Semantic Color Mapping: A Pipeline for Assigning Meaningful Colors to Text

Mennatallah El-Assady[1], Rebecca Kehlbeck[2], Yannick Metz[2],
Udo Schlegel[2], Rita Sevastjanova[2], Fabian Sperrle[2], and Thilo Spinner[2]*
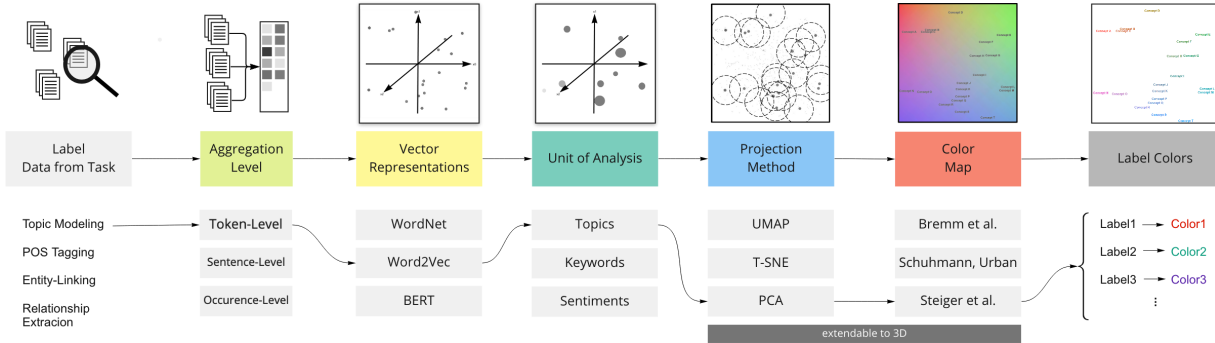
[1]ETH AI Center   [2]University of Konstanz

Figure 1: We structure the color map creation process into distinct stages: Based on the task, we (1) aggregate the data, (2) then choose a method to produce vector representations. (3) These representations can be combined based on a chosen unit of analysis. Afterward, we (4) use a projection method to map the units' high-dimensional vectors to 2- or 3-dimensional coordinates, which (5) can be assigned to color. The result is a semantically meaningful and informative color map. At each step of the pipeline, we are confronted with several design choices, which depend on the data, task, and processing steps. Some of the possible methods available at each stage are represented at the bottom. The highlighted path represents an exemplary sequence of design decisions.

ABSTRACT

Current visual text analytics applications do not regard color assignment as a prominent design consideration. We argue that there is a need for applying meaningful colors to text, enhancing comprehension and comparability. Hence, in this paper, we present a guideline to facilitate the choice of colors in text visualizations. The *semantic color mapping* pipeline is derived from literature and experiences in text visualization design and sums up design considerations, lessons learned, and best practices. The proposed pipeline starts by extracting labeled data from raw text, choosing an aggregation level to create an appropriate vector representation, then defining the unit of analysis to project the data into a low-dimensional space, and finally assigning colors based on the selected color space. We argue that applying such a pipeline enhances the understanding of attribute relations in text visualizations, as confirmed by two applications.

1 INTRODUCTION

Color is one of the most prominent visual variables in visualization design. Mapping data variables and attributes to semantically meaningful colors is among the main challenges for visualization creators [13]. From mapping concepts to colors based on cultural references [15, 20] to mapping them based on image queries [32, 39], researchers have suggested a multitude of techniques to bestow meaning to color. However, we need an alternative color assignment technique for applications where concepts do not have an inherent color association or where the distribution of color assignments would not make full use of the available color space.

This challenge is quite apparent for text data visualization. Text data is usually processed using computational linguistics algorithms before it is visualized [23]. After processing, the raw (unstructured)

text data is commonly transformed into (structured) hierarchical and high-dimensional data. For example, frequent concepts, topics, emotions, named entities, or other attributes can be extracted and visualized. Most commonly, these attributes are assigned colors based on a categorical color mapping that does not depict the inherent relation between the attributes. Surveys of text visualizations [17] show the lack of thoughtful color assignment in existing approaches.

As an example, let us consider an application that analyzes the thematic structure of a document collection and assigns a topic to each document. These topics are represented by some keywords as descriptors, and each document can also be represented by a set of keywords. Each document and topic keyword is, in turn, represented by an embedding vector. In this example, we have multiple levels of information, where similarities can be defined along the abstraction levels but also across different levels. Using a categorical colormap [12] to differentiate the topic groups would not allow for enough nuance to express similarities for the other levels.

Generally speaking, for text visualizations, we often want to depict the following relations using color: (1) the similarity across computed categories and groupings, e.g., topics or entities; (2) the similarity between keywords, n-grams, and word vectors; and (3) the similarity between a category and a keyword. An effective color assignment would allow for the investigation of these three types of relationships, facilitating the localization of individual attributes based on their color, as well as the comparison of different attributes.

To address this challenge, in this paper, we propose a guideline for semantic color mapping. It is described in the form of a pipeline that deduces best practices from several color assignment experiments and literature. We showcase the applicability of the proposed guideline in the context of text visualization applications in Sect. 4. The operationalization and in-depth evaluation of the pipeline is a subject of future work.

Contribution – We contribute a detailed description of the multistage process of semantic color mapping and a discussion of lessons learned, and best practices based on two application scenarios.

---

*Authors ordered alphabetically.

2 RELATED WORK

Color considerations for text can often be seen using the context of the case or task at hand in combination with semantic resonant colors of words. Semantic resonant colors map object words to their real-world colors, e.g., yellow for banana or blue for sky. Based on the assumption that meaningful colored words improve task solving performance, Lin et al. [21] conducted a study to investigate the hypothesis and propose colors for such context words. They found that semantic resonant color assignment significantly improves the chart reading time in their study. An example using semantic resonant colors for words or, in this case, categories is NEREx [9]. They use such resonant colors to support users in exploring entity graphs.

Not only are semantic resonant colors favorable in some situations, but other features are also essential for some situations without, e.g., meaningful resonant colors for words or in which the semantic resonant colors do not include the context of the task correctly [25]. Gramazio et al. [12] present various metrics with which a color scale can be generated, neglecting semantic resonant colors for words. But, their metrics could be further extended using such resonant colors to generate more meaningful color maps, including the context for words. Also, other considerations like the affection for colors are crucial properties for users [1]. Bartram et al. [1] demonstrate how small changes for color maps can steer the affection of colors for users into calm and positive. An extension of the technique with additional colors can lead to more focused semantic resonant colors.

Towards extending previous color scales, Steiger et al. [37] evaluate various 2D color maps for their usage as spatial color scales. They use various measures to investigate which properties the selected 2D color maps support for further analysis tasks in user applications. For example, Buchmüller et al. [3] use the previously discovered properties and further investigate the possibility of applying these color maps to fish swarms for collective behavior analysis. Primarily, 2D color maps with white or black in the center are unfavorable for such a task. When combining the previous considerations, we have to focus on the related work towards spatial color maps. Thus, such a finding is essential for 2D color maps in spatial tasks like those presented in this paper using 2D or 3D projections.

Sect. 3 presents all stages of our proposed pipeline in detail, incorporating further related work of each stage. Due to the scope of the paper, we do not include an extensive overview of color perception or color assignment research beyond the presented considerations and examples. However, our first introductory related work gives a few considerations for text and the use of colors for them. We identify a need for a semantic color mapping pipeline for 2D or 3D projections of words considering semantic resonant colors or spatial color maps to investigate the differences in the projections. The proposed pipeline in this work is further inspired by the arguments for enhancing the relative comparison of data objects using two-dimensional color assignments [37].

3 SEMANTIC COLOR MAPPING PIPELINE

In order to structure the design space, we present a pipeline encompassing the necessary steps to get from available data to semantic color labels. Fig. 1 shows the four distinct steps of the workflow we identified: (1) Aggregating the underlying data in a task-dependent way; (2) choosing a method for transforming the aggregated text data into numerical vector representations; (3) choosing an appropriate unit of analysis; and subsequently (4) applying a projection method. Finally, we can (5) apply the resulting coordinates onto a color map. At each step of the pipeline, we have a set of available design choices which influence the outcome. In the following section, we want to briefly highlight the key design considerations and possibilities.

Methodology – This paper aims to structure best practices and design consideration for color assignment in visual text analytics applications [18]. To come up with the proposed pipeline, we first identified a set of available methods from literature, as discussed in the previous section. Second, we analyzed the properties and characteristics of available methods. Lastly, based on our experience, extensive discussions, and literature reviews, we came up with the pipeline and collection of methods at each step. This paper points out task-specific considerations, which are further exemplified in application scenarios in Sect. 4.

Pipeline Dependencies – This section describes each step of the proposed pipeline independently. However, as the steps are not independent, we also provide considerations for the design as a guideline. That being said, further research and studies are needed to map out the best practices for design choices of pipeline steps.

3.1 Aggregation Level & Vector Representations

Text data can be analyzed in different granularity levels, depending on the use case at hand. The granularity levels typically include words/tokens, sentences, utterances, paragraphs, etc. On the token level, methods such as YAKE [6] allow the automatic extraction of important keywords from the textual data. Independent on the used level, its representation is usually numeric, captured in the form of a (potentially) high-dimensional vector. The reason is the ability to apply machine learning methods to such a numerical input. In the following, we showcase different forms of a vector representation that can be used to characterize text data.

Aggregation Level Considerations
Input: Raw or labeled data.
Output: Aggregated data.
Design Choices: Aggregation levels, e.g., token/word, sentence, paragraph, document
Dependencies: Analysis task

There are multiple ways of representing text through high-dimensional vectors [14]. We can use one of the early approaches that include **One-Hot Encoding Vectors** that encode word occurrence in a document and **Count Vectors** that encode word frequency. Different weighting techniques such as **Term Frequency – Inverse Document Frequency (TF-IDF)** [27] can be used to represent the word's importance in a document amongst a collection of documents. These early approaches represent words in a standalone manner though, i.e., the vectors do not encode any relationship between them. To represent word relationships, one can use more recent approaches that are built on a different learning assumption, i.e., words that occur in the same context are similar and thus should be represented through similar vectors. This principle is used in, e.g., **Co-Occurrence Vectors** [4, 5] and neural network based learning techniques such as **Word2Vec** [24] or **ConceptNet** [34]. The latter produces static word embeddings capturing the words' meaning, but lacks the ability to distinguish between polysemous words (words with multiple meanings).

The most recent advances in NLP are deep-learning-based language models (e.g., BERT [7]) that produce **contextualized word embeddings**, i.e., a unique embedding vector for each word's occurrence in the context. Before using contextualized word embeddings in a text application, we first need to make several analysis-related decisions. Language model architectures typically consist of multiple layers. Since the information that gets captured in the different layers varies [29,33], one has to decide which layer is appropriate for a given use case. We can use embedding vectors from one specific layer or combine embeddings from multiple layers by averaging or concatenating them. For some use cases, one can also extract embeddings from context-size zero, which sometimes are used as

replacements for static embeddings [19]. The list of possible vector representations does not end with the development of language models, though.

Depending on the application scenario, one can create further representation encodings, e.g., **topic distributions** [10] as well as diverse **scoring techniques** that capture text characteristics, or use **WordNet embeddings** that are built from semantic networks [30]. For other granularity levels, we might need to either average word level vectors or come up with new alternatives. For instance, to represent sentences, we can use neural networks that produce **sentence embeddings** [28].

Vector Representations Considerations

Input: Aggregated text data

Output: High-dimensional vectors

Design Choices: Traditional document vectors, static word embeddings, contextualized word embeddings, sentence embeddings, topic distributions, WordNet

Dependencies: Aggregation level

3.2 Unit of Analysis & Projection Methods

The following step is the translation from potentially high-dimensional vector representations into low-dimensional 2D or 3D coordinates. Besides choosing the actual dimensionality reduction method, we also have to consider the task-dependent unit of analysis. Based on our intended task, we need to apply a secondary aggregation step to combine vector representations into topics, concepts, or sentiments. For example, we might average the vectors of multiple tokens belonging to a concept to get a shared score representation for the whole concept. Alternatively, we may aggregate the 2D/3D coordinates after the dimensionality-reduction step. Later in this section, we discuss that the second approach is beneficiary if we have a growing data corpus.

Unit of Analysis Considerations

Input: High-dimensional vectors

Output: Aggregated sets of vector data

Design Choices: Aggregation based on chosen unit of interest, e.g. topic

Dependencies: Aggregation level, task

To translate the high-dimensional vectors into low-dimensional 1D/2D or 3D coordinates, we have the choice between a wide range of available projection techniques [11, 22]. Methods can be classified based on two main characteristics: *linearity* and *preservation of neighborhood*. Linear methods are simple, computationally efficient, and easy to interpret, but cannot capture distributions in complex higher-dimensional manifolds. Non-linear ones require more careful optimization of parameters, but work better for complex manifolds.

Depending on the output of the previous step, one also has to consider that different projection methods are applicable to different *input types*: Either high-dimensional samples, or an available distance metric. One might therefore choose a method based on the availability of respective input data from the previous step. Many methods can be used with high-dimensional input samples, but well-known approaches like **MDS**, **t-SNE**, and **UMAP** only require an available distance metric [11].

A prime example of a linear and global method is **Principal Component Analysis (PCA)**. **MDS** is the most well-known global and non-linear method. As the term suggests, the advantage of both methods is that they preserve global structure at the possible expense of local discriminability. In particular, one might expect good global coverage of the space, but similar data points might be projected onto very similar coordinates, and might therefore not be easily distinguishable from each other. Methods like **t-SNE**, **UMAP** or **PBC** are part of the non-linear and local methods. These methods are better at untangling local structures at the expense of global structural fidelity. This trade-off between global consistency and local discriminability might depend on the particular task. A recent surveys finds that **PBC**, **t-SNE**, **UMAP** and **IDMAP** produce consistently good results [11], both quantitatively and in terms of human perception, for 2D projections. Another survey found that **t-SNE**, **UMAP** and **neural auto encoder**-based approaches produce the best results for 3D projections [38].

When the tasks demand adding additional data points over time, e.g., for progressive topic modeling [10], we want to ensure continued stability of the 2D projection, which requires that we do not recompute the whole embedding, but instead embed additional data points into an existing embedding. Especially for non-linear embedding and projection methods, averaging vector representations can lead to larger jumps in the projected coordinates. Therefore, it is preferable to average the final low-dimensional projected coordinates instead. This ensures higher stability of the resulting coordinates and subsequent colors.

As a final post-processing step, we can apply a linear transformation like stretching along coordinate axes to fit the resulting projection to the shape of the chosen color space, e.g., a 2D rectangular color map, or a three-dimensional complex color space.

Projection Method Considerations

Input: Set of high-dimensional numeric vector representations

Output: 2D/3D coordinate per input

Design Choices: Local vs. global, linear vs. non-linear embedding methods, algorithm hyperparameters

Dependencies: Vector representations, color space

3.3 Color Mapping

In the next step, a color map is created, so that each of the projected 2D points is matched with a color. Zhou et al. give a general overview of color spaces in visualization [42]. Compared to the traditional use case of color maps in, e.g., scatter plots [41], the color in a semantic color map does not encode an external attribute such as frequency or importance. Instead, it only represents the similarity of points in the projection space, usually using the euclidean distance.

Choosing an appropriate color space is essential for extracting a good semantic color map from it. Many different color spaces, such as RGB, HSV, CIELab can be used to represent color values. For a semantic mapping, the colors of the color space should represent the similarity between projection points, meaning that close-by projection points are represented using colors that are perceived as similar. To be able to distinguish points that are far away, the space needs to contain many distinguishable colors - therefore the extracted 2D map should cover many different hues, and not be restricted to, e.g., only purple and greens.

Once an appropriate color space is found, we still need to create a mapping from the color space to the projection space. The easiest way to create such a map is to take a slice of the higher dimensional color space to create a fixed 2D color map. Steiger et al. offer a visual tool for the exploration of different precomputed 2D color maps [37]. They introduce multiple quantitative measures, e.g., perceived color distance, and color map properties such as perceptual linearity and number of distinguishable colors. Bremm et al. employ
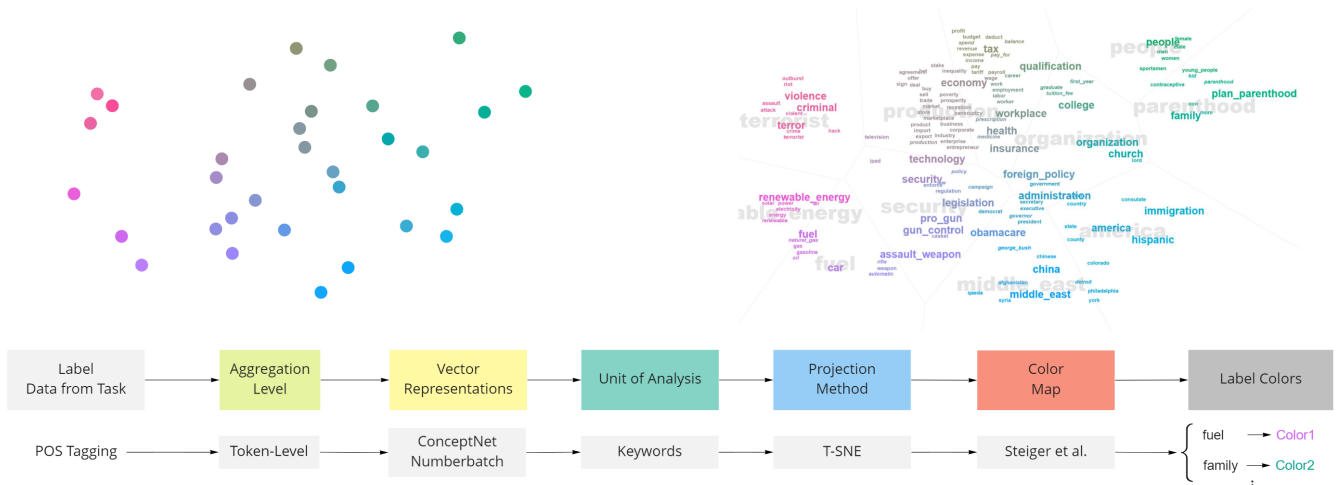
Figure 2: Semantic Concept Space Application [8]. On the left, we display the concept projection in the color space; on the right, the semantic concept space visualization, showing various concepts and their topics in a unified space.

a 2D semantic color map to compare descriptors, and base it on a slice of the RGB color cube, as their goal was to have high-contrast colors [2]. Steiger et al. use four colors as corner points and interpolate between them to create a 2D color map [36]. Each corner of the 2D map is assigned a color, equalized across intensity and saturation, and all other positions are interpolated using the CIELab color space. We will present two 2D example applications using semantic color maps in Section 3.

One big challenge for creating these semantic color maps is that simply taking a slice of a 3D color space cube might result in areas where the saturation is very low or uneven. Further, different color hues at the same saturation level are perceived differently by humans, as observed by Steiger et al. [37]. The outline of the 2D projection does not necessarily have to be a rectangle or ellipse - it could also contain empty regions without points. To keep regions of the projection space as separated as possible, it might make sense to sample the color map to match the distribution of the projection. Adaptive sampling of the color space, as proposed by Koutrouli et al. [16] or Nardini et al. [26], allows for more evenly perceived saturation levels on most positions on the 2D map. It also helps to maintain small but well-distinguishable regions on the map, with smooth transitions between regions. Adaptive color maps can also support users that might be afflicted with vision-based deficiencies, as suggested by Waldin et al. [40].

Color Map Considerations
Input: 2D/3D coordinate per input
Output: Color per coordinate
Design Choices: Color space (e.g., RGB, HSV, CIELab), color perception (e.g., proximity, cultural meaning), space coverage
Dependencies: Coordinate positions, task

4 APPLICATION SCENARIOS

In the following, we present two application examples that utilize the proposed semantic color mapping pipeline.

4.1 Semantic Concept Spaces

In *Semantic Concept Spaces* [8], we proposed a tool to visualize concepts and their topics in a unified space. The goal was to create a semantic concept space visualization using 2D projections of word embeddings. Words from a given corpus were filtered based on their Part-Of-Speech tags, enriched with semantically similar words, on the token level. We extracted vector embeddings for all tokens using ConceptNet [34]. All collected words were then projected to 2D using t-SNE. As we also wanted to visualize the relation between topics and their descriptive keywords, topics were represented using the embeddings of a selection of the topics' important keywords, by aggregating the t-SNE positions to a single position for each topic. Using an agglomerate hierarchical clustering, words were aggregated based on their 2D position and high-dimensional similarity, and assigned to a concept hierarchy of three ranks. Words on the highest rank were represented using gray color, and used to create Voronoi cells that indicate global groups of concepts. Each concept of the middle rank was assigned a color, based on an underlying 2D semantic color map Fig. 2. We chose a slice of the CIELab color space, at a luminosity of 60 percent, similar to Steiger et al. [37]. Words in the lower ranks (keywords) inherited the color of their parent concept. On one hand, this allowed users to quickly see which concepts are similar to each other, as close concepts have similar colors. On the other hand, keywords might have another color than their direct neighbors, showing then a mismatch between the high-dimensional similarity of words and their closeness according to the 2D projection. The concept space showing all three hierarchy ranks can be observed in Fig. 2.

4.2 Topic Model Icicle

We implemented a visual analytics system for continuous topic model refinement in a streaming system [35]. The main visualization is shown in Fig. 3. To determine topic colors, we start at the token level, extracting the top 20 keywords using YAKE [6]. We chose to operate on the token-level as topics are typically evaluated through their descriptive keywords–hence, we wanted keyword changes to be reflected in the topic's color. We then extract DistilBert [31] embeddings for the descriptor keywords, and project them onto the color map by Bremm et al. [2] using UMAP. To determine the final topic color, we compute the weighted average position of the topic's keywords, using the YAKE keyword scores as weights. We aggregate the UMAP positions rather than the DistilBert vectors to ensure that the final result is within the bounds of the previously computed UMAP projection. To optimize the color assignment, we remove frequent, domain-specific keywords with little semantic information, which has two benefits. First, it reduces potential distortion in the UMAP projection by reducing the number of data points. Second, removing semantically uninteresting words means that the limited number of available colors can be better distributed across the available topics. First, it ensures that keywords with more descriptive content can be projected with less distortion. Second,

Figure 3: Topic Model Refinement Application [35]. On the left, we display the keyword distribution in the color space; on the right, the sankey diagram showing topic development through color changes.

it keeps these keywords from taking up room in the color map and leaves more colors available for other keywords and concepts.

5 DISCUSSION

Based on our experience utilizing semantic color mapping in the presented applications, in this section, we discuss important design considerations and best practices.

5.1 Design Considerations

We discuss three design considerations for the pipeline.

Task Support – The main consideration for using the proposed pipeline is to figure out which tasks can be supported by using such color mapping. In particular, which unit of analysis will be used to anchor the color assignment and how is the data distributed for this analysis level, i.e., are there distinct clusters, etc.

Unit of Analysis Distribution – One of the decisive criteria for the effectiveness of the color assignment is the distribution of the attributes that are mapped to color. If the attributes do not equally cover distinct areas in the color space, then the resulting visualization will not make full use of the available colors. In particular, if the attributes are mainly clustered in the middle of the space, they might end up being assigned a non-expressive color, e.g., gray or white. On the other hand, having balanced clustered throughout the space can result in meaningful color mapping, which, in turn, could be the basis of a discrete or categorical color assignment.

Subjective Similarity – Color distances and similarities are subjective, culture-dependent, and perception-dependent [15]. Considering for whom, where, and in which context a visualization is designed, is crucial for the choice of color space. In some languages and cultures, certain color shades can not be distinguished or named as easily as in others [15], influencing the detection of differences in color assignment. Taking such issues into account will determine the inclusivity, accessibility, and effectiveness of the visual design.

5.2 Best Practices

We discuss three best practices based on our application experiences.

Projection and Color Space Match – To make use of the full spectrum of the color space, the projection (shape) has to match the color space (shape). If they do not align, some areas of the color space might not be covered by any data points, such as in Fig. 3. To address such issues, rotating and scaling the projection or the color space to align them could allow for a better use of available colors.

Incremental or Streaming Analysis – In many applications, objects could be progressively added to the visualization. Thus, visualization designers need to consider how adding objects to the projection space would skew the vectorization and color mapping. For example, the new vectors might be projected outside the color space, or they might end up cluttered in one area, e.g., in the middle of the color space. Recalculating the last steps of the semantic color mapping pipeline would solve this issue, but cause a discontinuation to the existing colors in the visualization. Allowing for manually triggering color reassignments could avoid issues resulting from continuously adding objects to the color space.

Color Perception – One of the most important issues to consider when using this pipeline is that some objects might be mapped close to each other in the space, resulting in color similarities below the *just-noticeable-difference*. In addition, due to perception issues, such as color blindness and others, users might face difficulties distinguishing areas in the used color space. Allowing users to choose from a list of colormaps would empower them to pick the best perspective based on their understanding and perception.

6 CONCLUSION

In this paper, we have presented a pipeline for semantic color mapping, as a guideline for meaningful color assignment in visual text analytics applications. Our proposed approach is based on a multi-step process that starts with a task-based extraction of labels from the text data; followed by the determination of the aggregation level to generate vector representations; afterward, we define the unit of analysis to aggregate and project the vectors; finally, the projection is used for a color assignment based on given color space. In addition, this paper presented two application scenarios that utilize the proposed pipeline. Finally, we concluded with a discussion of design considerations and best practices to guide users.

In the future, we plan to operationalize the proposed pipeline by implementing it as a service or library to support practitioners to implement our identified considerations in their works. In addition, we aim to study the design considerations and tradeoffs of users directly in a user study. The study empirically can show the influence of semantic resonant colors for words against the spatial differences of 2D color maps. Thus, such a study can provide guidance for text visualization designers based on empirical results. These results can be further investigated and improve the design considerations and best practices for the proposed pipeline with additional operationalization in the to-be-developed service or library.

REFERENCES

[1] L. Bartram, A. Patra, and M. Stone. Affective color in visualization. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pp. 1364–1374, 2017.

[2] S. Bremm, T. v. Landesberger, J. Bernard, and T. Schreck. Assisted Descriptor Selection Based on Visual Comparative Data Analysis. *Computer Graphics Forum*, 30(3):891–900, 2011. doi: 10.1111/j.1467 -8659.2011.01938.x

[3] J. F. Buchmüller, U. Schlegel, E. Cakmak, E. Dimara, and D. A. Keim. Spatialrugs: Enhancing spatial awareness of movement in dense pixel visualizations. *ArXiv*, abs/2003.12282, 2020.

[4] J. A. Bullinaria and J. P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526, 2007.

[5] J. A. Bullinaria and J. P. Levy. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. *Behavior research methods*, 44(3):890–907, 2012.

[6] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020. doi: 10.1016/j.ins.2019.09.013

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] M. El-Assady, R. Kehlbeck, C. Collins, D. A. Keim, and O. Deussen. Semantic concept spaces: Guided topic model refinement using word-embedding projections. *IEEE Trans. Vis. Comput. Graph.*, 26(1):1001–1011, 2020. doi: 10.1109/TVCG.2019.2934654

[9] M. El-Assady, R. Sevastjanova, B. Gipp, D. Keim, and C. Collins. Nerex: Named-entity relationship exploration in multi-party conversations. In *Computer Graphics Forum*, vol. 36, pp. 213–225. Wiley Online Library, 2017.

[10] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins. Progressive learning of topic modeling parameters: A visual analytics framework. *IEEE transactions on visualization and computer graphics*, 24(1):382–391, 2017.

[11] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea. Toward a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2153–2173, 2021. doi: 10.1109/TVCG.2019.2944182

[12] C. Gramazio, D. H. Laidlaw, and K. B. Schloss. Colorgorical: Creating discriminable and preferable color palettes for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23:521–530, 2017.

[13] C. G. Healey. Choosing effective colours for data visualization. In *Proceedings of IEEE Visualization*, pp. 263–270. IEEE, 1996.

[14] Q. Jiao and S. Zhang. A brief survey of word embedding and its recent development. In *IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, pp. 1697–1701, 2021. doi: 10.1109/IAEAC50856.2021.9390956

[15] C. Kim, U. Hinrichs, S. M. Mohammad, and C. Collins. Lexichrome: Text construction and lexical discovery with word-color associations using interactive visualization. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pp. 477–488, 2020.

[16] M. Koutrouli, J. H. Morris, and L. J. Jensen. U-cie [/ju: 'si:/]: Color encoding of high-dimensional data. *bioRxiv*, 2022. doi: 10.1101/2021. 12.02.470966

[17] K. Kucher and A. Kerren. Text visualization browser : A visual survey of text visualization techniques. 2014.

[18] K. Kucher and A. Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2015.

[19] A. Lauscher, T. Lueken, and G. Glavaš. Sustainable modular debiasing of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4782–4797. Association for Computational Linguistics, Punta Cana, Dominican Republic, Nov. 2021.

[20] S. Lin, J. Fortuna, C. Kulkarni, M. Stone, and J. Heer. Selecting semantically-resonant colors for data visualization. *Computer Graphics Forum (Proc. EuroVis)*, 2013.

[21] S. Lin, J. Fortuna, C. Kulkarni, M. Stone, and J. Heer. Selecting semantically-resonant colors for data visualization. In *Computer Graphics Forum*, vol. 32, pp. 401–410. Wiley Online Library, 2013.

[22] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, 2017. doi: 10.1109/TVCG.2016.2640960

[23] S. Liu, X. Wang, C. Collins, W. Dou, F. Ouyang, M. El-Assady, L. Jiang, and D. A. Keim. Bridging text visualization and mining: A task-driven survey. *IEEE transactions on visualization and computer graphics*, 25(7):2482–2504, 2018.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, eds., *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*, 2013.

[25] K. Mukherjee, B. Yin, B. E. Sherman, L. Lessard, and K. B. Schloss. Context matters: A theory of semantic discriminability for perceptual encoding systems. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):697–706, 2021.

[26] P. Nardini, M. Chen, F. Samsel, R. Bujack, M. Böttinger, and G. Scheuermann. The making of continuous colormaps. *IEEE Trans. Vis. Comput. Graph.*, 27(6):3048–3063, 2021. doi: 10.1109/TVCG. 2019.2961674

[27] J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, vol. 242, pp. 29–48. Citeseer, 2003.

[28] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[29] A. Rogers, O. Kovaleva, and A. Rumshisky. A Primer in BERTology: What We Know About How BERT Works. *Trans. of the Association for Computational Linguistics*, 8:842–866, 2020.

[30] C. Saedi, A. Branco, J. António Rodrigues, and J. Silva. WordNet embeddings. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pp. 122–131. Association for Computational Linguistics, Melbourne, Australia, July 2018. doi: 10.18653/v1/W18-3016

[31] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv e-prints*, pp. 1–5. arXiv:1910.01108, 2020.

[32] V. Setlur and M. C. Stone. A linguistic approach to categorical color assignment for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):698–707, 2016. doi: 10.1109/TVCG. 2015.2467471

[33] R. Sevastjanova, A.-L. Kalouli, C. Beck, H. Hauptmann, and M. El-Assady. LMFingerprints: Visual Explanations of Language Model Embedding Spaces through Layerwise Contextualization Scores. In *Eurographics Conf. on Visualization (EuroVis)*, 2022 (accepted).

[34] R. Speer, J. Chin, and C. Havasi. ConceptNet 5.5: An open multilingual graph of general knowledge. pp. 4444–4451, 2017.

[35] F. Sperrle, H. Schäfer, D. Keim, and M. El-Assady. Learning Contextualized User Preferences for Co-Adaptive Guidance in Mixed-Initiative Topic Model Refinement. *Computer Graphics Forum*, 40(3):215–226, 2021. doi: 10.1111/cgf.14301

[36] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. A. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. *Comput. Graph. Forum*, 33(3):401–410, 2014. doi: 10.1111/cgf.12396

[37] M. Steiger, J. Bernard, S. Thum, S. Mittelstädt, M. Hutter, D. A. Keim, and J. Kohlhammer. Explorative analysis of 2 d color maps. 2015.

[38] Z. Tian, X. Zhai, G. van Steenpaal, L. Yu, E. Dimara, M. Espadoto, and A. Telea. Quantitative and qualitative comparison of 2d and 3d projection techniques for high-dimensional data. *Information*, 12(6), 2021. doi: 10.3390/info12060239

[39] N. Umezu and E. Takahashi. Visualizing color term differences based on images from the web. *Journal of Computational Design and Engineering*, 4(1):37–45, 2017.

[40] N. Waldin, M. Bernhard, P. Rautek, and I. Viola. Personalized 2d color maps. *Comput. Graph.*, 59:143–150, 2016. doi: 10.1016/j.cag.2016. 06.004

[41] Y. Wang, X. Chen, T. Ge, C. Bao, M. Sedlmair, C. Fu, O. Deussen, and B. Chen. Optimizing color assignment for perception of class separability in multiclass scatterplots. *IEEE Trans. Vis. Comput. Graph.*, 25(1):820–829, 2019. doi: 10.1109/TVCG.2018.2864912

[42] L. Zhou and C. D. Hansen. A survey of colormaps in visualization. *IEEE Trans. Vis. Comput. Graph.*, 22(8):2051–2069, 2016. doi: 10.1109/TVCG.2015.2489649