110.3917 [cs] (Oct 2011), http://arxiv.org/abs/1110.3917, arXiv: 1110.3917

[50] Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems. pp. 4765–4774 (2017)

[51] Martins, R.M., Coimbra, D.B., Minghim, R., Telea, A.C.: Visual analysis of dimensionality reduction quality for parameterized projections. Computers & Graphics **41**, 26–42 (Jun 2014). https://doi.org/10.1016/j.cag.2014.01.006, http://www.sciencedirect.com/science/article/pii/S0097849314000235

[52] McInnes, L.: UMAP: Uniform Manifold Approximation and Projection (Jan 2020), https://github.com/lmcinnes/umap, last accessed on: 2017-07-02

[53] McInnes, L., Healy, J., Astels, S.: hdbscan: Hierarchical density based clustering. Journal of Open Source Software **2**(11), 205 (2017)

[54] McInnes, L., Healy, J., Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [cs, stat] (Feb 2018), http://arxiv.org/abs/1802.03426, arXiv: 1802.03426

[55] Merkel, D.: Docker: Lightweight linux containers for consistent development and deployment. Linux journal (239), 2 (2014)

[56] Mühlbacher, T., Piringer, H., Gratzl, S., Sedlmair, M., Streit, M.: Opening the black box: Strategies for increased user involvement in existing algorithm implementations. IEEE Transactions on Visualization and Computer Graphics **20**(12), 1643–1652 (2014)

[57] Nguyen, L.H., Holmes, S.: Ten quick tips for effective dimensionality reduction. PLoS Computational Biology **15**(6), e1006907 (2019)

[58] Pearson, K.: LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**(11), 559–572 (Nov 1901). https://doi.org/10.1080/14786440109462720, https://doi.org/10.1080/14786440109462720

[59] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[60] Ratner, B.: The correlation coefficient: Its values range between+ 1/- 1, or do they? Journal of Targeting, Measurement and Analysis for Marketing **17**(2), 139–142 (2009)

[61] Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 1135–1144 (2016)

[62] Rieck, B., Leitte, H.: Persistent homology for the evaluation of dimensionality reduction schemes. Computer Graphics Forum **34**(3), 431–440 (2015). https://doi.org/10.1111/cgf.12655, https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12655

[63] Ryley, P.: star-rating.js: A zero-dependency library that transforms a select with numerical-range values (i.e. 1-5) into a dynamic star rating element (Jan 2020), https://github.com/pryley/star-rating.js, last accessed on: 2016-10-06

[64] Sacha, D., Zhang, L., Sedlmair, M., Lee, J.A., Peltonen, J., Weiskopf, D., North, S.C., Keim, D.A.: Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis. IEEE Transactions on Visualization and Computer Graphics **23**(1), 241–250 (Jan 2017). https://doi.org/10.1109/TVCG.2016.2598495

[65] Sarveniazi, A.: An Actual Survey of Dimensionality Reduction. American Journal of Computational Mathematics **04**, 55–72 (Jan 2014). https://doi.org/10.4236/ajcm.2014.42006

[66] Sedlmair, M., Heinzl, C., Bruckner, S., Piringer, H., Möller, T.: Visual Parameter Space Analysis: A Conceptual Framework. IEEE Transactions on Visualization and Computer Graphics **20**(12), 2161–2170 (Dec 2014). https://doi.org/10.1109/TVCG.2014.2346321

[67] Sedlmair, M., Aupetit, M.: Data-driven evaluation of visual quality measures. In: Computer Graphics Forum. vol. 34, pp. 201–210. Wiley Online Library (2015)

[68] Shapley, L.S.: A value for n-person games. Contributions to the Theory of Games **2**(28), 307–317 (1953)

[69] Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: Proceedings 1996 IEEE Symposium on Visual Languages. pp. 336–343. IEEE (1996)

[70] Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 3145–3153. Journal of Machine Learning Research (2017)

[71] Silva, R.R.O.d., Rauber, P.E., Telea, A.C.: Beyond the Third Dimension: Visualizing High-Dimensional Data with Projections. Computing in Science Engineering **18**(5), 98–107 (Sep 2016). https://doi.org/10.1109/MCSE.2016.90

[72] Sips, M., Neubert, B., Lewis, J.P., Hanrahan, P.: Selecting good views of high-dimensional data using class consistency. In: Computer Graphics Forum. vol. 28, pp. 831–838. Wiley Online Library (2009)

[73] Székely, G.J., Rizzo, M.L., Bakirov, N.K.: Measuring and testing dependence by correlation of distances. Annals of Statistics **35**(6), 2769–2794 (2007)

[74] Tatu, A., Albuquerque, G., Eisemann, M., Schneidewind, J., Theisel, H., Magnork, M., Keim, D.: Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In: 2009 IEEE Symposium on Visual Analytics Science and Technology. pp. 59–66. IEEE (2009)

[75] Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science **290**(5500), 2319–2323 (Dec 2000). https://doi.org/10.1126/science.290.5500.2319, https://science.sciencemag.org/content/290/5500/2319

[76] Tibshirani, R.: Regression shrinkage and selection via the LASSO. Journal of the Royal Statistical Society: Series B (Methodological) **58**(1), 267–288 (1996)

[77] Torgerson, W.S.: Theory and methods of scaling. Theory and methods of scaling, Wiley, Oxford, England (1958)

[78] Ulyanov, D.: Multicore-TSNE: This is a multicore modification of Barnes-Hut t-SNE by L. Van der Maaten with python and Torch CFFI-based wrappers. https://github.com/DmitryUlyanov/Multicore-TSNE (2016), last accessed on: 2020-09-03

[79] Van Der Maaten, L.: Learning a Parametric Embedding by Preserving Local Structure. In: Artificial Intelligence and Statistics. pp. 384–391 (Apr 2009)

[80] Van Der Maaten, L.: Accelerating t-SNE using tree-based algorithms. The Journal of Machine Learning Research **15**(1), 3221–3245 (2014)

[81] Van Der Maaten, L., Hinton, G.: Visualizing Data using t-SNE. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008), http://jmlr.org/papers/v9/vandermaaten08a.html

[82] Van Der Maaten, L., Postma, E., Van den Herik, J.: Dimensionality reduction: A comparative review. Journal of Machine Learning Research **10**(66-71), 13 (2009)

[83] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., J. van der Walt, S., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., Van Mulbregt, P., contributors, S..: Scipy 1.0: fundamental algorithms for scientific computing in python. Nature methods **17**(3), 261–272 (2020)

[84] Wang, F., Sun, J.: Survey on distance metric learning and dimensionality reduction in data mining. Data Mining and Knowledge Discovery **29**(2), 534–564 (Mar 2015). https://doi.org/10.1007/s10618-014-0356-z, https://doi.org/10.1007/s10618-014-0356-z

[85] Wang, J., Liu, X., Shen, H.W., Lin, G.: Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. IEEE Transactions on Visualization and Computer Graphics **23**(1), 81–90 (2016)

[86] Wang, Y., Feng, K., Chu, X., Zhang, J., Fu, C.W., Sedlmair, M., Yu, X., Chen, B.: A perception-driven approach to supervised dimensionality reduction for visualization. IEEE Transactions on Visualization and Computer Graphics **24**(5), 1828–1840 (2017)

[87] Wattenberg, M., Viégas, F., Johnson, I.: How to Use t-SNE effectively. Distill **1**(10), e2 (Oct 2016). https://doi.org/10.23915/distill.00002, http://distill.pub/2016/misread-tsne

[88] Wilkinson, L., Anand, A., Grossman, R.: Graph-theoretic scagnostics. In: IEEE Symposium on Information Visualization, 2005. INFOVIS 2005. pp. 157–164. IEEE (2005)