Fonts Like This but Happier: A New Way to Discover Fonts

Tugba Kulahcioglu
Rutgers University
Piscataway, NJ, USA
tugba.kulahcioglu@rutgers.edu

Gerard de Melo
Hasso Plattner Institute/University of Potsdam
Potsdam, Germany
gdm@demelo.org

ABSTRACT

Fonts carry strong emotional and social signals, and can affect user engagement in significant ways. Hence, selecting the right font is a very important step in the design of a multimodal artifact with text. Currently, font exploration is frequently carried out via associated social tags. Users are expected to browse through thousands of fonts tagged with certain concepts to find the one that works best for their use case. In this study, we propose a new multimodal font discovery method in which users provide a reference font together with the changes they wish to obtain in order to get closer to their ideal font. This allows for efficient and goal-driven navigation of the font space, and discovery of fonts that would otherwise likely be missed. We achieve this by learning cross-modal vector representations that connect fonts and query words.

CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**; • **Human-centered computing** → *Interactive systems and tools.*

KEYWORDS

typography; multimodal retrieval

ACM Reference Format:
Tugba Kulahcioglu and Gerard de Melo. 2020. Fonts Like This but Happier: A New Way to Discover Fonts. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413534

1 INTRODUCTION

Given that thousands of fonts are now freely available online, selecting among them is typically carried out via associated social tags. However, supporting users in deciding which fonts to pick is challenging when this is based only on such tagging. The ability of users to explore the different fonts is limited both by the incompleteness of the tagging and the limited tag inventory. If the tag inventory grows, the risk of missing tags for fonts increases. Even in an ideal scenario with a large tag inventory and in the absence of any missing tag associations, users would still suffer from the large number of fonts they would need to browse through to eventually find the ideal font for their use case.

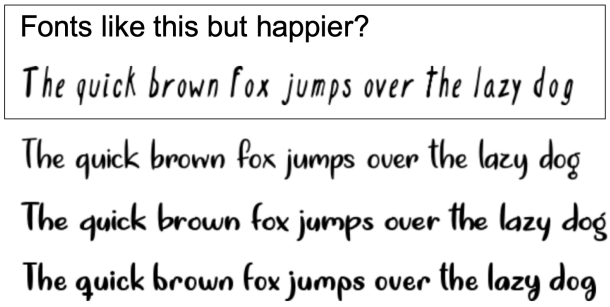

Figure 1: An example font search using the proposed multimodal querying strategy.

In a recent user study, Wu et al. interviewed design practitioners regarding their font selection process and the challenges they faced [37]. Unsurprisingly, one of the main difficulties reported by the participants was identifying fonts that match a particular semantic profile. One participant reported this as follows:

> "When I'm looking for a particular font, [I] know what feeling [I] want the font to have. But I just spent so much time browsing and browsing, and still couldn't find the one."

This suggests the need for systems that support a more open-ended form of font discovery, allowing users to search for arbitrary attribute query words, including ones that are not present as tags in the data at all.

In the same user study, the participants also expressed their desire to *slightly modify* fonts that otherwise partially fulfilled their needs but were "just a little bit off" [37]. They further emphasized the need for *unique* fonts, so as to avoid very popular fonts and better differentiate their design product from those of competitors.

In this paper, we propose a new multimodal font discovery method in which users provide a reference font that is visually similar to what they are seeking but only partially fulfills their needs, along with the changes they would like to obtain to get closer to their ideal font. Figure 1 shows an example of this. If the user likes the style of a certain font, but needs a *happier* version of it, they can provide that font as a reference and indicate the change(s) they wish to have.

Using this mechanism, the users not only satisfy their need to *slightly modify* a font, but also have the ability to explore niche sections of the available font inventory to find a *unique* font, without spending their effort on reviewing fonts that are far from what they need. We enable this form of search strategy by embedding fonts

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413534

and words into a joint cross-modal representation space, enabling the use of multimodal vector arithmetic.

The above technique not only enables novel font discovery methods, but also helps overcome other semantic challenges, specifically, the challenges of limited tag inventories and of missing font–tag connections. Users obtain access to the entire vocabulary that the language (in our case English) provides, and a font need not be tagged with the specific words that users associate them with, since the method is able to infer such connections.

The rest of the paper is organized as follows. Section 2 covers pertinent related work. Section 3 describes our data acquisition process to procure a large tagged font collection. In Section 4, we introduce our method to induce cross-modal vector representations. Section 5 then presents how we can use our method to search for fonts based on an arbitrary desired attribute, while Section 6 describes how we can invoke it to estimate font similarity, so as to find fonts based on a reference font. These are the two key building blocks of our multimodal search strategy, which is presented in Section 7. We conclude the paper in Section 8 with a brief summary and discussion of our results.

2 RELATED WORK

2.1 Font Analytics

There has been growing interest in computational approaches to analyzing fonts not just visually but with regard to their semantic associations. These studies rely on crowdsourcing [27], surveys [31], and Web data [5] to obtain a reference dataset that labels fonts with regard to semantic attributes such as *happy*, *thanksgiving*, or *pixel*. Crowdsourcing and surveys yield datasets that are small (tens-to-hundreds of fonts, tens of attributes) but clean and complete, i.e., they provide a fairly accurate labeling of every font with regard to every attribute. In contrast, crawled Web data is large (thousands of fonts, thousands of attributes) and noisy (i.e., missing many font–attribute connections). Based on small amounts of crowdsourced data, previous work [17, 19] has explored predicting attributes for new fonts, but limited to the very small original inventory of semantic attributes, without an ability to infer new tags. Another study [18] explored inferring emotions associated with fonts, which were then used to recommend fonts for a large inventory of English words based on a crowdsourced emotion lexicon. However, this method only works for a limited set of fonts for which a complete labeling of required semantic attributes is available. Chen et al. [5] use a Web dataset very similar to ours and propose a generative feature learning algorithm to infer font–tag connections. However, different from our study, they do not consider unseen tags.

2.2 Font Exploration Methods

In addition to tag based search, O'Donovan et al. [27] implement a similarity-based search method, in which they learn a font similarity metric using a crowdsourced dataset to find fonts that are similar to a reference font. The same dataset is leveraged in Section 6 of our paper, which also offers a discussion of how the results compare.

Wang et al. [36] propose a deep convolutional neural network (CNN) approach to help users identify the fonts employed in a photographic image. The hidden layers of such trained deep convolutional neural networks can also be used as vector representations of fonts [19]. In Section 6, we compare CNN embeddings with our induced vector representations, and find that our method yields improved font similarity predictions.

FontJoy [28] is an online tool that uses deep convolutional font embeddings to find pairings of fonts, i.e., fonts that share an overarching theme but have a pleasing contrast. Jiang et al. [15] present a font-pairing method based on font-pair data crawled from PDF documents on the Web.

Choi et al. [6] work with a Web dataset similar to ours, and develop an inspiration tool that provides unexpected but useful font images or concept words in response to a user query. While their tool limits the queries to known tags from the data, we explore the zero-shot case. For this, we induce a cross-modal representation to facilitate font exploration. There has been extensive work on connecting images and text [1, 7, 22, 25, 33], while our cross-modal vector space connects fonts and words.

Other methods directly seek to recommend fonts given the text that is to be rendered. To this end, Shirani et al. [34] explore a series of deep neural network models that assess a short input text and perform multi-label classification to select the best-fitting ones among 10 different display fonts. Kawaguchi & Suzuki [17] recommend fonts and colors for creating e-book cover pages automatically by classifying both fonts and the e-book text with regard to 12 emotional attributes.

2.3 Impact of Fonts

Marketing. There is substantial marketing-related research on the impact of the choice of font on consumer attitudes. One study [30] shows that fonts that are not in line with the intended message can negatively affect a company's perception in terms of professionalism, trust, and intent to act. Another study [10] determined that fonts perceived as *natural* increase the perception of associated products as being *healthy*. Velasco et al. [35] identify interesting connections between taste and fonts, e.g., *round* typefaces indicating a *sweet* taste in the context of food packaging.

Documents. There are also studies comparing the perception of the same textual content using different fonts. Shaikh et al. [32] show that different fonts may give rise to different perceptions of an email message. Juni & Gross [16] similarly find that the same text in a news article can be perceived as being funnier or angrier based on the characteristic traits of the fonts being used.

Human-Computer Interaction. Lewis & Walker [21] ask users to press a certain key if the words *slow* or *heavy* appear, and another key if *fast* or *light* appears. They repeat such tasks with fonts that match or do not match the underlying meaning of the presented words. Fonts that are coherent with the word meaning are found to decrease the user response times. Hazlett et al. [13] ask users to mark a displayed word as positive or negative. Once again, the coherence between the font connotations and the word meaning is found to increase user performance in the described task.

Visualization. Kulahcioglu et al. [20] proposed a method to generate word cloud visualizations with a particular emotional impact. To this end, their method automatically recommends suitable fonts and colors to obtain the desired effect.

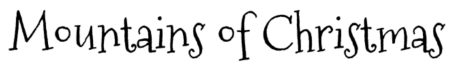

Figure 2: Sample font "Mountains of Christmas" with the tags: *serif, christmas, bouncy, staggered, curly, cute, playful, casual, warm, fun, handwritten, text, google web.*

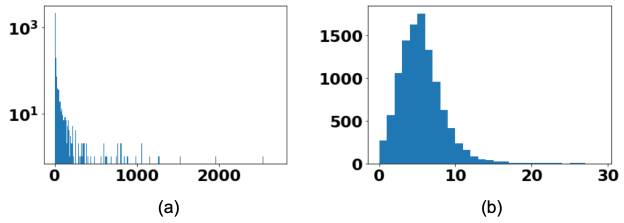

Figure 3: Histograms analyzing tag frequencies. (a) Distribution of tag frequencies across the entire dataset. (b) Distribution of tag counts per individual fonts.

3 FONT TAGGING DATA

Our study assumes a large collection of fonts along with substantial (yet incomplete) social tagging. In the following, we describe how we procure such a dataset.

3.1 Data Crawling

We collected font–tag associations from www.1001fonts.com, a website that catalogs font files along with user-assigned tags. In Figure 2, a sample font name is shown together with its associated tags. As for most such Web resources, font families are tagged as a whole, e.g., the *italic* or **bold** versions of a typeface are not tagged separately. Similar to previous work [5], we adopt the "regular" version of a font family for use in our dataset. Unlike previous studies, however, we apply a series of data cleaning steps to reduce the noise to the extent possible.

3.2 Data Cleaning

We filter out irrelevant fonts and tags as an attempt to clean otherwise noisy Web data.

3.2.1 Filtering Out Fonts. Dingbat fonts are fonts that consist entirely of symbols instead of alphabetical or numerical characters. They are used for decorative or symbolic purposes. As they are not relevant in rendering text, we discard all fonts assigned the *dingbat* tag in the data, which accounts for around 600 fonts.

3.2.2 Filtering Out Attributes. As we are interested in tags that describe semantic attributes of fonts and enable font discovery along such attributes (e.g., "happier"), we eliminate around 100 tags that merely denote font families (e.g., *serif*, *sans-serif*, *slab serif*) or other types of information (e.g., *google web*, *10pt*, *12pt*) that are not directly related to font semantics. We also eliminate a few tags that are not in English. We retain typographical tags that have the potential to provide semantic connections, such as *wide*, *handwritten*, *gothic*, *poster*, and *outlined*.

Figure 4: Tags *digital, hairline, bouncy, playful, futuristic, halloween, graffiti, handwritten,* and *children* rendered using examples of fonts tagged accordingly in the dataset.

Figure 5: Sample emotion-expressing attributes rendered using fonts tagged accordingly in the dataset.

As a concrete example, for the font given in Figure 2, the tags *serif*, *text*, *google web*, and *medium* are eliminated, leaving the font with the tags *christmas*, *bouncy*, *staggered*, *curly*, *cute*, *playful*, *casual*, *warm*, *fun*, *handwritten*, and *light*.

3.3 Dataset Summary

After the above filtering, the resulting dataset contains around 10.4K fonts, 2.6K tags, and 54K font–tag assignments, with an average of 5 tags per font. Figure 3 shows the distributions of (a) overall tag frequencies and (b) tag counts per font. Most tags are used to tag fewer than a hundred fonts, and most fonts have fewer than 10 tags. Figure 4 displays three font examples for nine selected tags from the dataset, aiming to give a feeling of the range of the semantic connections. Figure 5 provides examples of fonts for sample emotion-expressing attributes. Figure 8 in Section 5 also provides examples of fonts for the ten most frequent attributes.

4 CROSS-MODAL REPRESENTATION LEARNING

In order to facilitate identifying fonts that are similar to a given input font but differ along a particular attribute ("like this but happier"), we induce a cross-modal vector representation space. This not only allows us to jointly embed both fonts and query

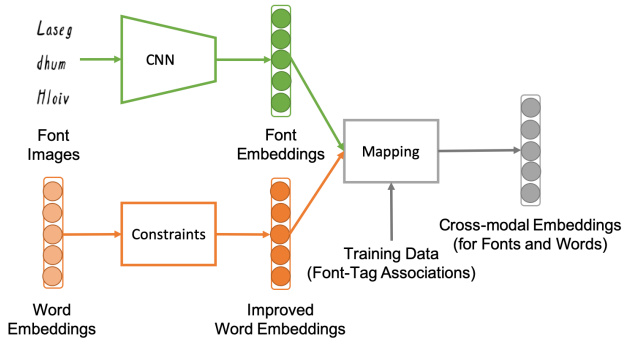

Figure 6: Overview of the proposed cross-modal representation induction method.

words in a single vector space, but also allows us to conduct vector arithmetic to locate fonts that better match a given semantic profile.

Our vector space induction method is summarized in Figure 6. We induce font embeddings using a deep convolutional neural network, and induce word embeddings by modifying pretrained distributed word embeddings to better satisfy antonymy and synonymy constraints. The final step is to connect the aforementioned font and word embeddings in a single cross-modal vector space.

Our method assumes as input a set $\mathcal{F}$ of fonts, which are associated with a set $\mathcal{A}$ of font attributes via a Boolean font–attribute matrix $\mathbf{M} \in \{0,1\}^{|\mathcal{F}|\times|\mathcal{A}|}$ based on the data described in Section 3.

4.1 Font Embedding Induction

Our first goal is obtain a font embedding matrix $\mathbf{F} \in \mathbb{R}^{|\mathcal{F}|\times d}$ that in its rows provides a d-dimensional vector representation $\mathbf{v}_f \in \mathbb{R}^d$ for each font $f \in \mathcal{F}$.

These vector representations are expected to reflect visual similarity, i.e., fonts f, f' that are visually similar ought to have similar vectors $\mathbf{v}_f$, $\mathbf{v}_{f'}$. To achieve this, for each font $f \in \mathcal{F}$, we generate an image rendering a fixed set of 14 different letters from the alphabet using that font so as to demonstrate its visual characteristics.

We then feed these images into a deep convolutional neural network with residual connections, specifically a ResNet-18 [14] model pre-trained on ImageNet [8]. For each font, we extract the resulting 512-dimensional latent representation from the average pooling layer of the model.

Finally, for dimensionality reduction to $d = 300$ dimensions, we apply Principal Component Analysis (PCA) and project every latent font representation into the space spanned by the first d principal components in order to obtain the desired matrix $\mathbf{F}$ with d-dimensional vectors $\mathbf{v}_f \in \mathbb{R}^d$ for fonts $f \in \mathcal{F}$.

4.2 Word Embedding Induction

Our next goal is to induce vector representations of tags. We start out with the widely used 300-dimensional word2vec vectors pretrained on a large Google News dataset [23], which provides a word embedding matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}|\times d}$ for a large vocabulary $\mathcal{V}$ of English words. The vectors are based on contextual information and the corresponding vector similarities reflect distributional similarity.

However, distributional similarity in general and word2vec word vectors in particular tend to give similar representations to words with opposite meaning such as *formal* and *informal* [26]. To alleviate this issue, we apply the Counter-fitting algorithm [24] to transform the original word embedding matrix $\mathbf{W}$ into a new embedding matrix $\mathbf{W}'$ subject to antonymy constraints A and synonymy constraints S. The algorithm minimizes the loss function

$$\begin{aligned}\ell(\mathbf{W},\mathbf{W}') = &\sum_{(u,w)\in A} 1 - d(\mathbf{v}'_u,\mathbf{v}'_w)\\ &+ \sum_{(u,w)\in S} d(\mathbf{v}'_u,\mathbf{v}'_w)\\ &+ \sum_{w\in\mathcal{V}}\sum_{u\in N(w)} \max(\mathbf{0}, d(\mathbf{v}'_u,\mathbf{v}'_w) - d(\mathbf{v}_u,\mathbf{v}_w)),\end{aligned} \quad (1)$$

where the notation $\mathbf{v}_w$ denotes the vector for w in $\mathbf{W}$, $\mathbf{v}'_w$ denotes the vector for w in $\mathbf{W}'$, and $N(w)$ denotes the set of nearest neighbors of w in $\mathbf{W}'$ with cosine similarity $\geq \tau = 0.8$. For the setting of τ as well as the constraint sets A and S, which are extracted from PPDB [11] and WordNet [9], we follow the original study [24].

The resulting output embeddings $\mathbf{W}'$ are of the same dimensionality as the input embeddings, i.e., 300-dimensional.

4.3 Cross-Modal Font–Word Representations

Finally, we induce a cross-modal vector space that jointly embeds both fonts and words. We start out with the font embedding matrix $\mathbf{F}$ from Section 4.1 and the modified word embedding matrix $\mathbf{W}'$ from Section 4.2.

In order to be able to connect these two spaces, we draw on the font–attribute matrix $\mathbf{M} \in \{0,1\}^{|\mathcal{F}|\times|\mathcal{A}|}$ based on the tagging data described in Section 3, from which we enumerate a set of pairs

$$M = \left\{ (i,j) \;\middle|\; w_j \in \mathcal{V}, i \in \operatorname*{argmax}_{I\subset\{i|m_{ij}>0\},|I|\leq k} \sum_{i\in I} \frac{1}{\sum_{j'} m_{ij'}} \right\}. \quad (2)$$

Thus, for each tag w_j in our word embedding vocabulary $\mathcal{V}$, we retain the top-k fonts ranked in terms of the inverse of the number of tags those fonts have. The intuition of this filtering is that fonts with fewer non-zero entries $m_{ij'}$ in $\mathbf{M}$ tend to more specifically represent their tags compared to fonts that have numerous different tags. In our experiments, we consider different choices of k.

We construct new font and word alignment matrices $\mathbf{F}_0$, $\mathbf{W}_0$, such that the n-th row contains the font embedding from a normalized version of $\mathbf{F}$ for the font in the n-th entry in M, or the word embedding from a normalized version of $\mathbf{W}'$ for the tag mentioned in that entry, respectively. For this normalization of $\mathbf{F}$ and $\mathbf{W}'$, we first normalize each row to have a length of 1, then apply column-wise mean centering, and thereafter re-normalize each row to again have unit length. [4]. To facilitate a mapping between the font and word representations, we follow a framework originally proposed for cross-lingual alignment [3]. We apply a variant of Mahalanobis whitening by computing $\mathbf{F}_1 = \mathbf{F}_0(\mathbf{F}_0^\intercal\mathbf{F}_0)^{-\frac{1}{2}}$, $\mathbf{W}_1 = \mathbf{W}_0(\mathbf{W}_0^\intercal\mathbf{W}_0)^{-\frac{1}{2}}$ so as to decorrelate different columns, as this simplifies the cross-modal mapping.

To learn a mapping, we solve what is known as the Procrustes problem, which, following Schönemann (1966) [29], can be achieved

by computing a singular value decomposition (SVD) of $\mathbf{F}_1^\intercal \mathbf{W}_1$ as $\mathbf{U}\Sigma\mathbf{V}^\intercal = \mathbf{F}_1^\intercal \mathbf{W}_1$ to obtain orthogonal projection matrices $\mathbf{U}$, $\mathbf{V}$ of the two spaces into a single target space. We apply this mapping as $\mathbf{F}_2 = \mathbf{F}_1\mathbf{U}\Sigma^{\frac{1}{2}}$ and $\mathbf{W}_2 = \mathbf{W}_1\mathbf{V}\Sigma^{\frac{1}{2}}$, where $\Sigma^{\frac{1}{2}}$ is incorporated for a symmetric reweighting of the columns in both matrices according to their cross-correlation. Subsequently, we apply a coloring operation that reverses the aforementioned Mahalanobis whitening, by computing $\mathbf{F}_3 = \mathbf{F}_2\mathbf{U}(\mathbf{F}_0^\intercal \mathbf{F}_0)^{\frac{1}{2}}\mathbf{U}$ and $\mathbf{W}_3 = \mathbf{W}_2\mathbf{V}(\mathbf{W}_0^\intercal \mathbf{W}_0)^{\frac{1}{2}}\mathbf{V}$.

The final cross-modal output embedding matrix $\mathbf{E}$ provides vectors for fonts $f \in \mathcal{F}$ taken from $\mathbf{F}_3$ in its first $|\mathcal{F}|$ rows and subsequently provides vectors for words $w \in \mathcal{V}$ taken from $\mathbf{W}_3$.

5 ZERO-SHOT ATTRIBUTE-BASED RETRIEVAL

In this section, we describe and evaluate how our cross-modal embeddings enable zero-shot support for novel attributes. The goal is to be able to retrieve suitable fonts for a new attribute a that does not at all occur in the font–tag dataset used to induce the embeddings. In light of the incompleteness of social tags, this is an important task for the font domain. Additionally, it is also important as an indicator of the potential of our proposed multimodal discovery method, for which it serves as a key building block.

5.1 Method

We first obtain a cross-modal embedding matrix $\mathbf{E}$ following the three steps of our technique as described in Section 4. To predict the fonts associated with an attribute $a \in \mathcal{V}$, even if $a \notin \mathcal{A}$, we can consult $\mathbf{E}$ to obtain the cross-modal embedding $\mathbf{e}_a$ for a as well as the cross-modal embedding vectors $\mathbf{e}_f$ for fonts f, and simply select those fonts $f \in \mathcal{F}$ that maximize

$$\frac{\mathbf{e}_f^\intercal \mathbf{e}_a}{||\mathbf{e}_f||_2 \, ||\mathbf{e}_a||_2}, \tag{3}$$

i.e., the ones most similar to a in terms of cosine similarity.

5.2 Evaluation

To evaluate this, we apply the above method for the 100 most frequent attributes a in $\mathcal{A}$ using leave-one-out cross-validation. Thus, for each target attribute a, we separately induce a different cross-modal embedding matrix $\mathbf{E}$ based only on the data for $\mathcal{A} \setminus a$, i.e., excluding a completely from $\mathbf{M}$. The above method is used to retrieve suitable fonts for attribute a without it having observed any annotations of this attribute at all.

Figure 8 shows top three fonts as predicted by this method for the most frequent ten attributes. As an example, for the attribute *handwritten*, representations are induced on the data excluding any tagging of fonts with the tag *handwritten*. The three fonts presented for *handwritten* are fonts with font vectors of the highest cosine similarity to our vector representation of the word *handwritten*.

The check marks next to the fonts indicate that the font is tagged with the corresponding attribute in the Web dataset, and hence the prediction is deemed accurate. The second font for *handwritten* has this symbol, confirming its accuracy. Nonetheless, as the Web dataset is known to have missing tag annotations, the lack of an association in the dataset does not necessarily mean that the prediction is inaccurate. In the case of *handwritten*, all of the three predicted fonts appear to represent the attribute, thus being accurate predictions.

Figure 7: Top three fonts for the attributes *narrow* and *wide* as predicted by cross-modal embeddings based on unconstrained original word vectors (without our modification, $k = \infty$).

Table 1: Mean reciprocal rank results for the 10, 50, and 100 most frequent attributes. Unconstrained representations: cross-modal embeddings based on the original unconstrained word vectors. Full method: cross-modal embeddings obtained based on modified word vectors, connected to fonts using different font filtering thresholds k.

	Top 10	Top 50	Top 100
Unconstrained representations	0.31	0.33	0.22
Full method – $k = 1$ Filtering	0.46	0.30	0.19
Full method – $k = 10$ Filtering	0.46	0.35	0.23
Full method – $k = 50$ Filtering	**0.54**	**0.46**	**0.33**
Full method – $k = 100$ Filtering	0.45	0.40	0.28
Full method – $k = \infty$ (No Filtering)	0.49	0.34	0.23

To quantitatively evaluate the results in this setting, *precision* and *recall* are not well-suited, due to the incomplete tag annotations. Instead, in Table 1 we report the *mean reciprocal rank* for the top 10, top 50, and top 100 most frequent attributes, which is based on the rank of the first predicted font for an attribute that is also tagged as such in the Web dataset. This gives us a lower bound on the performance of our method. In addition to the results for our full method, for comparison, we also evaluate a variant of our method that omits the constraint procedure from Section 4.2 and instead projects regular word2vec embeddings into a common space with font embeddings, without filtering ($k = \infty$).

5.3 Results

Based on the results, our method retrieves fonts that are tagged with the corresponding tag in the Web dataset in very early positions of the ranked list; i.e., approximately the 2nd result for the top 50 attributes, and the 3rd result for the top 100 most frequent attributes when using top $k = 50$ fonts for training (Section 4.3).

Our full cross-modal induction procedure results in better performance compared to the unconstrained variant, as the latter is more likely to conflate attributes with different meanings. Figure 7 shows the top three predictions for the attributes *wide* and *narrow*

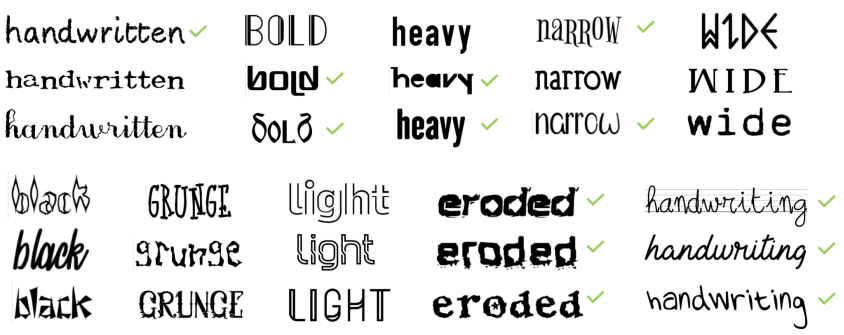

Figure 8: Top three fonts for the most frequent ten attributes as predicted by our zero-shot attribute-based retrieval method. The check marks represent results that are also tagged with the corresponding attribute in the Web dataset.

using the unconstrained variant. The fonts seem to represent attributes that are antonyms of the intended attributes. This explains the difference in performance between the two approaches.

In Figure 8, we observe that, in certain instances, the intended meaning of an attribute is different in the Web dataset compared to the word vectors. For example, the word *black* as a font tag is typically used to represent very thick typefaces, while based on the word vectors, it appears to be interpreted as a *dark* and *pessimist* concept in zero-shot attribute-based prediction. Note that this issue can easily be avoided if we move away from the zero-shot setting and instead incorporate a few instances of the tag into our training.

Another interesting observation is that ambiguity may affect the results in some cases. The tag *light*, for instance, is commonly used to characterize fonts with thin lines. However, in our case, the top three most similar fonts show other kinds of characteristics that may creatively exemplify being *light* in the sense of not being heavy, or perhaps are associated with *light* in the sense of lighting. Technically, distributional word vectors encode a linear superposition of all observed senses of a word [2]. Similarly, tags in social tagging platforms are often used in ambiguous ways [12].

6 ZERO-SHOT FONT SIMILARITY

We proceed to show how our cross-modal representations enable the prediction of font similarity scores in a zero-shot setting, i.e., for fonts for which we do not possess any social tag annotations. This as well is a useful building block for many font-related tasks, including our proposed multimodal discovery method.

For evaluation, we draw on a crowdsourced dataset from O'Donovan et al. [27]. In their study, in each task, a user was given a reference font and asked to select one out of two provided font options that are most similar to the reference font. The dataset (which will be referred to as $\mathcal{T}$) contains 2,340 such questions using 200 fonts, and a total of 35,287 user responses. In our experiments, we exclude questions and responses related to one single specific font for which we were not able to obtain the font file.

6.1 Method

We first train a cross-modal embedding matrix $\mathbf{E}$ as described in Section 4. For a question with a reference font f_r, and possibly similar font options f_a and f_b, we consult $\mathbf{E}$ to obtain the corresponding vectors $\mathbf{e}_{f_\mathrm{r}}$, $\mathbf{e}_{f_\mathrm{a}}$, $\mathbf{e}_{f_\mathrm{b}}$ and simply select the font

$$f = \underset{f \in \{f_\mathrm{a}, f_\mathrm{b}\}}{\operatorname{argmax}} \frac{\mathbf{e}_f^\mathsf{T} \mathbf{e}_{f_\mathrm{r}}}{||\mathbf{e}_f|| \, ||\mathbf{e}_{f_\mathrm{r}}||}. \tag{4}$$

Table 2: Individual user choice based experiment results for different user agreement levels. The column for the 0.5 agreement shows the results for the entire dataset, as the agreement cannot be lower than 0.5. Oracle shows the maximum possible accuracy, as users don't always agree.

	User Agreement					
	≥0.5	≥0.6	≥0.7	≥0.8	≥0.9	=1
Font	70.20	73.05	77.06	81.20	86.48	90.64
Unconstr.	70.50	73.36	77.51	81.80	87.14	90.68
Full ($k = \infty$)	70.85	73.77	77.94	82.31	87.70	91.55
Full ($k = 50$)	**70.89**	**73.86**	**78.07**	**82.44**	**88.03**	**91.78**
Oracle	81.29	85.23	89.51	93.53	97.26	100.00

Table 3: Majority-choice based experiment results for different user agreement levels. The column for the 0.5 agreement shows the results for the entire dataset, as the agreement cannot be lower than 0.5. The maximum accuracy in each column is 100%, as for each question, the option with the majority of the votes is considered as the user choice.

	User Agreement					
	≥0.5	≥0.6	≥0.7	≥0.8	≥0.9	=1
Font	77.39	79.25	82.68	84.97	88.21	90.43
Unconstr.	77.78	79.45	83.10	85.67	89.08	90.61
Full ($k = \infty$)	**78.26**	80.00	83.58	86.21	89.63	91.48
Full ($k = 50$)	78.00	**80.05**	**83.71**	**86.29**	**89.96**	**91.65**

6.2 Evaluation

We ensure a zero-shot evaluation setting by obtaining a new cross-modal embedding matrix $\mathbf{E}$ based on training data M that includes only tag associations for fonts $f \in \mathcal{F} \setminus \mathcal{T}$, i.e., fonts not considered

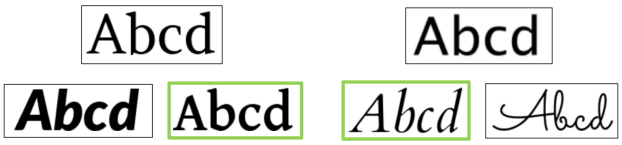

Figure 9: Sample high user-agreement questions that our method also agrees with (i.e., all users and our method pick the highlighted options).

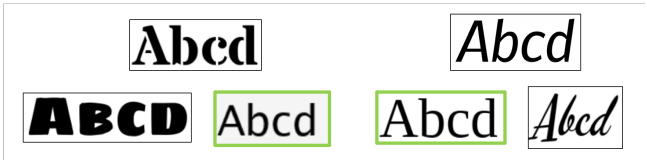

Figure 10: Sample high user-agreement questions that are answered differently by our method (i.e., all users select the highlighted options, whereas our method selects the others).

in the evaluation data. For each question in the referred dataset, we can then predict the answer for the similarity question and compare it against the answers provided by the human annotators.

For some questions, users have strong agreement (e.g., all users that answered the question select the same option), whereas for others the agreement is lower (e.g., 8 users selecting option A, 7 users selecting option B). We thus analyze the performance of our method for different user agreement levels. Figures 9 and 10 show examples of questions with high user agreement, where our method agrees with the users on the examples from Figure 9 and disagrees with them for the ones in Figure 10.

We provide quantitative results in Tables 2 and 3. The results in Table 2 consider each user response for a question as a separate data point, and report the percentage of agreement between our method and user responses for different user agreement thresholds. In contrast, the results in Table 3 consider each question as a data point, and assume that the option with the majority of the user votes is deemed the correct response for that question.

6.3 Results

In both tables, results are compared for our original font embeddings **F** ("Font"), unconstrained ("Unconstr.") cross-modal embeddings obtained using original word2vec word vectors without the constraint-based modification from Section 4.2 and with $k = \infty$, and our full-fledged method to obtain the cross-modal embedding matrix **E** ("Full").

Overall, for all user agreement levels, the best results are obtained using our full method. In all but one cases, our method obtains the best results when filtering top $k = 50$ fonts as described in Section 4.3, compared to using all available data for training.

This shows that our method of incorporating semantic information into the visual font embeddings via cross-modal alignment yields a representation that is slightly closer to human perception.

We find that as the user agreement increases, the accuracy of our method also increases. Analyzing the disagreements, one of the insights is that users very rarely rate an all-caps font as similar to a mixed-case font, whereas our method is likely to do so, such as for the question on the left in Figure 10. Such preferences could be learned using a supervised font similarity method.

Figure 11: Multimodal query samples with top-1 results.

Our unsupervised results come fairly close to the supervised results of O'Donovan et al. [27], who were able to reach an overall individual user choice accuracy of 76.04% (where the oracle upper bound is 80.79%) on the same evaluation dataset, except for the one missing font in our experiments. Their method, however, is a supervised one that learns a similarity metric on a training fold of this evaluation dataset, and also uses a complete labeling of a set of semantic attributes of the fonts, whereas our method is completely unsupervised with regard to font similarity, and, as mentioned above, we also completely omit any available tag information about the tested fonts in order to make it a zero-shot experiment.

7 MULTIMODAL FONT DISCOVERY

At this point, the results from Section 6 show how our cross-modal representations allow us to find similar fonts based on a reference font, while earlier, in Section 5, we saw how we can find fonts matching a desired attribute specification.

In this section, we show how these two notions can be combined to enable a novel form of multimodal font discovery, and demonstrate its results through sample queries.

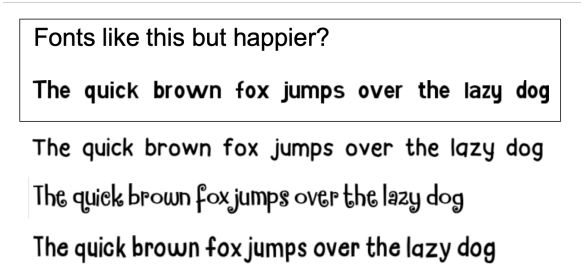

Figure 12: A multimodal query providing alternative directions in the top-3 results.

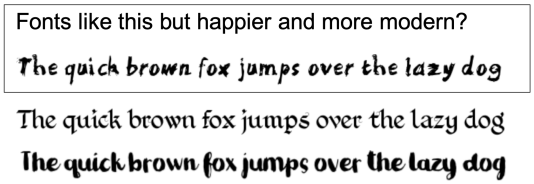

Figure 13: Multimodal query sample with two attributes used to modify the reference font.

Figure 14: Multimodal query samples with top-2 results.

7.1 Method

We create the cross-modal embedding matrix $\mathbf{E}$ as described in Section 4 using the Web font–tag dataset detailed in Section 3. For a given font $f \in \mathcal{F}$ and any suitable word $w \in \mathcal{V}$, our goal is to find fonts $f' \in \mathcal{F}$ that are similar to f and, at the same time, represent the attribute w. Note that this word w need not have occurred as a tag in our tagging dataset.

We first lookup in $\mathbf{E}$ the cross-modal representations of f and w as $\mathbf{e}_f$, $\mathbf{e}_w$, respectively, and then compute a target cross-modal representation

$$\mathbf{e}_t = \mathbf{e}_f + \mathbf{e}_w. \tag{5}$$

Given this target, we select the fonts $f' \in \mathcal{F}$ that maximize

$$\frac{\mathbf{e}_t^\top \mathbf{e}_{f'}}{||\mathbf{e}_t|| \, ||\mathbf{e}_{f'}||}. \tag{6}$$

7.2 Examples

We demonstrate our method using results for sample queries. Figure 11 showcases sample queries that yield potentially relevant fonts as the top-1 result. The samples cover the query attributes *futuristic*, *confident*, *elegant*, *fun*, and *professional*, together with reference fonts with strong profiles. The multimodal queries are able to achieve the modifications mandated by the specified attributes while retaining the visual aesthetics of the reference fonts. In another example given in Figure 12, the second result appears to be significantly different from the first and third results. Yet, all results seem relevant to the query. This variety enables users to navigate in different directions in their intended search space. The examples from Figure 13 show that it is also possible to expand our method to include multiple attributes. This is particularly useful when the user does not have any particular reference font as a starting point but instead simply starts from a neutral default one.

Limitations. As observed in the experiments from Section 6, in some cases, users' perception of similarity can diverge from the embedding's notion of similarity. The top query in Figure 14 shows a case where an outlined reference font yields a first result that is not an outlined font. Despite the similarity between the reference font and the first result, user experiments are needed to assess to what extent users would agree. Another issue is that for reference fonts that already strongly incorporate the specified attribute, the results may not seem as strong, as e.g., for the second query in Figure 14. Thus, further research is necessary to evaluate how users perceive the query results for reference fonts depending on how strongly the reference already reflects the specified attribute (e.g., strongly reflects, weakly reflects, does not reflect it).

8 CONCLUSION

In this paper, we develop a cross-modal representation for fonts and words, and use it to enable zero-shot attribute-based font retrieval as well as similarity-based font retrieval. Our experiments provide insights on properties of cross-modal embeddings for fonts and words. Tag-based retrieval requires an accurate representation space that properly reflects contrasts between different attributes. Accordingly, our full method based on semantic constraints and top-k training data filtering shows improved results compared to the unconstrained baseline.

We further show that font and attribute-based retrieval can be combined by proposing a novel multimodal font searching strategy that allows the user to specify a reference font together with the changes they wish to solicit. This allows users to quickly locate new fonts that may better satisfy their design requirements.

In terms of future work, one observation, discussed in Section 5, is that for ambiguous words, the distribution of meanings may differ between the typographical tags and the general word embeddings. We focus mostly on semantic attributes in this study, rather than typographic ones, and see the interaction between the context of the two as a direction for future work.

REFERENCES

[1] Malihe Alikhani, Sreyasi Nag Chowdhury, Gerard de Melo, and Matthew Stone. 2019. CITE: A Corpus Of Text-Image Discourse Relations. In *Proceedings of 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)* (Minneapolis, MN, USA). Association for Computational Linguistics, Stroudsburg, PA, USA, 570–575. https://www.aclweb.org/anthology/N19-1056

[2] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear Algebraic Structure of Word Senses, with Applications to Polysemy. *Transactions of the Association for Computational Linguistics* 6 (2018), 483–495. https://doi.org/10.1162/tacl_a_00034

[3] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and Improving Bilingual Word Embedding Mappings with a Multi-Step Framework of Linear Transformations. In *AAAI Conference on Artificial Intelligence*. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16935

[4] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 789–798. https://doi.org/10.18653/v1/P18-1073

[5] Tianlang Chen, Zhaowen Wang, Ning Xu, Hailin Jin, and Jiebo Luo. 2019. Large-scale Tag-based Font Retrieval with Generative Feature Learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 9116–9125.

[6] Saemi Choi, Shun Matsumura, and Kiyoharu Aizawa. 2019. Assist Users' Interactions in Font Search with Unexpected but Useful Concepts Generated by Multimodal Learning. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. 235–243.

[7] Gerard de Melo and Gerhard Weikum. 2010. Providing Multilingual, Multimodal Answers to Lexical Database Queries. In *Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010)* (Valetta, Malta). ELRA, Paris, France, 348–355. http://www.lrec-conf.org/proceedings/lrec2010/summaries/312.html

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 248–255.

[9] Christiane Fellbaum (Ed.). 1998. *WordNet: An Electronic Lexical Database*. The MIT Press. http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20{&}path=ASIN/026206197X

[10] Alysha Fligner. 2013. *The effect of packaging typeface on product perception and evaluation*. Ph.D. Dissertation. The Ohio State University.

[11] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Atlanta, Georgia, 758–764. https://www.aclweb.org/anthology/N13-1092

[12] Jonathan Gemmell, Maryam Ramezani, Thomas Schimoler, Laura Christiansen, and Bamshad Mobasher. 2009. The Impact of Ambiguity and Redundancy on Tag Recommendation in Folksonomies. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, New York, USA) *(RecSys '09)*. Association for Computing Machinery, New York, NY, USA, 45–52. https://doi.org/10.1145/1639714.1639724

[13] Richard L Hazlett, Kevin Larson, A Dawn Shaikh, and Barbara S Chaparo. 2013. Two studies on how a typeface congruent with content can enhance onscreen communication. *Information Design Journal* 20, 3 (2013).

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[15] Shuhui Jiang, Zhaowen Wang, Aaron Hertzmann, Hailin Jin, and Yun Fu. 2019. Visual Font Pairing. *IEEE Transactions on Multimedia* (2019).

[16] Samuel Juni and Julie S Gross. 2008. Emotional and persuasive perception of fonts. *Perceptual and motor skills* 106, 1 (2008), 35–42.

[17] Haruka Kawaguchi and Nobutaka Suzuki. 2018. Recommending Colors and Fonts for Cover Page of EPUB Book. In *Proceedings of the ACM Symposium on Document Engineering 2018* (Halifax, NS, Canada) *(DocEng '18)*. Association for Computing Machinery, New York, NY, USA, Article 48, 4 pages. https://doi.org/10.1145/3209280.3229086

[18] Tugba Kulahcioglu and Gerard de Melo. 2018. FontLex: A Typographical Lexicon based on Affective Associations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. http://www.lrec-conf.org/proceedings/lrec2018/summaries/1059.html

[19] Tugba Kulahcioglu and Gerard de Melo. 2018. Predicting Semantic Signatures of Fonts. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. IEEE, 115–122. https://ieeexplore.ieee.org/document/8334448

[20] Tugba Kulahcioglu and Gerard de Melo. 2019. Paralinguistic recommendations for affective word clouds. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 132–143.

[21] Clive Lewis and Peter Walker. 1989. Typographic influences on reading. *British Journal of Psychology* 80, 2 (1989), 241–257.

[22] Bingchen Liu, Kunpeng Song, Yizhe Zhu, Gerard de Melo, and Ahmed Elgammal. 2020. TIME: Text and Image Mutual-Translation Adversarial Networks. *ArXiv* 2005.13192 (2020). https://arxiv.org/abs/2005.13192

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[24] Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892* (2016).

[25] Sreyasi Nag Chowdhury, William Cheng, Gerard de Melo, Simon Razniewski, and Gerhard Weikum. 2020. Illustrate Your Story: Enriching Text with Images. In *Proceedings of WSDM 2020* (Houston, TX, USA). ACM, New York, NY, USA. https://dl.acm.org/doi/abs/10.1145/3336191.3371866

[26] Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, 454–459. https://doi.org/10.18653/v1/P16-2074

[27] Peter O'Donovan, Janis Libeks, Aseem Agarwala, and Aaron Hertzmann. 2014. Exploratory font selection using crowdsourced attributes. *ACM Trans. Graph.* 33, 4 (2014), 92:1–92:9.

[28] Jack Qiao. [n.d.]. Fontjoy – Generate font combinations with deep learning. http://fontjoy.com/. Accessed: 2017-07-29.

[29] Peter Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika* 31, 1 (1966), 1–10.

[30] A Dawn Shaikh. 2007. The Effect of Website Typeface Appropriateness on the Perception of a Company's Ethos. *Usability News* 9, 2 (2007).

[31] A Dawn Shaikh, Barbara S Chaparro, and Doug Fox. 2006. Perception of fonts: Perceived personality traits and uses. *Usability News* 8, 1 (2006), 1–6.

[32] A Dawn Shaikh, Doug Fox, and Barbara S Chaparro. 2007. The effect of typeface on the perception of email. *Usability News* 9, 1 (2007), 1–7.

[33] Lei Shi, Kai Shuang, Shijie Geng, Peng Su, Zhengkai Jiang, Peng Gao, Zuohui Fu, Gerard de Melo, and Sen Su. 2020. Contrastive Visual-Linguistic Pretraining. *ArXiv* 2007.13135 (2020). https://arxiv.org/abs/2007.13135

[34] Amirreza Shirani, Franck Dernoncourt, Jose Echevarria, Paul Asente, Nedim Lipka, and Thamar Solorio. 2020. Let Me Choose: From Verbal Context to Font Selection. In *Proceedings of ACL 2020*. Association for Computational Linguistics.

[35] Carlos Velasco, Alejandro Salgado-Montejo, Fernando Marmolejo-Ramos, and Charles Spence. 2014. Predictive packaging design: Tasting shapes, typefaces, names, and sounds. *Food Quality and Preference* 34 (2014), 88 – 95.

[36] Zhangyang Wang, Jianchao Yang, Hailin Jin, Eli Shechtman, Aseem Agarwala, Jonathan Brandt, and Thomas S Huang. 2015. DeepFont: Identify your font from an image. In *Proc. ACM Multimedia*. ACM, 451–459.

[37] Y Wayne Wu, Michael Gilbert, and Elizabeth Churchill. 2019. "I Kept Browsing and Browsing, But Still Couldn't Find the One": Salient Factors and Challenges in Online Typeface Selection. In *IFIP Conference on Human-Computer Interaction*. Springer, 225–234.