Typeface Semantic Attribute Prediction from Rasterized and Vectorized Font Representations

Suvir Mirchandani Lucia Zheng Julia Gong
{smirchan, zlucia, jxgong}@stanford.edu

1. Introduction and Motivation

Typefaces comprise an important component of the aesthetic quality of any written work. With the rise of printers and digital word processors has come a tremendous diversity of typefaces, each slightly different in visual effect and design. With this growing number of choices, there has been an increasing need to intelligently categorize and characterize typefaces for searching, grouping, pairing, and more. Most approaches thus far have categorized typefaces by their font styles—for example, 'bold' or 'italic'—as well as other *typographic attributes*, such as 'all-capitals' or 'cursive'. However, while these characteristics can be descriptive for some fonts, these aspects fail to capture some of the more aesthetic qualities—what we call *semantic attributes*—that we as humans associate with typefaces in abstract senses, such as 'artistic', 'boring', or 'gentle', which may be desirable when one is looking for typefaces that have particular semantic attributes. We are thus interested in applying machine learning to build on existing work for generalizing the representation and characterization of typeface semantic attributes beyond existing semantic attribute datasets.

In particular, our aim is to improve on the work of Kulahcioglu & de Melo (2018) by building a model that can accurately predict an associated semantic attribute vector for a given typeface based on existing human-labeled data (O'Donovan et al., 2014) and font embeddings. In our case, the semantic attribute vector is a 31-dimensional vector where each entry is a value between 0 and 1 that represents how strongly the given semantic attribute describes the typeface. To this end, our task is to extend an open-source dataset called FontJoy[†]—which contains font names and corresponding rasterized images, of which there are 1883 (1861 unique)—by inducing a dataset of the corresponding font semantic attributes for each FontJoy font using 161 semantically labeled fonts from the dataset by O'Donovan et al. (2014). We only use the 161 of the 200 fonts in this dataset (e.g. 'Source Sans Pro Semibold') that overlap between both datasets. We also introduce a novel downstream task of *semantic attribute regression* to measure how effectively models can predict the 31 semantic attributes given rasterized images of a typeface, along with our experimental findings for this task.

The input to our dataset induction algorithm is the dataset of 161 human-labeled 31-dimensional real-valued semantic attribute vectors with entries between 0 and 1 for each typeface in the semantic attribute dataset of O'Donovan et al. (2014), along with the FontJoy embeddings for the 1861 unique typefaces to be used in dataset induction. The outputs are the corresponding 31-dimensional semantic attribute vectors for all 1861 FontJoy typefaces.

The input to our downstream semantic attribute prediction task is the full induced dataset of size 1861 from our dataset induction task. Our train-validation-test split is 70-10-20, respectively. Given rasterized images of aligned representative glyphs from the training fonts in FontJoy—see Figure 1—along with their ground-truth (the induced dataset semantic attribute vectors from above), our model aims to learn the parameters used to regress the 31-dimensional semantic attribute vector for a given font in inference.

2. Related Work

Prior work for analysis and generation of typefaces has been largely focused on typographic attributes and less on semantic attributes. These works use unsupervised methods to cluster images of typefaces into groups with similar appearances in contexts such as OCR (Öztürk et al., 2000; Avilés-Cruz et al., 2004) and in isolated typeface data (Azadi et al., 2017; Zhang et al., 2018; Lin et al., 2019; Lopes et al., 2019). While strong for capturing typographic font attributes, these works unfortunately do not handle the level of abstraction of semantic attributes, which may be more useful for human purposes. The primary works that have focused on semantic attributes mostly focus on crowd-sourcing the semantic datasets themselves (O'Donovan et al., 2014) or inducing a larger dataset from human labels using simple k-nearest neighbor (k-NN) methods (Kulahcioglu & de Melo, 2018). In this paper, we aim to improve on this work for dataset induction using non-parametric supervised learning methods to create more large-scale, accurate semantic attribute datasets. In addition, prior work has not discussed semantic

[†]See FontJoy, https://github.com/Jack000/fontjoy

Figure 1. Examples of input images of typefaces in the dataset that vary in their semantic attributes. All images are grayscale since they are antialiased black-and-white images.

attribute prediction from visual font representations, a novel downstream task that our work introduces.

Finally, state-of-the-art systems have utilized deep networks to learn latent spaces for typefaces to aid in typeface generation, interpolation, and font style transfer (Azadi et al., 2017; Zhang et al., 2018; Lopes et al., 2019; Lin et al., 2019). While these systems are well-suited to create powerful latent spaces for typefaces, they are not as interpretable due to their end-to-end nature. We posit that semantic attribute vectors are a more interpretable and thus useful set of attributes to aid typeface searching, selection, and pairing.

3. Method

3.1. Dataset Overview

As mentioned prior, our dataset consists of rasterized images, FontJoy embeddings, and human-labeled semantic attributes for 161 fonts. Examples of rasterized images are shown in Figure 1. The FontJoy embeddings were obtained using a deep Convolutional Neural Network (CNN) font embedding space model, reduced to 200-dimensional vectors via PCA. The task is to use the FontJoy embeddings and semantic attributes for these 161 fonts to induce semantic attributes for all 1861 fonts in FontJoy. Therefore, the dataset for our downstream task will consist of rasterized images, typographic attributes, and semantic attributes for 1861 distinct typefaces.

3.2. k-NN Semantic Attribute Induction

We experiment with different data induction methods to regress the semantic attribute vector for each typeface in FontJoy. As our baseline, we replicate the best-performing dataset induction method reported by Kulahcioglu & de Melo (2018) for large-scale semantic signature induction with a k-NN learning model. Kulahcioglu & de Melo (2018) perform k-NN regression by calculating the distance between FontJoy embeddings using $k = 4$, the cosine distance metric, and relative distance weighting to induce the larger semantic attribute dataset from the small crowdsourced dataset of O'Donovan et al. (2014).

To perform k-NN, we use the FontJoy metadata and font names from the small 200 font O'Donovan et al. (2014) dataset to determine the intersection between the two datasets: 161 common fonts. Then, to predict semantic attributes for all 1861 fonts in FontJoy, we use the font embedding vectors obtained from the FontJoy CNN model as the input vectors on which to perform k-NN, and we take the ground-truth semantic attributes to be the data labels. Thus, the averaging over nearest neighbors in inference yields the averaged O'Donovan dataset semantic attributes of the nearest font neighbors in the FontJoy embedding space. The resulting trained model is used to induce the typographic and semantic attributes for all 1861 fonts in FontJoy.

In order to improve on the baseline model, we formulated and tested a few more k-NN models for dataset induction. We performed grid search over multiple values of k (from 1 to 10), three distance metrics—cosine, Manhattan, and Euclidean—and weighting schemes for the nearest neighbors—inverse distance, unweighted, and relative weighting. The highest-performing model was found to be the model with $k = 7$, a cosine distance metric, and inverse distance weighting. The cosine distance metric is detailed in equation (1), and the k-NN weighting of nearest neighbors f_i w.r.t. a query embedding f' with the inverse distance weighting w_i (before normalization) are shown in equation (2). For reference, relative distance weighting is shown in equation (3). Results are discussed in section 4.

$$1 - \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2}\sqrt{\sum_{i=1}^{d} y_i^2}} \tag{1}$$

$$\vec{f} = \sum_{i=1}^{k} w_i \vec{f_i} \qquad w_i = \frac{1}{|d(f', f_j)|} \tag{2}$$

$$w_i = \frac{1}{k-1} \cdot \frac{\sum_{j=1\ i\neq j}^{k} d(f', f_j)}{\sum_{j=1}^{k} d(f', f_j)} \tag{3}$$

3.3. Support Vector Regressor

For our second approach, we use Support Vector Regressors (SVRs), a model related to SVMs suited to the context of our regression problem. SVRs produce real-valued output while maintaining the main feature that characterizes the algorithm: maximal margin. Since the output is a real value, a margin of tolerance (epsilon) is set in approximation to the SVM for the problem.

In non-linear SVR models, the kernel functions transform the data into a higher dimensional feature space to make it possible to perform linear separation of the data. For our experiments, we choose the non-linear Gaussian Radial Basis Function (RBF), given that the FontJoy embeddings

are high-dimensional and require a non-linear model. The RBF kernel function is shown in equation (4).

$$K(x, x') = \exp\left(-\gamma||x - x'||^2\right). \quad (4)$$

We perform exhaustive cross-validation grid search over the reasonable hyperparameter ranges of γ (1e-3 to 1e-9) and C (1e-2 to 1e4) to tune a SVR model for each of the 31 semantic attributes. We choose parameters with the highest mean cross-validation scores. Thus, each SVR model takes a FontJoy embedding as input and outputs a real-valued number for its prediction of the value of the corresponding attribute. Intuitively, γ defines how far the influence of a single training example reaches, with lower values meaning farther. They can be interpreted as the inverse of the radius of influence of samples the model selects as support vectors. C behaves as a regularization parameter.

3.4. Mean-Shift Clustering

The mean-shift algorithm is a density-based clustering algorithm that doesn't require specifying the number of clusters as a hyperparameter. We explore these algorithms because they are cluster-shape-independent, thus not requiring any prior assumptions on dataset structure. To perform data induction, we mean-shift cluster the FontJoy embeddings in an unsupervised manner, and in inference, predict the cluster of the font and take a weighted average (inverse distance) of all of the training set fonts in the predicted cluster.

3.5. Ensembled Learning Models

For our final data induction model, to leverage the power of the three best-performing aforementioned data induction models (4-NN, 7-NN, and SVR), we ensemble a meta-estimating voting regressor that takes a weighted average over the individual predictions of these models. The weighted average is inversely proportional to the error achieved by the model. We also try a stacked regressor that stacks together the predictions of individual estimators to feed as input to a final linear regression estimator.

The error metric we use to measure the quality of each of these models in accurately inducing each of our resulting datasets for the 1861 semantic font attribute vectors, is the leave-one-out cross-validation error procedure of Kulahcioglu & de Melo (2018). We calculate this model-invariant metric for each data induction model m. We discuss this metric more precisely in section 4.

3.6. Downstream Semantic Vector Prediction Methods

To investigate how well models can predict semantic attributes given the rasterized images for a particular typeface, we test two baseline models and our proposed model.

We use two linear regression (LR) baseline models: the 1-step LR and 2-step LR. The 1-step LR predicts semantic attribute vectors directly from the pixel-linearized image input. On the other hand, the 2-step LR predicts semantic attribute vectors sequentially: it first predicts typographic attribute vectors (6-dimensional vectors denoting the attributes 'capitals', 'cursive', 'display', 'italic', 'monospace', 'serif' from the O'Donovan et al. (2014) dataset, which we also induce as aforementioned) from the linearized image input, and then predicts the semantic attribute vectors from the typographic attribute vectors. We try this 2-step LR model since results from Kulahcioglu & de Melo (2018) suggest that font categories (which are related to typographic attributes) are correlated with certain semantic attributes. We wish to compare the performance of this model to a 1-step LR model to explore the predictive power of typographic attributes as an intermediate interpretation of the image input.

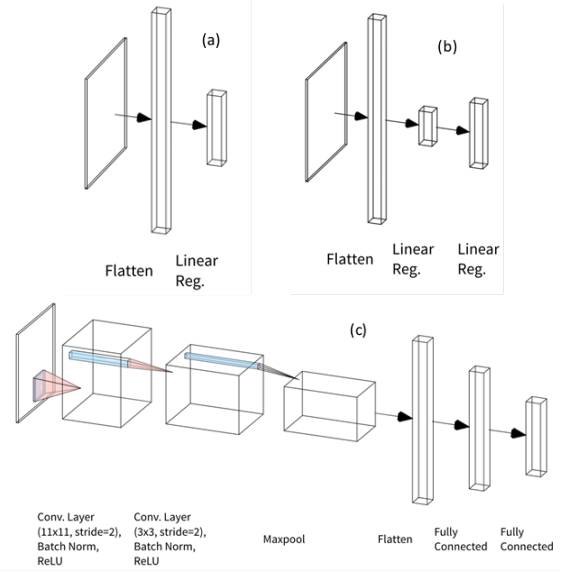


Figure 2. Proposed model architectures for semantic attribute vector prediction task: 1-step linear regressor (a), 2-step linear regressor (b), and Convolutional Neural Network (c).

Our proposed model is a deep Convolutional Neural Network (CNN) with two convolutional layers, along with batch normalization and ReLU activations. We resize all input images to 64×64 grayscale images (1 channel) and the output layer of 31 semantic attributes comes after the final fully connected layer. Figure 2 illustrates the specific architectures for each of our semantic vector prediction models.

4. Results and Discussion

4.1. Dataset Induction Results and Discussion

To evaluate our dataset induction models, we calculate the induction error for each model m using leave-one-out cross-validation per Kulahcioglu & de Melo (2018), where error

Model	Error
Lowest Error Mean-Shift	0.099
Kulahcioglu & de Melo (2018) KNN	0.084
Lowest Error KNN	0.080
Lowest Error SVR	0.079
Lowest Error Ensemble	0.076

Table 1. Data Induction Errors for each of our highest-performing models, along with baseline model Kulahcioglu & de Melo (2018).

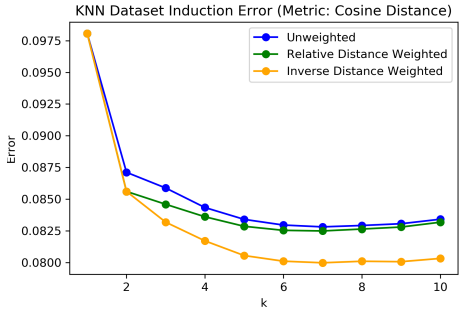


Figure 3. Dataset induction error for all k-NN experiments using the cosine distance for each of the three nearest neighbor weighting schemes across values of k. The inverse distance metric performs the best across the board and performs best at k = 7.

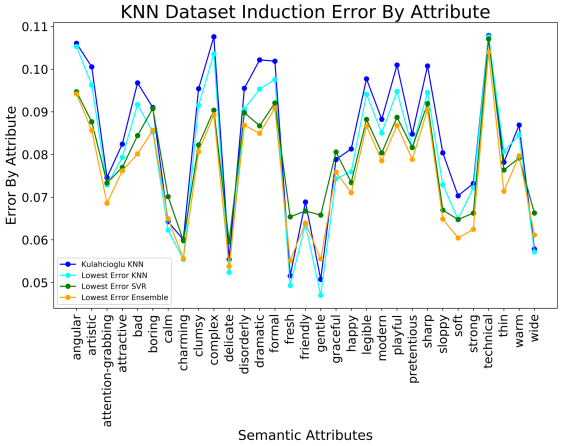


Figure 4. Dataset Induction Error of each model for each of the 31 semantic attributes. The ensemble model (orange) performs best overall as compared to the baseline (dark blue), our best-performing k-NN (cyan), and our best SVR (green).

is measured as the average in difference over semantic attributes between the predicted and the ground-truth semantic attribute vectors for the 161 fonts that are common between the FontJoy dataset and the dataset by O'Donovan et al. (2014). To be more precise, let F_{161} be the set of font embedding vectors in FontJoy with known font attribute labels, let f be a font embedding vector from the FontJoy CNN, let $\hat{a}$ be a predicted 31-dimensional semantic attribute vector, and let a be a ground truth 31-dimensional semantic attribute vector. The error metric is calculated as follows.

- For each font embedding $f \in F_{161}$, we train m on F_{161} f and predict $\hat{a}$ for f. Then, we take the element-wise difference between $\hat{a}$ and the ground truth semantic attribute vector a to calculate a 31-dimensional error vector over semantic attributes: $e = \hat{a} - a$.
- To get a total error estimate over predictions for m, we average over the element-wise absolute values of the e vectors obtained for each of the 161 fonts $f \in F_{161}$ to obtain the average error $\bar{e}$.
- The final real-valued average model error is the average over the attributes, the 31 elements of $\bar{e}$.

The results for each of our data induction models is shown in Table 1. Here, we notice that our ensemble model performs the best in terms of overall error, which makes sense, since it has the extra benefit of taking a weighted vote among three high-performing models and is thus more robust to the weaknesses of each. Notably, all of our highest-performing models outperform the baseline from the literature. We also plot the k-NN Dataset Induction Error as a function of k and each of the weighting schemes in Figure 3. The inverse distance metric performs the best across the board and performs best at $k = 7$. The plots for the Manhattan and Euclidean distance metrics look similar.

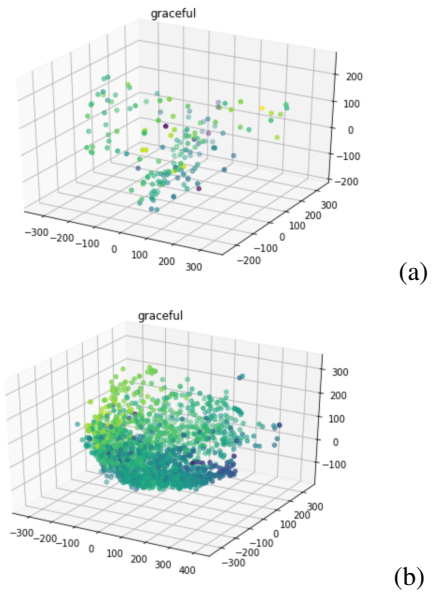
We also noticed that k-NN distance metrics that consider magnitudinal difference as well as angular distance, in particular the Manhattan and Euclidean distance, performed more poorly than distance metrics that consider only angular distance, like cosine distance. This is likely due to the high-dimensionality of the font embedding vectors.

The k-NN inverse distance weighting scheme also outperformed the relative weighting scheme. We postulate this is because weighting vectors that lie closer to the queried vector much higher than those farther from it—inversely proportional rather than proportional—reflects the fact that closer vectors in the FontJoy embedding space should hold more weight in determining semantic similarity.

In addition, the SVR outperforms the k-NN models. We believe this is because our task requires compression of higher-dimensional embedding information into lower-dimensional distance metrics. For k-NN, the cosine similarity metric completely discards magnitude information and purely uses angular distance. Conversely, the SVR kernel must transform embeddings into higher-dimensional space to make the data linearly characterizable, which we believe may allow it to preserve a more optimal combination of the original higher-dimensional information.

In Figure 4, we also plot the Dataset Induction Error for each of the 31 semantic attributes to visualize the performance of our models against the baseline. We also qualitatively

evaluate the results of our dataset induction. In Figure 5, we plot the result of PCA (reduced 31-dimensional semantic vectors to 3 dimensions) on the original 161-size (a) and ensemble-induced (b) datasets. We then color each of these semantic vectors by the value of an example semantic attribute, 'graceful'. Upon visual examination, we make two observations: first, the induced dataset colors, or attribute values, follow the general trend of the values in the original dataset. This suggests that the induction successfully reflects an overall spectrum of this attribute across the original dataset. Second, the induced dataset has smooth color transitions, which is a profound result of performing clustering using the FontJoy embeddings of nearby points. This shows that the FontJoy embeddings are valuable predictors of the relative differences in semantic attributes across fonts.



(a)

(b)

Figure 5. Reduced-dimensionality semantic vectors colored by the 'graceful' semantic attribute value for the original human-labeled dataset (a) and ensemble model induced dataset (b).

4.2. Semantic Vector Prediction Results and Discussion

For each of our three semantic vector prediction models, we compute the root mean squared error (RMSE) and coefficient of determination R^2 for both training and validation, as shown in Table 2. Most notably, the CNN outperforms both linear baseline models. We believe this is because the baselines cannot leverage spatial information since the image pixels must be linearized for linear regression. The convolutional filters can characterize glyphs by learning the relationships between features in a spatial sense, enabling it to "see" edges, curves, and higher-level characteristics.

In comparing the two baselines, we found that the 2-step LR model had a lower RMSE than the 1-step LR model. This result suggests that mapping the rasterized image input down to the intermediate typographic attributes improves the performance of the linear regression model. We can interpret this by the loose analogy for the two-step linear model as a neural network with a human-defined 6-dimensional hidden space; mapping the rasterized image input to typographic attributes acts as a forcing function that reduces the input to an intermediate typographic attribute vector subspace that has been shown to be closely linked with semantic signatures (Kulahcioglu & de Melo, 2018). While it doesn't leverage spatial information, it still learns some abstract features of the fonts before predicting semantic attributes, allowing it to outperform the single step model. Moreover, it appears that the 1-step LR overfit during training, which we believe is due to an imbalance in the high-dimensional input space and low-dimensional output space.

Model	RMSE_{tr}	R^2_{tr}	RMSE_{val}	R^2_{val}
1-step LR	1.34753	**0.99999**	0.07974	0.71879
2-step LR	0.10327	0.50344	0.11761	0.41222
CNN	**0.04263**	0.88317	**0.06434**	**0.81176**

Table 2. Results from experiments for the three proposed models in training (tr) and validation (val), with best performances bolded.

5. Conclusion

Characterizing semantic attributes of fonts is of special interest for font search, selection, pairing, and more. In this work, we leverage the existing typeface semantic attribute dataset from O'Donovan et al. (2014) (size 200) and deep font CNN embeddings from FontJoy (size 1861) to improve significantly upon prior work for inducing a larger dataset of semantic attributes for all 1861 FontJoy fonts. We create low-error 7-NN and support vector regressor models, which we ensemble with the model replicated from Kulahcioglu & de Melo (2018) to create our lowest-error semantic attribute induction model. We also introduce a novel downstream task from this resulting induced dataset of paired font images and semantic attributes—semantic attribute vector prediction from rasterized font images—for which we build a convolutional neural network that outperforms our linear regression baselines. We hope our lower-error semantic attribute dataset induction can also encourage further research in other downstream tasks related to prediction and learning of semantic typeface attributes.

6. Future Work

For semantic vector dataset induction, we hope to further investigate both unsupervised clustering algorithms, such as spectral clustering, and semi-supervised algorithms that leverage partially labeled data to induce lower-error datasets. For our downstream task, we wish to create higher-performing CNNs for rasterized images and investigate using LSTMs to process serialized SVG image commands.

Contributions

S.M. implemented the project dataloaders, performed data exploration, and created the baseline and CNN models for the downstream task. L.Z. implemented data cleansing, result visualization, and the nearest neighbors and SVR approaches for dataset induction. J.G. wrote the experimentation boilerplate code, the mean-shift clustering approach for dataset induction, and the bulk of the poster and report.

Project Code. Our code can be found at the project repository on GitHub (https://github.com/zlucia/cs229-project).

References

Avilés-Cruz, C., Villegas, J., Arechiga-Martínez, R., and Escarela-Perez, R. Unsupervised font clustering using stochastic version of the em algorithm and global texture analysis. In Sanfeliu, A., Martínez Trinidad, J. F., and Carrasco Ochoa, J. A. (eds.), *Progress in Pattern Recognition, Image Analysis and Applications*, pp. 275–286, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-30463-0.

Azadi, S., Fisher, M., Kim, V. G., Wang, Z., Shechtman, E., and Darrell, T. Multi-content GAN for few-shot font style transfer. *CoRR*, abs/1712.00516, 2017. URL http://arxiv.org/abs/1712.00516.

Kulahcioglu, T. and de Melo, G. Predicting semantic signatures of fonts. pp. 115–122, 01 2018. doi: 10.1109/ICSC.2018.00025.

Lin, X., Li, J., Zeng, H., and Ji, R. Font generation based on least squares conditional generative adversarial nets. *Multimedia Tools Appl.*, 78(1):783–797, January 2019. ISSN 1380-7501. doi: 10.1007/s11042-017-5457-4. URL https://doi.org/10.1007/s11042-017-5457-4.

Lopes, R. G., Ha, D., Eck, D., and Shlens, J. A learned representation for scalable vector graphics. *CoRR*, abs/1904.02632, 2019. URL http://arxiv.org/abs/1904.02632.

O'Donovan, P., Libeks, J., Agarwala, A., and Hertzmann, A. Exploratory Font Selection Using Crowdsourced Attributes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 33(4), 2014.

Zhang, Y., Zhang, Y., and Cai, W. Separating style and content for generalized style transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Öztürk, S., Sankur, B., and Abak, A. Font clustering and classification in document images. *European Signal Processing Conference*, 2015, 01 2000.