Enrico: A Dataset for Topic Modeling of Mobile UI Designs

Luis A. Leiva
Aalto University
Finland
firstname.lastname@aalto.fi

Asutosh Hota
Aalto University
Finland
firstname.lastname@aalto.fi

Antti Oulasvirta
Aalto University
Finland
firstname.lastname@aalto.fi

ABSTRACT

Topic modeling of user interfaces (UIs), also known as layout design categorization, contributes to a better understanding of the UI functionality. Starting from Rico, a large dataset of mobile UIs, we revised a random sample of 10k UIs and concluded to Enrico (shorthand of Enhanced Rico), a human-supervised high-quality dataset comprising 1460 UIs and 20 design topics. As a validation example, we train a deep learning model for three different UI representations (screenshots, wireframes, and embeddings). The screenshot representation provides the highest discriminative power (95% AUC) and a competitive accuracy of 75% (a random classifier achieves 5% accuracy in the same task). We discuss several applications that can be developed with this new public resource, including e.g. semantic UI captioning and tagging, explainable UI designs, smart tutorials, and improved design search capabilities.

CCS CONCEPTS

• Information systems → Database design and models; **• Human-centered computing →** *Interaction design process and methods.*

KEYWORDS

User interface design; Layout classification; Machine learning; Neural networks

ACM Reference Format:
Luis A. Leiva, Asutosh Hota, and Antti Oulasvirta. 2020. Enrico: A Dataset for Topic Modeling of Mobile UI Designs. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '20 Extended Abstracts), October 5–8, 2020, Oldenburg, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3406324.3410710

1 INTRODUCTION AND RELATED WORK

Topic modeling of user interfaces (UIs), also known as layout design categorization, contributes to a better understanding of the UI functionality. Advancements in design mining have contributed to creating more usable and engaging UIs, however more research is needed to understand how UIs are designed and how content is presented to the users. Of particular interest is the mobile app ecosystem: Current marketplaces such as Google's Play Store or Apple's App Store provide a high-level categorization for each app together with some metadata like app description and sample images. Unfortunately, this categorization is insufficient to derive meaningful insights about UI design, since every mobile app usually implements several UI layouts; e.g. a news app may show an introductory tutorial to first-time users or display messages in a modal window. Furthermore, different apps from different marketplace categories (say, Business and Fitness) may implement the same layout; e.g. a login screen that requests the user to enter a username and password. In this context, designers could gain valuable knowledge about UI layout structure and how it is used in different apps if they had access to this kind of data. Ultimately, topic modeling of UIs enables novel applications such as the ones we discuss in the last section of this paper.

So far, there is no public dataset that has associated UI designs with specific layout topics. Previous design mining applications and platforms such as Webzeitgeist [5] and GUIfetch [1] allowed for exploration and querying of web design datasets, but no UI layout modeling is possible because of the lack of labeled data. Same happens with app discovery services such as Appazaar, Xyo, or Appcrawlr. And while other datasets provide fine-grained app categories, they do not inform about UI design. For example, Berardi et al. [2] automatically inferred app categories using app metadata analysis. Zhu et al. [13] annotated 680 mobile apps according to two general-purpose levels, e.g. Games (Level 1) includes Action and Strategy (Level 2). There is even a mobile app dataset categorized according to Reiss' profiles [9]; e.g. honor, idealism, power, etc. that unfortunately have little practical use for UI designers. Rico [3], the largest dataset of mobile UIs, provides valuable design data but also lacks such layout design topics. To bridge this gap, we contribute Enrico (shorthand of Enhanced Rico), a curated dataset of mobile UIs drawn from Rico. We have manually revised a random sample of 10k UIs and concluded to a high-quality dataset comprising 1460 UIs, which is large enough for most data-driven design tasks, including training of deep learning models, as shown in this paper.

In this paper, we describe Enrico and, as a validation example, train a topic classifier that achieves a highly competitive Top-1 accuracy over 75% (a random classifier achieves 5% accuracy) and an AUC of 95% when trained on UI screenshots. We also discuss several applications that can be developed with this new public resource, including e.g. semantic UI captioning and tagging, explainable UI designs, smart tutorials, and improved design search capabilities.

2 RICO

The Rico dataset is probably the largest public repository of mobile app designs to date. It contains 72k UI screenshots with annotations about the UI elements (icon, button, text, navigation, etc.) in both textual (view hierarchies) and visual form (semantic wireframes).

However, since it was compiled with automated crawling and in-the-wild app usage, Rico is very noisy. We have noticed that most of the semantic wireframes do not represent their UI correctly. While this may have a negligible impact for some tasks, e.g. retrieval of

MobileHCI '20 Extended Abstracts, October 5–8, 2020, Oldenburg, Germany

ACM ISBN 978-1-4503-8052-2/20/10.
https://doi.org/10.1145/3406324.3410710

Table 1: Enrico topics, in alphabetical order, and the number of UI designs that belong to each topic.

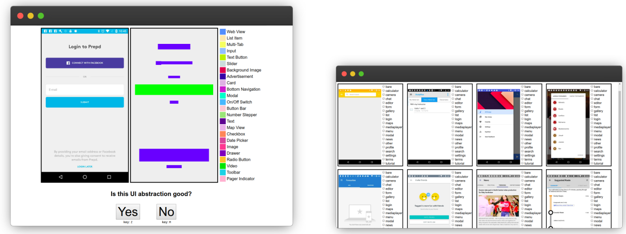
Topic	No.	Description
Bare	76	Largely unused area
Dialer	6	Number entry
Camera	8	Camera functionality
Chat	11	Chat functionality
Editor	18	Text/Image editing
Form	103	Form filling functionality
Gallery	144	Grid-like layout with images
List	265	Elements organized in a column
Login	141	Input fields for logging
Maps	9	Geographic display
Media Player	32	Music or video player
Menu	79	Items list in an overlay or aside
Modal	67	A popup-like window
News	59	Snippets list: image, title, text
Other	52	Everything else (rejection class)
Profile	63	Info on a user profile or product
Search	35	Search engine functionality
Settings	90	Controls to change app settings
Terms	39	Terms and conditions of service
Tutorial	163	Onboarding screen
Total	1460	

similar user interfaces [7], we estimate that about 10% of the UIs can only be deemed as high-quality design examples. Lee et al. [6] reported a similar rate in a recent study. The most common issues that we have identified include the following: (1) Mismatch between app screenshot and wireframe; (2) Mismatch between view hierarchy and wireframe; (3) No wireframe available or empty image; (4) Considerable overlaps among wireframe elements. These issues account for approximately 90% of the data we have scrutinized. Therefore, a manual verification pass is needed in order to produce a smaller dataset but of much higher quality.

3 ENRICO

We began by creating a web-based revision interface that displayed a Rico instance (a pair of screenshot and semantic wireframe) together with the color code of each UI component; see Figure 1(a). Each instance was randomly sampled from the large pool of 72k mobile UIs and was ranked as a good or bad design example by two human annotators. The goal of this revision procedure was to filter out poor design examples, as discussed previously. The annotators were given clear instructions, aimed at encouraging a systematic verification procedure: given a Rico instance, indicate if the semantic wireframe matches the elements shown in the corresponding UI screenshot. To ensure a consistent criteria, assessment conflicts were discussed (11 cases) and reached consensus.

Random sampling is, by definition, an unbiased selection method, so on average any sufficiently large random sample accurately represents the whole population. Eventually we revised 10k UIs and identified 20 different UI design topics (Table 1) by manually inspecting the collected designs. To our knowledge, the resulting UI taxonomy is rather comprehensive, with few rarer layout design types categorized (the 'Other' topic represents a merely 3% of the data).



(a) Revision interface. (b) Annotation interface.

Figure 1: Snapshots of our *revision* (a) and *annotation* (b) web-based interfaces. First, Rico designs were assessed as either good or bad examples (a). Then, the good designs were assigned a layout topic (b).

Finally, we created a web-based annotation interface that loaded the screenshots from the pool of *good* designs and let us assign the most plausible topic label to them; see Figure 1(b). Each screenshot was also labeled by two human annotators. To ensure a consistent criteria, label mismatches were discussed (3 cases) and reached consensus. Table 1 indicates the list of design topics and the number of UIs belonging to each topic. The "Other" topic label is meant to group designs that do not clearly belong to any of the identified UI topics, and can be considered a "rejection" or out-of-distribution class in machine learning parlance [4].

4 EXPERIMENTS

To exemplify the kind of tasks that can be conducted with Enrico, we train a topic classifier for each of the following UI representations: Screenshot, Wireframe, and Embedding. Screenshot and wireframes are encoded as RGB images of 256×128 px resolution, to speed up training. Embeddings are computed with a deep convolutional autoencoder that takes a UI wireframe as input (Figure 2). The encoder part creates a 32×16×32 bottleneck layer, which outputs a latent vector (embedding) of the input wireframe. The decoder part is a "mirror" of the encoder where the pooling operations are replaced by upsampling operations. The decoder is only needed to ensure that the resulting UI embeddings are able to capture the layout structure successfully (Figure 3). The autoencoder is trained with the Adam optimizer (learning rate $\eta = 0.001$ and decay rates $\beta_1 = 0.9, \beta_2 = 0.999$) using mean squared error as loss function, since its goal is to reconstruct the input wireframe.

All our topic classifiers are essentially the same convolutional neural net, inspired by the VGG16 architecture [12], which comprises 5 convolutional blocks with max pooling and 0.2 dropout rate. Then, depending on the input representation, the architecture is slightly modified. For screenshot input, a batch normalization layer is added after each convolutional block. For wireframe input, an extra fully connected layer is added before the output layer. For embedding input, the bottleneck layer of the autoencoder is connected to a softmax layer. These three models are trained with the

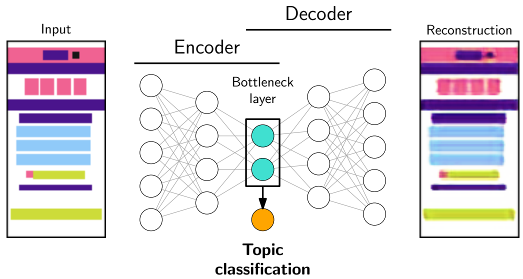


Figure 2: We trained a deep convolutional autoencoder to create UI embeddings. The encoder part is re-engineered to classify any embedding into one of our 20 topics (Table 1).

Adam optimizer ($\eta = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$) using categorical cross-entropy as loss function, since their goal is to classify the input data into one of the 20 UI topics (Table 1).

We randomly split the 1460 UIs into three data partitions: 80% of the data is used for training (with 15% of the training data used as model validation) and the remaining 20% is used for testing. Topic class weights were computed to ensure balanced splits of the data partitions, given that some design topics are more frequent than others; see Table 1.

Table 2: Classification performance results for different UI design representations. All metrics are reported in percentage.

	Top-k accuracy						
Input	$k = 1$	$k = 3$	$k = 5$	**Prec.**	**Recall**	**F1**	**AUC**
Screenshot	75.8	88.4	92.9	76.6	75.8	75.4	95.1
Wireframe	39.4	66.1	78.7	50.1	39.4	40.2	85.3
Embedding	50.9	73.6	84.3	60.2	50.9	51.1	89.5

Table 2 shows the classification performance results. As illustrated by the top-k accuracy columns, our models are able to identify the right topic most of the time. As a reference, a random classifier would be $1/20 = 5\%$ accurate. We also report the usual retrieval-based metrics (Precision, Recall, and F-measure) to illustrate model performance further.

We can see that using UI screenshots (top row) as input achieves the best results. This is further corroborated by the Area Under the Curve (AUC) score, which measures the discriminative power of any classifier [10]. Topic classification via UI embeddings has potential, as evidenced in Figure 3 (bottom row) and further validated in Figure 4, where it can be observed that UI topics are reasonably well separated. Each dot in the figure represents a 2D projection of the UI embeddings using the UMAP algorithm [8]. Excepting the 'list' topic, which is the largest one and the more diverse in Enrico, all topics are clustered around concrete areas of the plot. Finally, we argue that the weaker performance of the semantic wireframes (middle row) is probably due to the chosen color scheme to represent each UI element, which was inherited from the Rico dataset. For example, 'Image' and 'Date Picker' have a very similar (red-based) color, so both UI elements may become indistinguishable to any classifier that considers color as a discriminative feature.

5 DISCUSSION

This paper represents our ongoing efforts toward creating next-generation UI design tools informed by topic modeling tasks. In a nutshell, designers can use Enrico to create novel applications that rely on a high-level understanding of a given UI layout.

5.1 Applications

In the following we discuss some examples of the developments that are possible with Enrico and our deep learning models.

Semantic UI captioning: Create short descriptions of the UI, conditioned to the most probable topic, so that the description is more understandable. These short descriptions can also be used e.g. as ALT text for screen readers, improving thus accessibility and search engine optimization.

Automatic UI tagging: Designers can query our screenshot classifier and request e.g. the top-3 most likely topics of a given UI layout and display that information as inline tags to the users.

Annotation interfaces: Our screenshot classifier can speed up the annotation process for new UIs (Figure 1(b)) by arranging the topics list from higher to lower probability (n-best list) so that the annotator can quickly select the most suitable topic.

Explainable UI designs: Developers can arrange the softmax vector of any of our models from high to low probability and display a histogram-based visualization to inform "verbally" about the structure of a given UI, inspired by prior work on text classification [11].

Tag cloud visualization: Designers can communicate graphically the n-best list of topics with a tag cloud, where each tag is rendered with a font size proportional to the topic probability, so larger tags indicate higher importance.

Smart tutorials: The concept of "UI tours" is very popular in web design, as a means to provide contextual help. By knowing the topic of a given UI layout, designers can provide more adequate tooltips of what the users can do on the UI.

Improved search and retrieval: Researchers can enhance query-by-image retrieval models [3, 7] by incorporating explicit semantic labels about a UI layout design. These labels should help disambiguate and improve search.

5.2 Limitations

We should point out that our revision procedure, where the original Rico UIs were deemed as either good or bad, might have been impacted by the cultural background of our human annotators. However, the annotation instructions were clear and encouraged a systematic procedure. Further, the few mismatches among annotators (11 cases out of 1460) were discussed until reaching consensus.

The experiments we have conducted in this paper are just an example of the computational modeling tasks that are possible with Enrico. We are confident that better models can be developed if more architectures and input representations are considered. For example, recurrent neural nets and transformers would allow for a

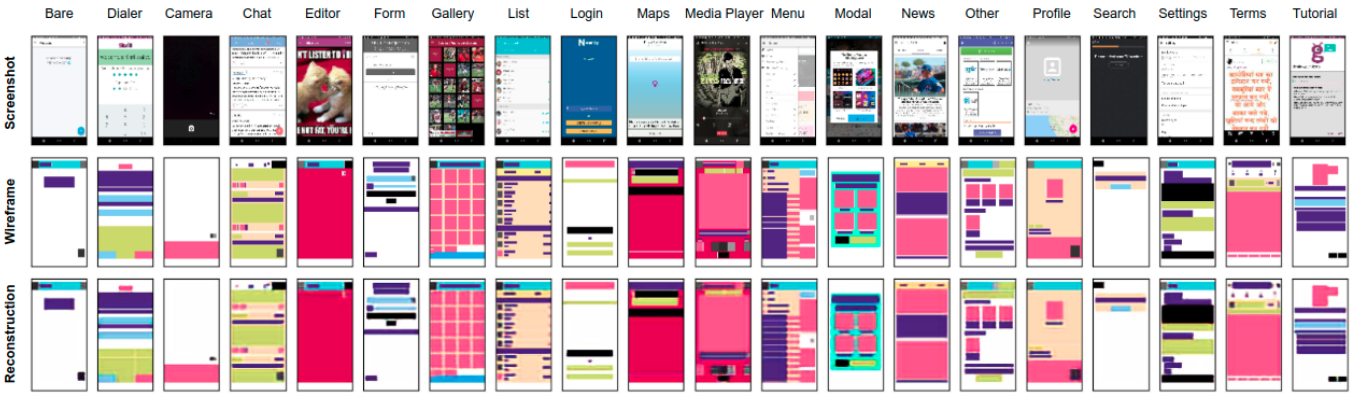


Figure 3: Examples of Enrico UIs. The layout reconstructions (bottom row) were computed with our deep convolutional autoencoder.

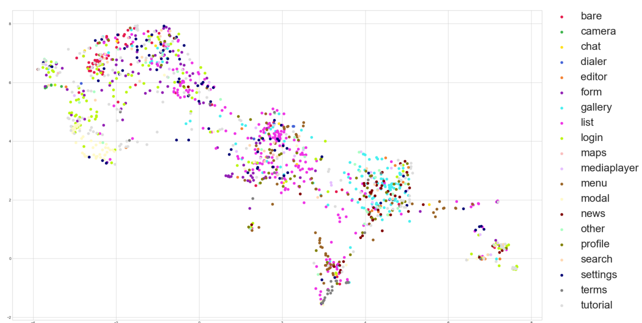


Figure 4: Enrico's latent space: 2D manifold projection (UMAP algorithm) of the UI embeddings.

non-pixel based representation of the UI, which has potential for improving topic classification performance.

Finally, there are several 2D projection algorithms that could be used to visualize the latent space of Enrico (Figure 4). Each of these algorithms produce different projections, resulting in different clusterings. Further, modern projection algorithms are stochastic, so different runs with the same hyperparameters may yield different results. Therefore, it remains unknown which algorithm would produce the best visualizations of Enrico for UI designers.

6 CONCLUSION AND FUTURE WORK

Enrico is a new reference dataset for mobile UI design that researchers, designers, and developers can use to build a wide array of applications that require understanding of design layout functionality. We manually revised 10k UIs to ensure that Enrico would be a reasonably large dataset, though we plan to revise more UIs in the future. We also plan to combine our three analyzed input formats (see Experiments) in a consolidated model and include high-level features from apps metadata (also available in Enrico).

ACKNOWLEDGMENTS

We acknowledge the computational resources provided by the Aalto Science-IT project. We thanks Crista Kaukinen for helping us with the UI labeling tasks. This work has been supported by the Academy of Finland (grant no. 318559). Enrico is available at https://github.com/luileito/enrico.

REFERENCES

[1] F. Behrang, S. P. Reiss, and A. Orso. 2018. GUIfetch: Supporting App Design and Development Through GUI Search. In *Proc. MOBILESoft*.
[2] G. Berardi, A. Esuli, T. Fagni, and F. Sebastiani. 2015. Multi-Store Metadata-Based Supervised Mobile App Classification. In *Proc. SAC*.
[3] B. Deka, Z. Huang, C. Franzen, J. Hibschman, D. Afergan, Y. Li, J. Nichols, and R. Kumar. 2017. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In *Proc. UIST*.
[4] D. Hendrycks and K. Gimpel. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *Proc. ICLR*.
[5] R. Kumar, A. Satyanarayan, C. Torres, M. Lim, S. Ahmad, S. R. Klemmer, and J. O. Talton. 2013. Webzeitgeist: Design Mining the Web. In *Proc. CHI*.
[6] C. Lee, S. Kim, D. Han, H. Yang, Y.-W. Park, B. C. Kwon, and S. Ko. 2020. GUIComp: A GUI Design Assistant with Real-Time, Multi-Faceted Feedback. In *Proc. CHI*.
[7] T. F. Liu, M. Craft, J. Situ, E. Yumer, R. Mech, and R. Kumar. 2018. Learning Design Semantics for Mobile Apps. In *Proc. UIST*.
[8] L. McInnes, J. Healy, and J. Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. ArXiv:1802.03426. (2018).
[9] E. Platzer and O. Petrovic. 2011. Learning Mobile App Design From User Review Analysis. *iJIM* 5, 3 (2011).
[10] D. Powers. 2011. Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. *J. Mach. Learn. Technol.* 2, 1 (2011).
[11] M. T. Ribeiro, S. Singh, and C. Guestrin. 2016. Why should I trust you? Explaining the predictions of any classifier. In *Proc. KDD*.
[12] K. Simonyan and A. Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. ICLR*.
[13] H. Zhu, E. Chen, H. Xiong, H. Cao, and J. Tian. 2014. Mobile App Classification with Enriched Contextual Information. *IEEE Trans. Mob. Comput.* 13, 7 (2014).