LMU Munich
Frauenlobstraße 7a
D–80337 München
Institut für Informatik

Master Thesis

Intent Prediction with Vectorized Sequential UI Tree Data

August Oberhauser
august.oberhauser@campus.lmu.de

Course of Study:	Medieninformatik
Examiner:	Prof. Dr. Sven Mayer
Supervisor:	Florian Bemmann, M.Sc.
Commenced:	June 20, 2023
Completed:	January 8, 2024

Kurzfassung

<Short summary of the thesis>

Abstract

The interaction of a user with an end device such as a smartphone or a computer is very diverse and difficult to predict. Nevertheless, user-specific (personalized) as well as global (collaborative) patterns can possibly be worked out with the help of preceding user interactions. These could be used to predict the intention of a user or a group of users. It is interesting to know to what level of detail these predictions can be made reliably. By making use of the continous on-device data an attempt can be made to gain more insights in the user behavior or even forecast their next actions.

It suggests itself to implement this with the help of user interactions in sessions on Android devices. For this purpose, the Sequential UI Tree data of the device could be tracked, filtered and labeled and then trained with a machine learning model to find similar interaction sequences and then make predictions. These can then be very coarse, such as predicting the next app. Or they can be very detailed, e.g., determining the next user action, such as filling out a form field.

A concept will be developed on how a model for predicting user intent could be built and how it could be applied to the user session. To this end, possibilities for collecting and vectorizing sequential UI trees (e.g., from the Android Accessibility Service) will be discussed (e.g., via Recurrent Neural Network (*RNN*) [11] [1] [5], *Seq2Seq* Model [2], *Screen2Vec* Model [9], *Intention2Text* [14], *Html2Vec* [13]), which are designed to predict the user intent. Here, privacy and feature pre-filtering in UI data plays an important role. After that, personalized as well as collaborative data can be used in a hybrid approach. This model should then be made available to the user in an Android app service and, depending on the level of detail, suggest upcoming apps or actions to the user at a suitable time. It should also be considered whether the user can contribute to the learning process and improve suggested actions through feedback (labeling). The performance of the model can be measured, for example, by indicators such as the amount of training data and time spent on the learning process. The effectiveness can be evaluated by accuracy metrics in predicting, for example, app categories [10] or complete test sequences via Rico [3] or ERICA [4].

Furthermore, the machine learning model could provide the following benefits in addition to intent prediction:

- reduction of the complexity and size of the UI tree
- creation of user groups that have similar behavior when using digital UI systems [7]
- elimination of technical expertise on individual features that would be required to manually compare user sessions [6]
- consideration of a user's history over time (sequential)
- comparison of user interactions without providing privacy invasive information
- supporting app developers to improve their app design and usability
- application in psychology and market research
- pre-loading of processes on devices (energy savings) [12]

As listed above, many fields of application can profit by elaborating such a system. It would be exciting to know, how the concrete concept would look like and if it can be implemented successfully e.g. to improve the user experience on end-user devices.

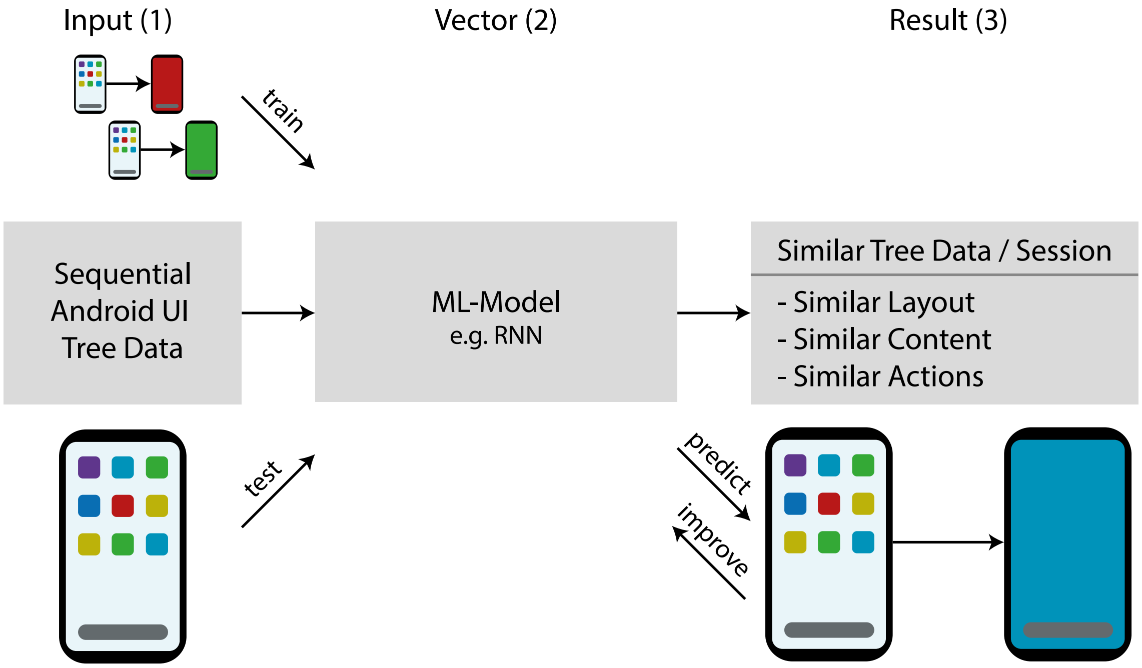


Figure 1: Possible procedure using a Machine-Learning algorithm to predict the next intent from a beginning user session: The input (1) can be a sequence of Android tree data. With help of a Machine-Learning-Model (2) (e.g. RNN) a vector representation can be trained and then predict the most probable action or screen (3) from a given starting sequence, but also can be improvde through the users feedback.

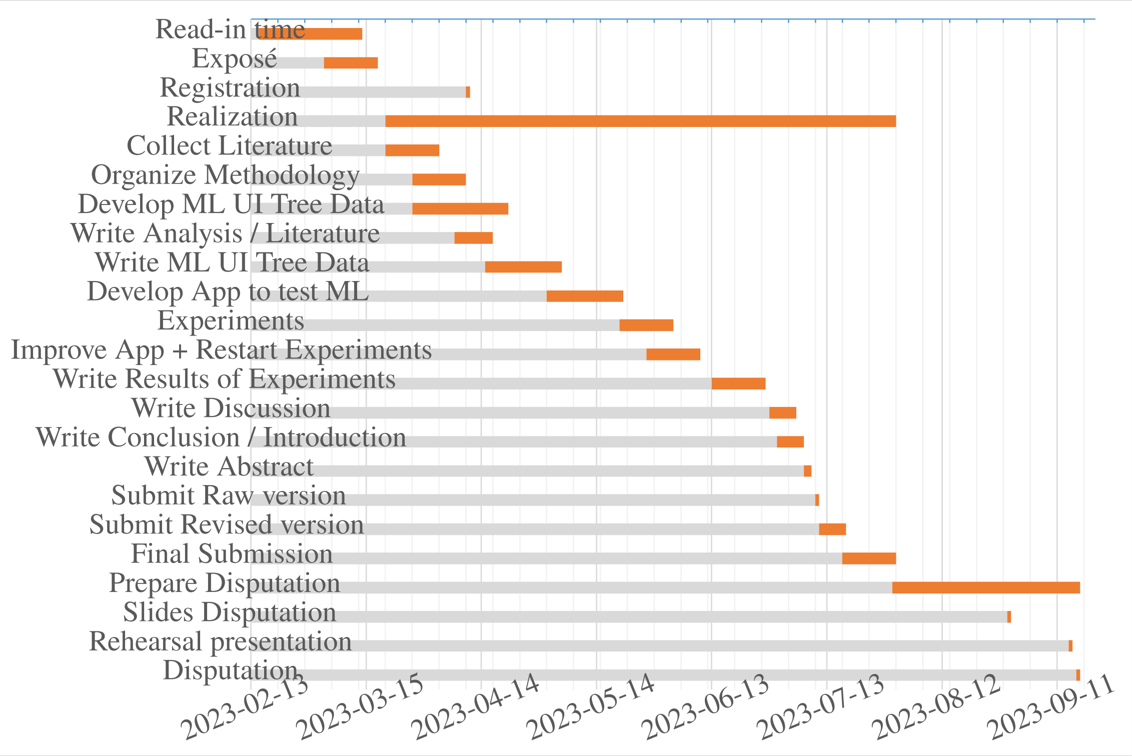


Figure 2: Schedule as a Gantt Chart

Contents

1 Introduction **13**

1.1 Necessarity of Vectors for Android UI 13

2 Related Work **15**

2.1 UI Tree and Datasets 15

2.1.1 ERICA 15

2.1.2 RICO / RicoSCA 15

2.1.3 Mobile UI CLAY Dataset 15

2.2 Vector models 17

2.2.1 Doc2Vec and Word2Vec 17

2.2.2 Screen2Vec 17

2.2.3 Screen2Words 17

2.2.4 Intention2Text 17

2.2.5 Html2Vec 17

2.2.6 Tree2Vec 17

2.2.7 Activity2Vec 17

2.3 Time Series / Sequence models 17

2.3.1 Seq2Seq Model 17

3 Methodology **19**

3.1 Android UI Data 19

3.1.1 Data tree structure 19

3.1.2 Retrieval of UI data via Android Accessibility Service 19

3.2 Machine Learning 19

3.2.1 Preprocessing 20

3.2.1.1 Feature selection 20

3.2.1.2 Normalization 20

3.2.1.3 Padding 20

3.2.1.4 Embedding 20

3.2.2 Supervised vs Unsupervised vs Semisupervised 21

3.2.2.1 Supervised Learning 21

3.2.2.2 Unsupervised Learning 21

3.2.3 Under and Overfitting 22

3.2.4 Evaluation Metrics 22

3.3 Artificial Neural Nets 22

3.3.1 Classes of Neural Nets 22

3.3.1.1 Deep Neural Nets 22

3.3.1.2 Convolutional Neural Nets 22

3.3.1.3 Recurrent Neural Networks and LSTMs / GRU 22

3.3.1.4 Autoencoders 22
3.3.2 Tensorflow and Keras 22
3.4 Evaluation and Metrics 23
3.4.1 Mean Squared Error 23
3.4.2 F1 Score 23

4 Results **25**
4.1 Datasets 25
4.1.1 Rico 25
4.2 Preprocessing Android UI tree data 25
4.2.1 Filtering privacy invasive details 25
4.2.2 Normalization, Feature selection 25
4.3 Model 25
4.4 Evaluation 26
4.4.1 Mean Squared Error 26
4.4.2 F1 Score 26
4.5 Limitations 26

5 Application of Android UI tree vectors **27**
5.1 Automation and testing of Android apps 27
5.2 UI design similarities 27
5.3 Action prediction models, User behavior modeling 27
5.4 Behavioral analyses for smartphone usage patterns 27

6 Conclusion and Future Work **29**

List of Figures

1 Possible procedure using a Machine-Learning algorithm to predict the next intent from a beginning user session: The input (1) can be a sequence of Android tree data. With help of a Machine-Learning-Model (2) (e.g. RNN) a vector representation can be trained and then predict the most probable action or screen (3) from a given starting sequence, but also can be improvde through the users feedback. 4

2 Schedule as a Gantt Chart 5

3.1 https://developer.android.com/studio/debug/layout-inspector, https://github.com/mimuc/app-ins-gruene 20

4.1 Model loss vs validation loss 26

List of Tables

2.1 Values of Rico Node 16

1 Introduction

This is a typical human-computer interaction thesis structure for an introduction which is structured in four paragraphs as follows:

1.1 Necessarity of Vectors for Android UI

Motivation for transforming Android UI tree data to vectors

[TODO]: P1.1. What is the large scope of the problem?

[TODO]: P1.2. What is the specific problem?

[TODO]: P2.1. The second paragraph should be about what have others been doing

[TODO]: P2.2. Why is the problem important? Why was this work carried out?

[TODO]: P3.1. What have you done?

[TODO]: P3.2. What is new about your work?

[TODO]: P4.1. What did you find out? What are the concrete results?

[TODO]: P4.2. What are the implications? What does this mean for the bigger picture?

2 Related Work

Describe relevant scientific literature related to your work.

2.1 UI Tree and Datasets

2.1.1 ERICA

See: [4]

2.1.2 RICO / RicoSCA

"Rico is a public UI corpus with 72K Android UI screens mined from 9.7K Android apps. [...] We manually removed screens whose view hierarchies do not match their screenshots by ask- ing annotators to visually verify whether the bound- ing boxes of view hierarchy leaves match each UI object on the corresponding screenshot image.This filtering results in 25K unique screens."

Use web interface to gain tree and interaction traces

See [3]

2.1.3 Mobile UI CLAY Dataset

Learning to Denoise Raw Mobile UI Layouts for Improving Datasets at Scale

- Provides a so-called *CLAY* pipeline which denoises mobile UI layouts from incorrect nodes or adding semantics to it.
- better than heuristic approach
- results are dynamic and out of sync, invisible objects, misaligned, in the background (greyed out)
- aim: “large scale high quality layout dataset”
- 37.4 % of the screens contain invalid objects

See [8] https://github.com/google-research/google-research/tree/master/clay

Key	**Type**	**Shape**	**Description**
		Per Trace	
activity_name	string	(1, None)	Name of the activity: “com.my_app.AppName.MainActivity”
is_keyboard_deployed	bool	(1)	Indicates if the keyboard is shown
request_id	int	(1)	[TODO]: TODO
		Per Node	
abs-pos	bool	(1)	Indicates if position in bounds is relative to the parent or absolute to the screen boundaries
adapter-view	bool	(1)	Indicates that children are loaded via an adapter, see https://developer.android.com/reference/android/widget/AdapterV
ancestors	bool	(1)	[TODO]: TODO
bounds	bool	(1)	[TODO]: TODO
children	[node]	(1)	[TODO]: TODO
class	bool	(1)	[TODO]: TODO
clickable	bool	(1)	User can interact by press / click
content-desc	bool	(1)	[TODO]: TODO
draw	bool	(1)	[TODO]: TODO
enabled	bool	(1)	[TODO]: TODO
focusable	bool	(1)	[TODO]: TODO
focused	bool	(1)	[TODO]: TODO
font-family	bool	(1)	[TODO]: TODO
long-clickable	bool	(1)	[TODO]: TODO

2.2 Vector models

2.2.1 Doc2Vec and Word2Vec

2.2.2 Screen2Vec

2.2.3 Screen2Words

2.2.4 Intention2Text

2.2.5 Html2Vec

2.2.6 Tree2Vec

2.2.7 Activity2Vec

2.3 Time Series / Sequence models

2.3.1 Seq2Seq Model

3 Methodology

Start with your overall approach to the research. What research problem or question did you investigate? What type of data did you need to answer it? Quantitative, qualitative, or mixed? Primary or secondary? Experimental or descriptive? Describe the specific methods you used for data collection and analysis. How did you collect and analyze your data? What tools or materials did you use? How did you ensure the quality and accuracy of your data? Explain why you chose these methods over others. How do they relate to your research question and literature review? How do they address the limitations or gaps in existing research? How do they suit your research design and objectives? Evaluate and justify your methodological choices. How did they affect the outcome of your research? What challenges or difficulties did you encounter and how did you overcome them? How can you ensure the credibility and generalizability of your findings?

How this thesis is working? Apparatus, Procedure, Utilities

3.1 Android UI Data

3.1.1 Data tree structure

3.1.2 Retrieval of UI data via Android Accessibility Service

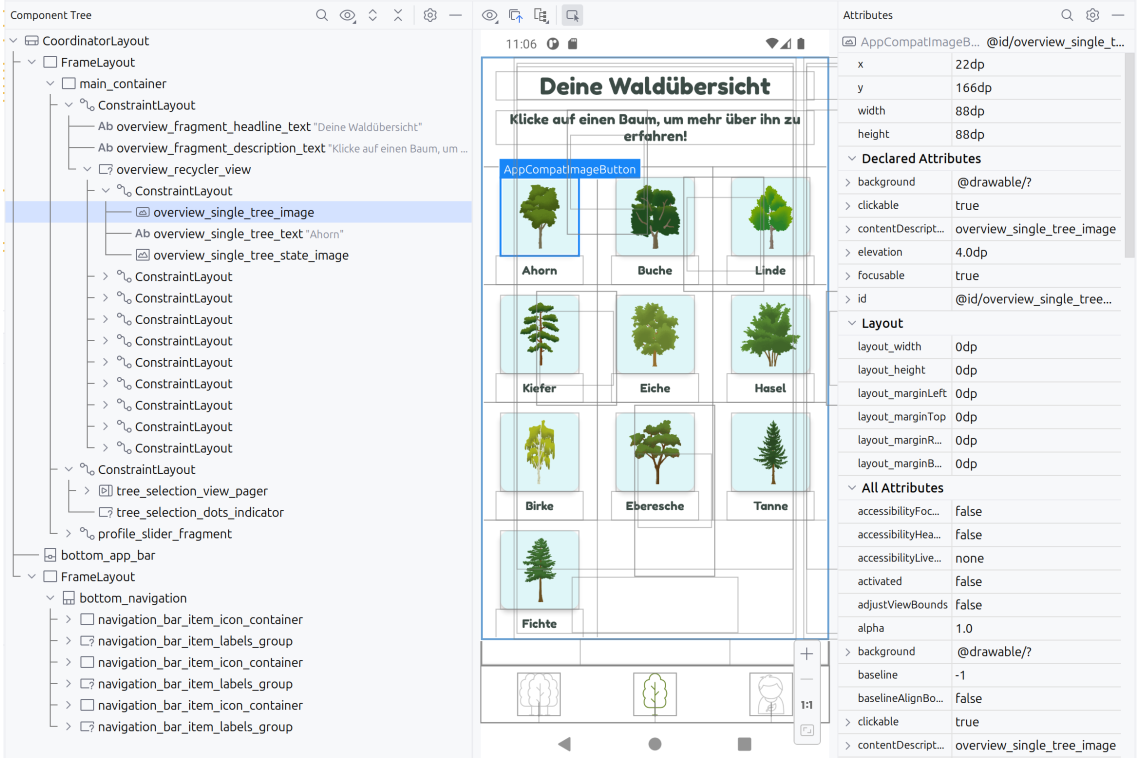
Semantics tree: https://developer.android.com/jetpack/compose/semantics https://android.googlesource.com/platform/fr
dev/uiautomator/library/src/com/android/uiautomator/core/AccessibilityNodeInfoDumper.java
https://github.com/Gustl22/android-accessibility/blob/c158808533d6fc017455184a7317555d3e6946f6/GlobalActionBa

3.2 Machine Learning

- Method of data analysis - Automation of model building, in contrast to statistical inference / models - Uses algorithms to iteratively learn - Find hidden relations / context / connections / patterns, without explicitly programming it - Generalization

Field of application: - Fraud detection - Network intrusion detection - Web search - Recommender Systems (Netflix, Amazon) - Advertising - Prediction (pricing,) - Text Analysis, LLMs - Pattern and image recognition - Email / Spam filtering

Drawbacks: - Biases, Discrimination



```
Component Tree
CoordinatorLayout
  FrameLayout
    main_container
      ConstraintLayout
        Ab overview_fragment_headline_text "Deine Waldübersicht"
        Ab overview_fragment_description_text "Klicke auf einen Baum, um ...
        overview_recycler_view
          ConstraintLayout
            overview_single_tree_image
            Ab overview_single_tree_text "Ahorn"
            overview_single_tree_state_image
          ConstraintLayout
          ConstraintLayout
          ConstraintLayout
          ConstraintLayout
          ConstraintLayout
          ConstraintLayout
          ConstraintLayout
          ConstraintLayout
          ConstraintLayout
      ConstraintLayout
        tree_selection_view_pager
        tree_selection_dots_indicator
      profile_slider_fragment
  bottom_app_bar
  FrameLayout
    bottom_navigation
      navigation_bar_item_icon_container
      navigation_bar_item_labels_group
      navigation_bar_item_icon_container
      navigation_bar_item_labels_group
      navigation_bar_item_icon_container
      navigation_bar_item_labels_group
```

Attributes

AppCompatImageB... @id/overview_single_t...

x	22dp
y	166dp
width	88dp
height	88dp
Declared Attributes	
background	@drawable/?
clickable	true
contentDescript...	overview_single_tree_image
elevation	4.0dp
focusable	true
id	@id/overview_single_tree...
Layout	
layout_width	0dp
layout_height	0dp
layout_marginLeft	0dp
layout_marginTop	0dp
layout_marginR...	0dp
layout_marginB...	0dp
All Attributes	
accessibilityFoc...	false
accessibilityHea...	false
accessibilityLive...	none
activated	false
adjustViewBounds	false
alpha	1.0
background	@drawable/?
baseline	-1
baselineAlignBo...	false
clickable	true
contentDescript...	overview_single_tree_image

Figure 3.1: https://developer.android.com/studio/debug/layout-inspector, https://github.com/mimuc/app-ins-gruene

3.2.1 Preprocessing

Tensors, Datasets

3.2.1.1 Feature selection

Such as Filtering privacy invasive details

Parameterizing the vectorization process a) Vector length b) Weighting of features c) Manipulating individual parameters of model

3.2.1.2 Normalization

3.2.1.3 Padding

3.2.1.4 Embedding

Category Embedding before LSTM

- Embedding layer Dimension near the actual average length of features (?)

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<hierarchy>
    <node index="0" text="" resource-id="" class="android.view.ViewGroup"
          package="de.lmu.treeapp" content-desc="" checkable="false"
          checked="false" clickable="false" enabled="true" focusable="false"
          focused="false" scrollable="false" long-clickable="false"
          password="false" selected="false" visible-to-user="true"
          bounds="[0,137][1440,2923]">
        <node index="0" text="" resource-id=""
              class="androidx.viewpager.widget.ViewPager" package="de.lmu.treeapp"
              content-desc="" checkable="false" checked="false" clickable="false"
              enabled="true" focusable="true" focused="false" scrollable="true"
              long-clickable="false" password="false" selected="false"
              visible-to-user="true" bounds="[4,141][1436,2707]">
            <node index="22" text="Eiche" resource-id=""
                  class="android.widget.TextView" package="de.lmu.treeapp"
                  content-desc="" checkable="false" checked="false"
                  clickable="false" enabled="true" focusable="false"
                  focused="false" scrollable="false" long-clickable="false"
                  password="false" selected="false" visible-to-user="true"
                  bounds="[116,182][1324,315]" />
        </node>
        <node index="1" text="" resource-id="" class="android.view.ViewGroup" ...>
            <node NAF="true" index="0" text="" resource-id=""
                  class="android.widget.FrameLayout" package="de.lmu.treeapp" ... />
        </node>
    </node>
</hierarchy>
```

Listing 3.1: Android Accessibility Node in XML.

3.2.2 Supervised vs Unsupervised vs Semisupervised

3.2.2.1 Supervised Learning

Supervised: Classification and regression Uses **labeled** examples: Input and output is known

Learns by comparison of the output it is provided with the output the model *predicts*.

Steps: - Data acquisition - Data cleaning / Preprocessing (Panadas) - Split into Training Data, Validation data, and Test data (cannot adapt the model after using the test data) - Train the model with the train data - Evaluate the model with the test data, then can adapt the model by the developer - Last deploy the model to production

3.2.2.2 Unsupervised Learning

Reinforcement learning

3.2.3 Under and Overfitting

3.2.4 Evaluation Metrics

3.3 Artificial Neural Nets

- Uses biological neuron systems as paradigm to generate mathematical models - can solve tasks by abstraction or generalization of data relations

Activation Functions Cost function Gradient - Regression: Continous Values - Classification: Multiple class - One Class

3.3.1 Classes of Neural Nets

3.3.1.1 Deep Neural Nets

Neural Net with more than one layer - Dense Layer

3.3.1.2 Convolutional Neural Nets

3.3.1.3 Recurrent Neural Networks and LSTMs / GRU

LSTM 4 dimensional

Limitations to only 3 dimensions, needs flattening

Sample dimension (X -> y) Time (Step) Dimension Feature Dimension Data, Quantity dimension, such as Image dimensions, or multiple nodes

TimeDistributedLayer

3.3.1.4 Autoencoders

Encoder, Decoder

3.3.2 Tensorflow and Keras

Layers FlattenLayer

Positive Integer to Dense Vectors of fixed size

3.4 Evaluation and Metrics

3.4.1 Mean Squared Error

3.4.2 F1 Score

4 Results

4.1 Datasets

- Problem with sequential data sets

4.1.1 Rico

- Too less frames.
- No transition between apps.

4.2 Preprocessing Android UI tree data

4.2.1 Filtering privacy invasive details

4.2.2 Normalization, Feature selection

Dealing with variable length data tf.io.VarLenFeature()

4.3 Model

Multiple approaches

AutoEncoder:

- Encoder -> Decoder -> LSTM -> Decoder
- Encoder -> LSTM -> Decoder
- LSTM -> Encoder -> Decoder (AutoEncoder)

Decoder can either only decode to x and y or to whole UI tree.

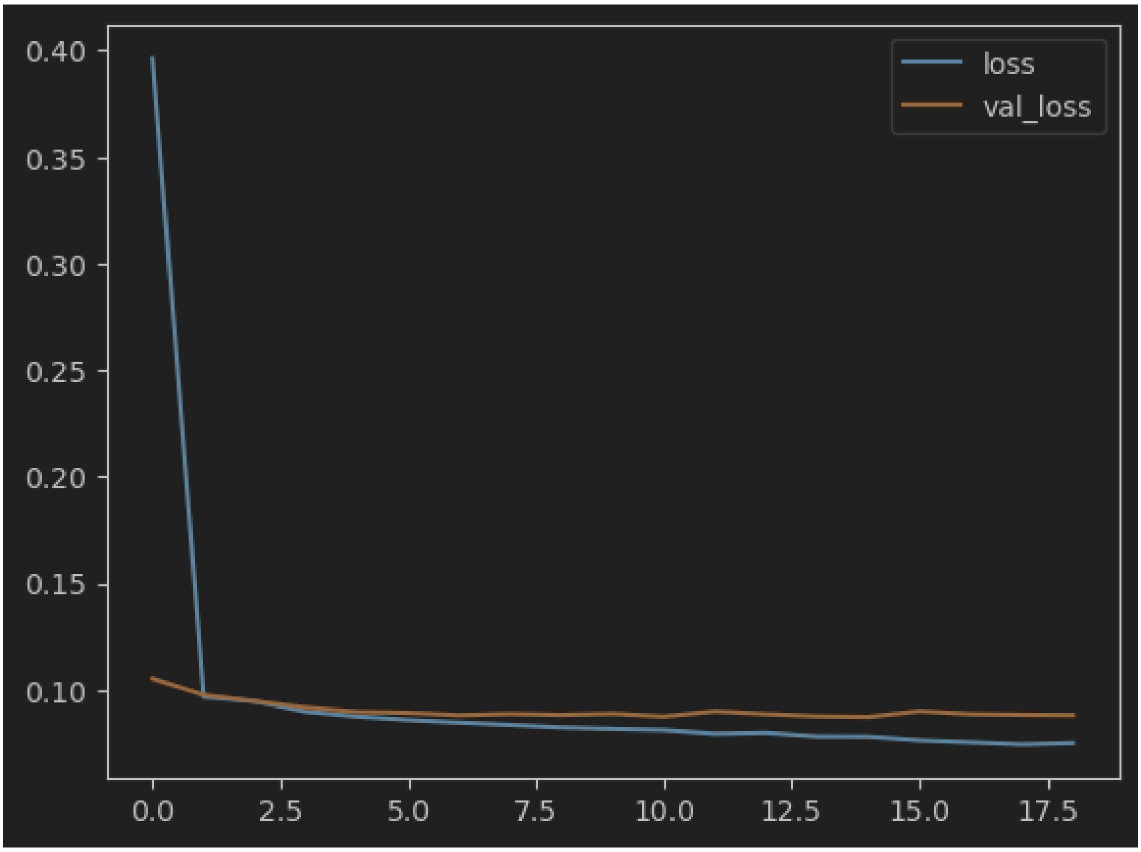


Figure 4.1: Model loss vs validation loss

4.4 Evaluation

4.4.1 Mean Squared Error

4.4.2 F1 Score

4.5 Limitations

Dataset Dataset is not through different apps, only in one app. Dataset is not detailed enough in the time steps, or not containing all data Dataset is not long enough Dataset has no paid apps or apps with login, which most services require Dataset has wrong data see [8]

Preprocessing Need more time to validate what are the core parameters to predict the next user intent

Model needs more investigation on what data is needed How many neurons are required to achieve this Play around with different layers, also Convolutional and pretrained embeddings

5 Application of Android UI tree vectors

5.1 Automation and testing of Android apps

5.2 UI design similarities

5.3 Action prediction models, User behavior modeling

5.4 Behavioral analyses for smartphone usage patterns

6 Conclusion and Future Work

Summary

Outlook

Future directions for research in this area

ChatGPT – Image Recognition – Limitation als Ausblick

Generate Dataset which overcomes the limitations

Make a study with actual feedback on a prediction system, visualization

References

[1] Saumya Bansal, Niyati Baliyan. "Remembering past and predicting future: a hybrid recurrent neural network based recommender system". In: *Journal of Ambient Intelligence and Humanized Computing* (2022), pp. 1–12.

[2] François Chollet. *A ten-minute introduction to sequence-to-sequence learning in Keras*. 2017. URL: `https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html` (visited on 02/15/2023).

[3] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, Ranjitha Kumar. "Rico: A mobile app dataset for building data-driven design applications". In: *Proceedings of the 30th annual ACM symposium on user interface software and technology*. 2017, pp. 845–854.

[4] Biplab Deka, Zifeng Huang, Ranjitha Kumar. "ERICA: Interaction mining mobile apps". In: *Proceedings of the 29th annual symposium on user interface software and technology*. 2016, pp. 767–776.

[5] Mauro Di Pietro. *Modern Recommendation Systems with Neural Networks*. 2022. URL: `https://towardsdatascience.com/modern-recommendation-systems-with-neural-networks-3cc06a6ded2c` (visited on 03/08/2023).

[6] Alireza Ghods, Diane J Cook. "Activity2vec: Learning adl embeddings from sensor data with a sequence-to-sequence model". In: *arXiv preprint arXiv:1907.05597* (2019).

[7] Kasthuri Jayarajah, Youngki Lee, Archan Misra, Rajesh Krishna Balan. "Need accurate user behaviour? pay attention to groups!" In: *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. 2015, pp. 855–866.

[8] Gang Li, Gilles Baechler, Manuel Tragut, Yang Li. "Learning to Denoise Raw Mobile UI Layouts for Improving Datasets at Scale". In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New Orleans, LA, USA: Association for Computing Machinery, May 2022, pp. 1–13. URL: `https://doi.org/10.1145/3491102.3502042`.

[9] Toby Jia-Jun Li, Lindsay Popowski, Tom Mitchell, Brad A Myers. "Screen2vec: Semantic embedding of gui screens and gui components". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–15.

[10] Google LLC. *Choose a category and tags for your app or game.* 2023. URL: `https://support.google.com/googleplay/android-developer/answer/9859673` (visited on 02/15/2023).

[11] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, Paolo Cremonesi. "Personalizing session-based recommendations with hierarchical recurrent neural networks". In: *proceedings of the Eleventh ACM Conference on Recommender Systems*. 2017, pp. 130–137.

[12] Zhihao Shen, Kang Yang, Wan Du, Xi Zhao, Jianhua Zou. "Deepapp: a deep reinforcement learning framework for mobile application usage prediction". In: *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 2019, pp. 153–165.

[13] Peihuang Wu, Jiakun Zhao. "Distributed representations of html page". In: *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*. IEEE. 2022, pp. 160–164.

[14] Che-Hsuan Yu, Hung-Yuan Chen, Fang-Yie Leu, Yao-Chung Fan. “Understanding mobile user intent using attentive sequence-to-sequence RNNs”. In: *Wireless Internet: 12th EAI International Conference, WiCON 2019, TaiChung, Taiwan, November 26–27, 2019, Proceedings 12*. Springer. 2020, pp. 51–61.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature