Supplemental Material for Visualizing Data using t-SNE

Laurens van der Maaten LVDMAATEN@GMAIL.COM
TiCC
Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands

Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
6 King's College Road, M5S 3G4 Toronto, ON, Canada

1. Experiments

In this supplementary material, we present the results of our experiments that compare the visualizations produced by t-SNE with those produced by seven other dimensionality reduction techniques on five datasets from a variety of domains. Some of these results were already presented in the paper, however, we present the results here in a different form.
The five datasets we employed in our experiments are described in subsection 1.1. Subsection 1.2 presents our experimental setup and the results are presented in subsection 1.3.

1.1 Datasets

The five datasets we employed in our experiments are: (1) the MNIST dataset, (2) the Olivetti faces dataset, (3) the COIL-20 dataset, (4) the word-features dataset, and (5) the Netflix dataset. In the paper, we already presented some results on the MNIST dataset, the Olivetti faces dataset, and the COIL-20 datasets. We first describe the two datasets that were not used in the main paper.
The word-features dataset (Mnih and Hinton (2007)) consists of 100-dimensional real-valued feature vectors for the 1,000 most common words in corpus of news articles from the period 1994-1996. The feature vectors were learned by trying to make the identity of the next word be as predictable as possible from the identities of the previous five words when the predictions are made using the feature vectors. The Netflix dataset[1] contains ratings of 17,770 movies originating from over 400,000 movie viewers. A Restricted Boltzmann Machine with 30 hidden units was trained on these ratings (see Salakhutdinov et al. (2007) for details of the training), yielding a dataset of 30-dimensional movie-specific features. We show visualizations for the 500 most popular movies.

1.2 Experimental setup

The setup of our experiments consists of two main stages. First, we denoise the data using PCA, reducing the dimensionality of the data to 30. Subsequently, we use one of the eight dimensionality reduction techniques in order to reduce the dimensionality of the data to two dimensions (for the MNIST, COIL-20, and Olivetti faces datasets) or three dimensions (for the word-features and Netflix datasets). We construct the resulting low-dimensional data representation by plotting the

1. The Netflix dataset is publicly available from `http://www.netflixprize.com`.

original data on the low-dimensional coordinates that result from the dimensionality reduction. For the word-features and Netflix datasets, the third dimension is visualized using a color encoding. The visualizations for the word-features and Netflix can best be viewed on a computer screen, which allows the reader to zoom in on the visualizations.
The parameter settings we employed in our experiments are listed in Table 1. In the table, $Perp$ represents the perplexity of a Gaussian kernel, i represents the maximum number of iterations, k represents the number of nearest neighbors employed in a neighborhood graph, λ represents the initial neighborhood radius that is used in CCA, and s represents the sample variance of the data.
The details on the minimization of the t-SNE cost function are discussed in the paper. In the optimization of the objective function of SNE, we use a momentum term of 0.5 for the first 1,500 iterations, and use a momentum term of 0.8 for the remaining 2,500 iterations. Furthermore, in the optimization of the SNE objective function, we initially add Gaussian noise with a variance of 0.3 after each iteration. The variance of the noise is reduced slowly by multiplying it by 0.99 after each iteration. In the experiments with Isomap, MVU, LLE, and Laplacian Eigenmaps, we only visualize datapoints that are vertices in the largest connected component of the neighborhood graph[2]. For computational reasons, in our experiments with MVU, we employ the FastMVU implementation that is described by Weinberger et al. (2007).

Technique	*Cost function parameters*
t-SNE	$Perp = 40$
Sammon mapping	none
CCA	$\lambda = 3s^2$
SNE	$Perp = 40$
Isomap	$k = 12$
MVU	$k = 12$
LLE	$k = 12$
Laplacian Eigenmaps	$k = 12 \quad \sigma = 1$

Table 1: Parameter settings for the experiments.

1.3 Results

In this subsection, we present the results of our experiments on the five datasets discussed in subsection 1.1. To keep the comparison fair, we did not employ the random-walk version of t-SNE discussed in the main paper. The random-walk version can be expected to produce better results on large datasets such as the MNIST and the Netflix datasets.
Below, we discuss the results of the experiments on each of the five datasets separately.

1) MNIST dataset. Figures 1 to 8 present our results on the MNIST dataset. t-SNE is much better than the other techniques at visualizing the natural classes in the data. In the visualization constructed by Sammon mapping, only the classes zero and one are fairly well separated from the other classes. CCA performs better than Sammon mapping in identifying the different classes, how-

2. The reader should note that by definition, Isomap, MVU, LLE, and Laplacian Eigenmaps can only reduce the dimensionality of data that gives rise to a neighborhood graph that is connected.

ever, the classes are still severely mixed up. In particular, the classes four, five, eight, and nine can be found anywhere in the visualization. In the visualization produced by SNE, some of the digit classes are fairly well separated from the other ones (in particular, classes zero, one, two, six, and seven). The classes four and nine, and the classes three, five, and eight are not separated in the SNE visualization. In contrast, in the t-SNE visualization, all classes are very clearly separated. Even the so-called 'continental' sevens form a small separate cluster. Moreover, t-SNE reveals much of the local structure of the data, such as the orientation of the ones, fours, sevens, and nines, as well as the 'curliness' of the twos.

2) Olivetti dataset. Figures 9 to 16 present our results experiments on the Olivetti faces dataset. t-SNE outperforms the other visualization techniques, in that it clearly constructs clusters of images belonging to the same face. The visualizations constructed by Sammon mapping and CCA also reveal some of this structure, but these visualizations do not contain any separation between dissimilar faces. The quality of the visualizations constructed by Isomap, MVU, Laplacian Eigenmaps, and LLE is clearly lower than that of t-SNE.

3) COIL-20 dataset. Our results on the COIL-20 dataset are presented in Figures 17 to 24. t-SNE clearly visualizes the rotation manifolds that constitute the COIL-20 dataset, and constructs a large separation between the rotation manifolds for most of the object classes. In addition, t-SNE aligns the rotation manifolds of the four toy car classes in the dataset because images of different cars in the same orientation are more similar than images of the same car in different orientations. Some of the rotation manifolds in the t-SNE visualization are not 'closed', suggesting that the gradient descent did not converge to the global minimum of the objective function. The failure of t-SNE to close some of the rotation manifolds may be due to the presence of the other object classes that 'block' such a closure. SNE also performs well on the COIL-20 dataset.
The visualizations produced by the other techniques are disappointing, although some of the rotation manifolds can be observed in the visualization produced by Sammon mapping and CCA. Techniques that are based on neighborhood graphs (such as Isomap, LLE, and MVU) do poorly on the COIL-20 dataset, because they are incapable of dealing with datasets that consist of widely separated submanifolds. These techniques could, of course, be applied separately to each connected subset of the neighborhood graph, but this would lose information about the relationships between the manifolds in different subsets.

4) Word-features dataset. Figures 25 to 32 present our results on the word-features dataset. For this dataset we used three-dimensional visualizations. The information in the first two dimensions is represented by the spatial locations of the words, whereas the information in the third dimension is represented using the color spectrum. Hence, two words are similar when they have *both* a small pairwise distance and a similar colour.
t-SNE produces many clusters of words that are semantically related. For instance, t-SNE visualizes names and abbreviations of months close together: *january - jan - feb - february - march - april - may - june - july - november - december*. Other identified semantic clusters entail forms of the verb *be* such as *'m - 're - are - is - was- were - been - being - would be*, human roles or professions such as *victims - refugees - prosecutors - lawyers - investigators - officers - doctors - voters - parents - friends - supporters -rebels - leaders - members - officials - official - sources*, verbs related to planning such as *agreed - continued - planned - scheduled*, and words related to time such as *a.m.* -

p.m. and *tuesday - wednesday - friday - saturday - sunday*. SNE has some good clusters around the edges of the map but suffers from the crowding effect in the center. The visualizations produced by the other techniques reveal much less semantic structure.

5) Netflix dataset. Our results on the Netflix dataset are presented in Figures 33 to 40. Although it is difficult to assess the quality of such visualizations, the visualization produced by t-SNE again reveals a lot of structure in the data. For instance, the three *Lord of the Rings* movies are tightly clustered together (the yellow cluster on the left). Right of this cluster there is a tight orange cluster that contains (some of) the *Harry Potter* movies. In the bottom of the visualization Micheal Moore's documentaries *Bowling for Columbine* and *Fahrenheit 9/11* are close together. In the right part of the visualization, there is, e.g., a cluster of *Austin Powers* movies. Furthermore, movies and corresponding follow-up movies are visualized close together or even on top of each other. Examples of such movies are *Kill Bill Vol. 1 - Kill Bill Vol. 2*, *Rush Hour - Rush Hour 2*, and *Bad Boys - Bad Boys 2*. Connoisseurs will also be able to identify groups of movies that represent a single a genre, such as a group of action movies in the top of the visualization (in orange/red). The other visualization techniques also reveal some of these structures, but not as well as t-SNE.

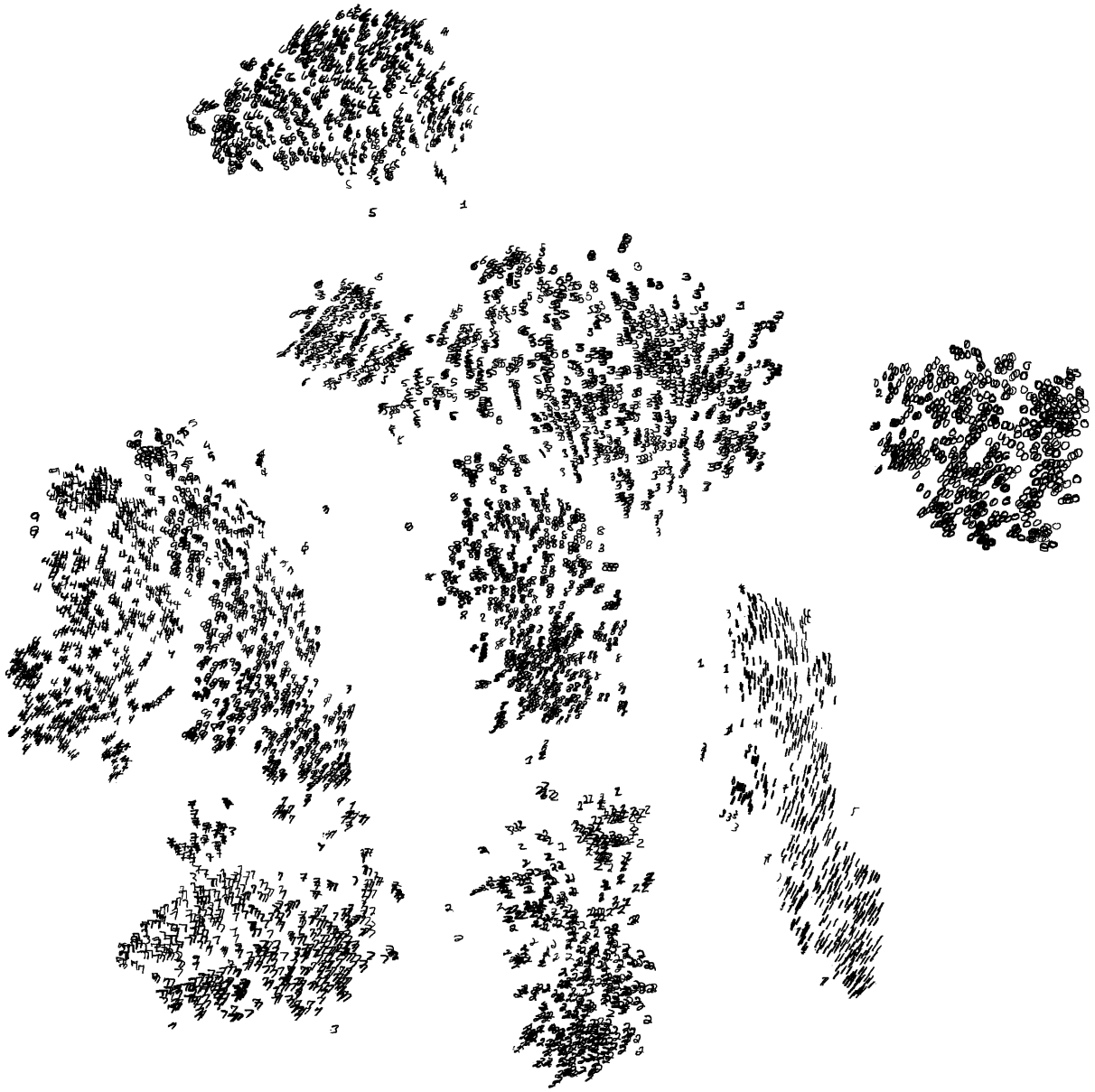

Figure 1: Visualization of 6,000 digits from the MNIST dataset produced by t-SNE.

Figure 2: Visualization of 6,000 digits from the MNIST dataset produced by Sammon mapping.

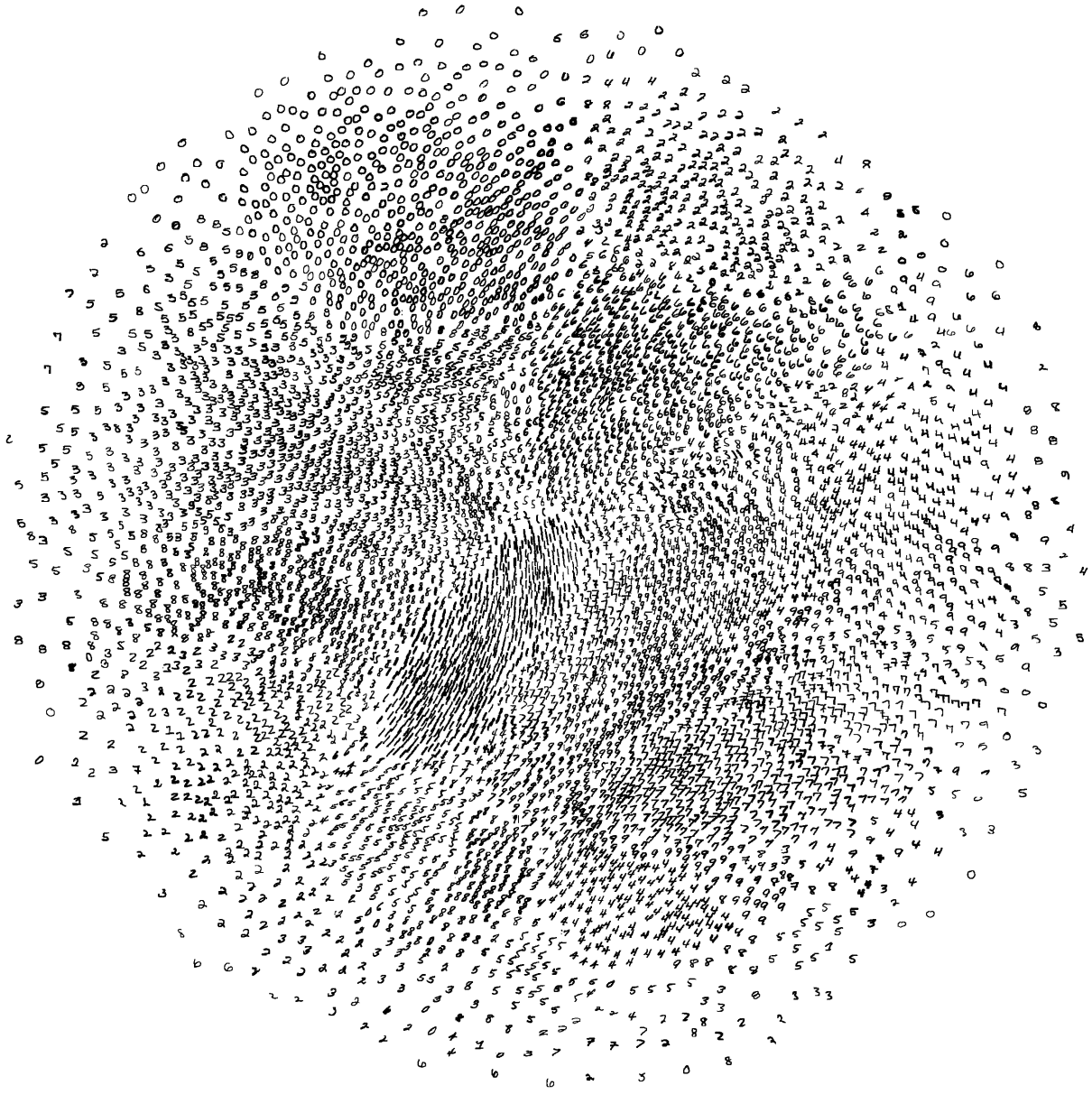

Figure 3: Visualization of 6,000 digits from the MNIST dataset produced by CCA.

Figure 4: Visualization of 6,000 digits from the MNIST dataset produced by SNE.

Figure 5: Visualization of 6,000 digits from the MNIST dataset produced by Isomap.

Figure 6: Visualization of 6,000 digits from the MNIST dataset produced by MVU.

Figure 7: Visualization of 6,000 digits from the MNIST dataset produced by LLE.

Figure 8: Visualization of 6,000 digits from the MNIST dataset produced by Laplacian Eigenmaps.

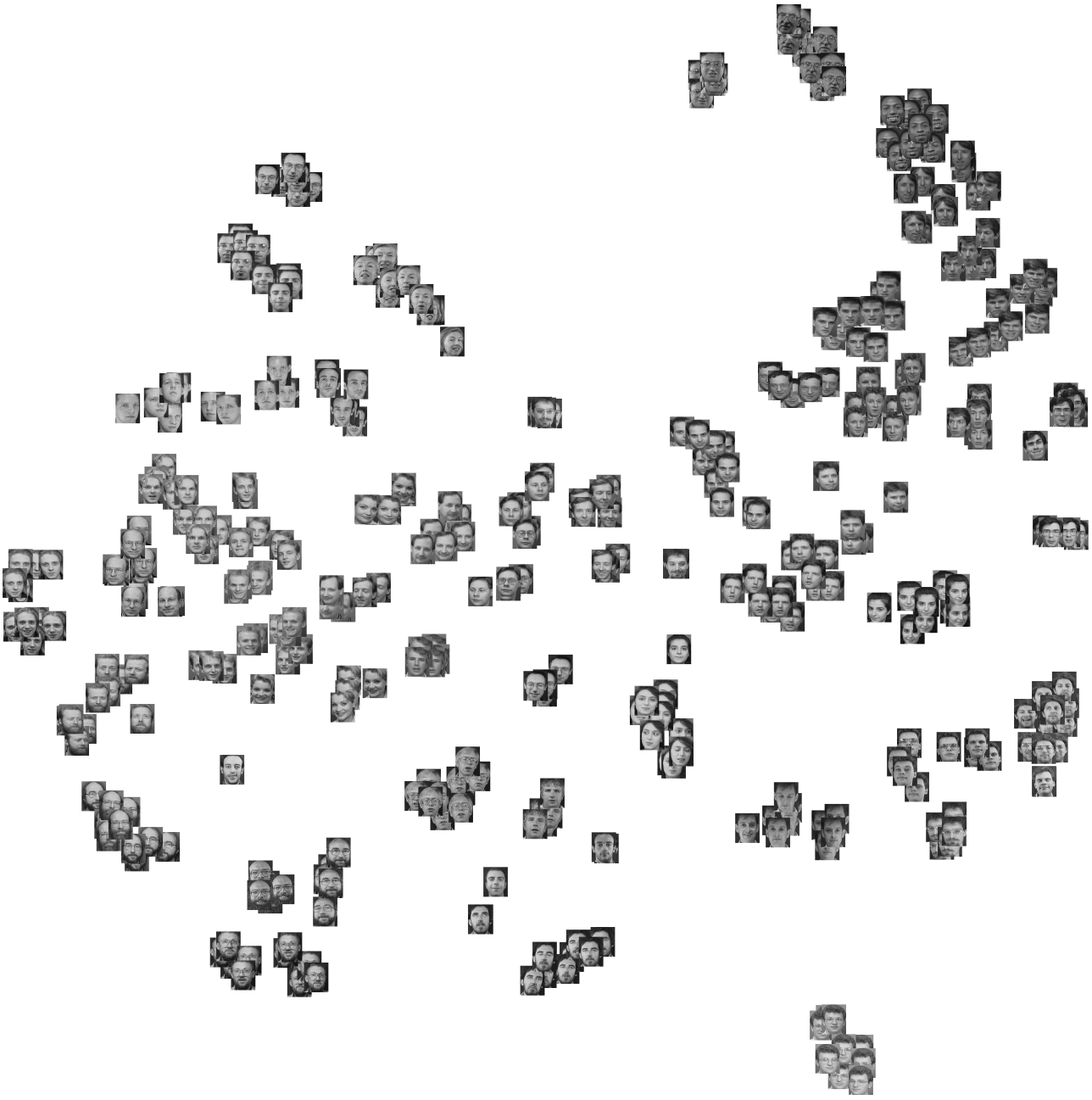

Figure 9: Visualization of the Olivetti faces dataset produced by t-SNE.

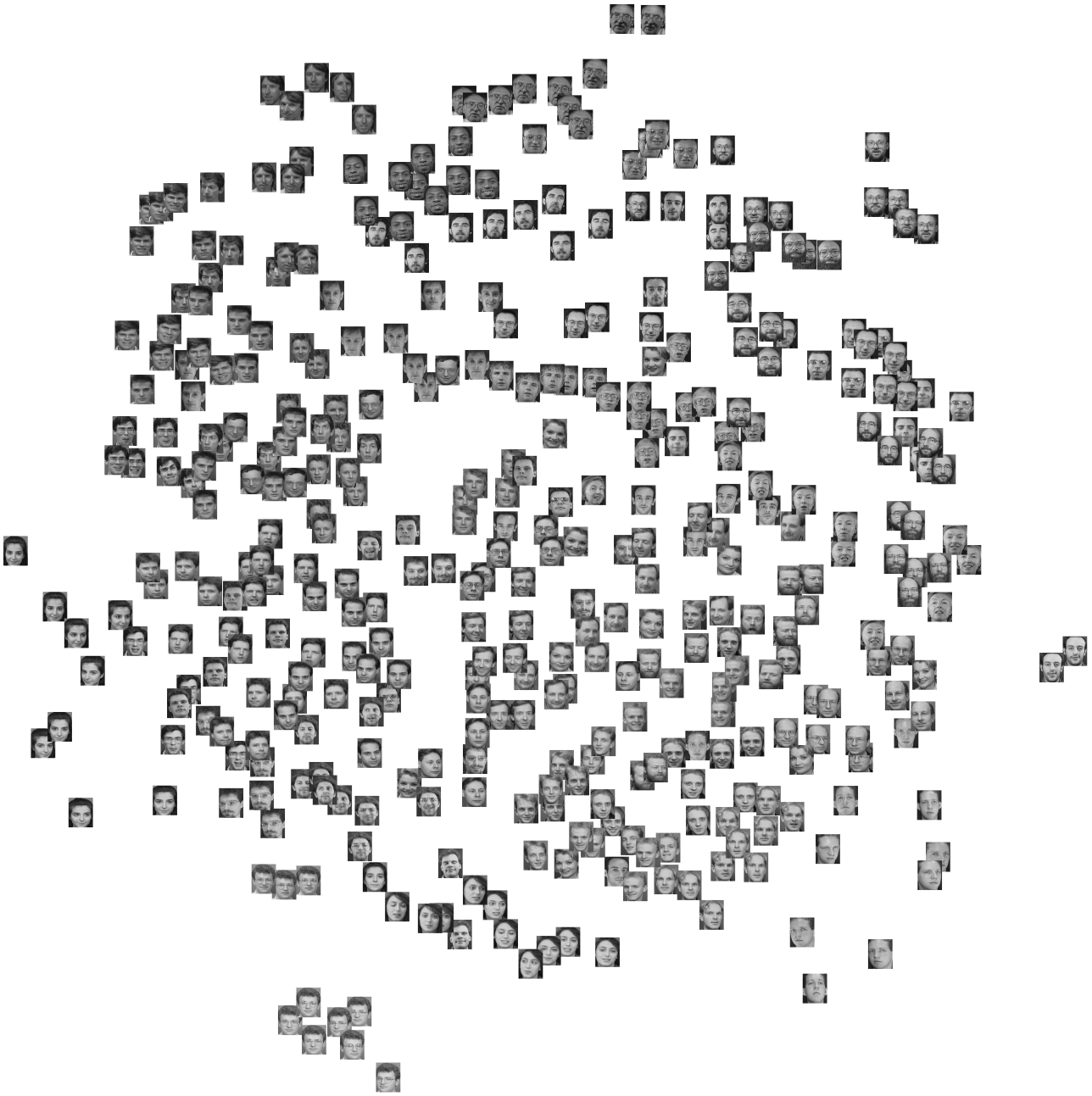

Figure 10: Visualization of the Olivetti faces dataset produced by Sammon mapping.

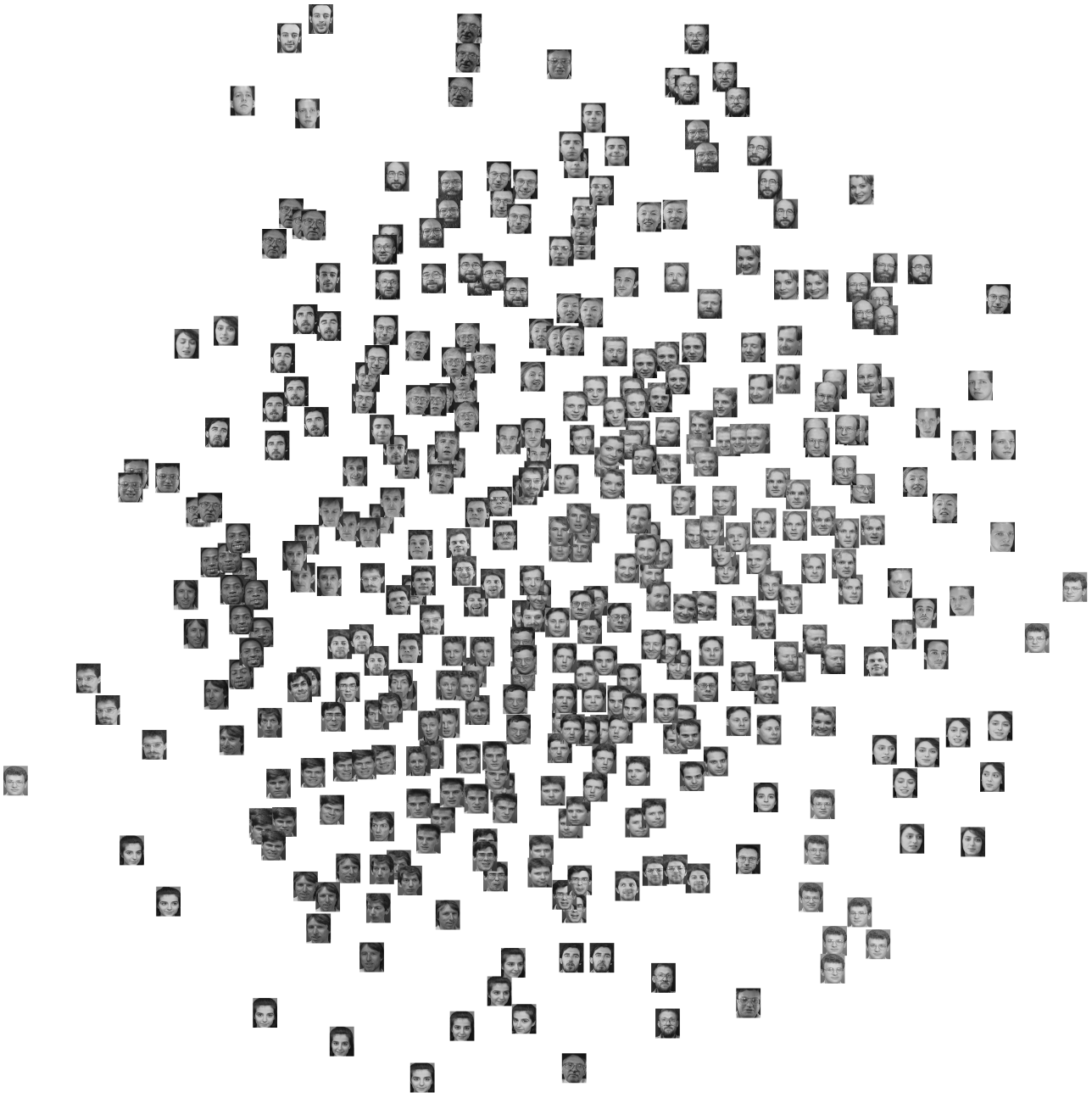

Figure 11: Visualization of the Olivetti faces dataset produced by CCA.

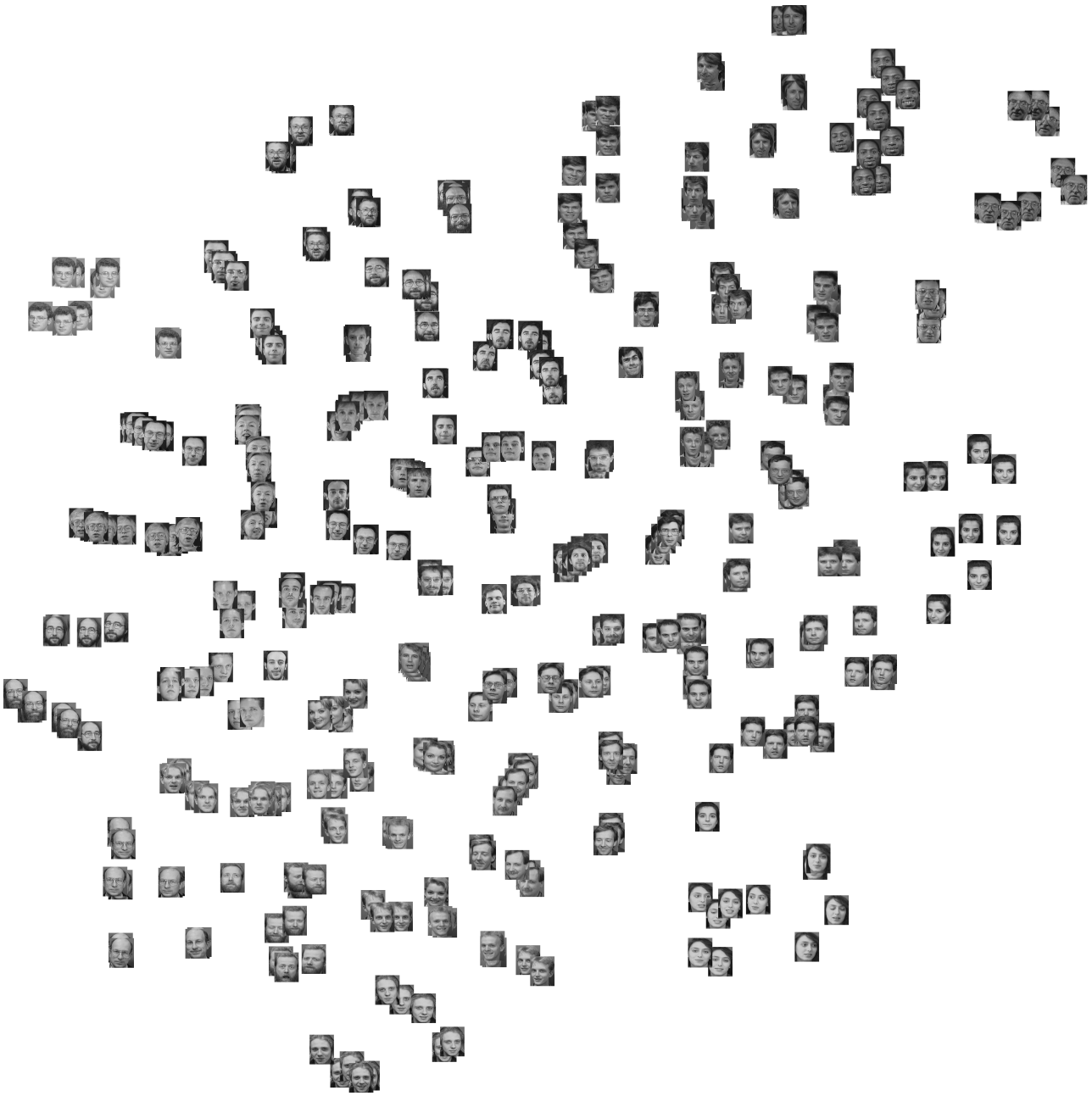

Figure 12: Visualization of the Olivetti faces dataset produced by SNE.

Figure 13: Visualization of the Olivetti faces dataset produced by Isomap.

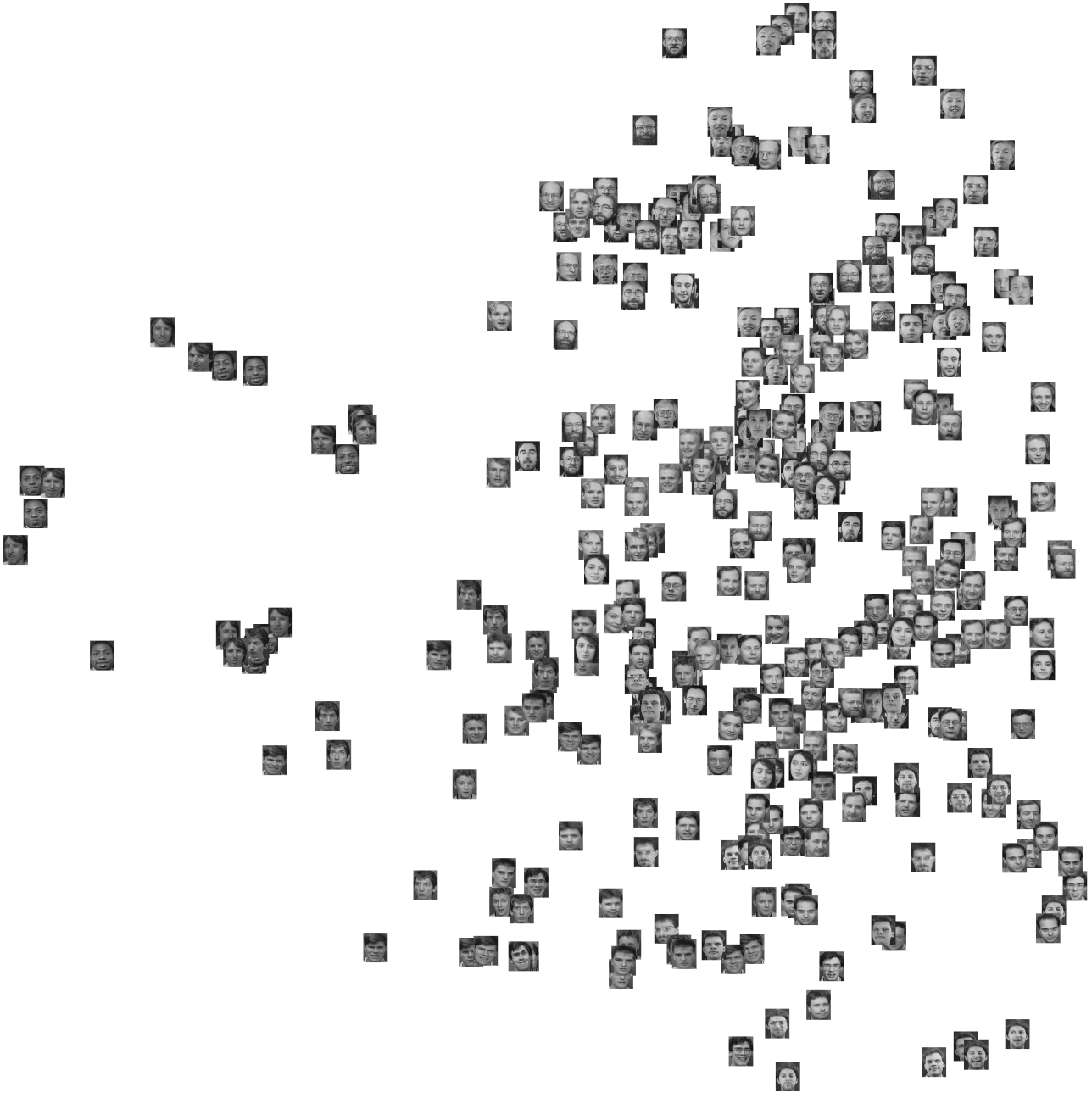

Figure 14: Visualization of the Olivetti faces dataset produced by MVU.

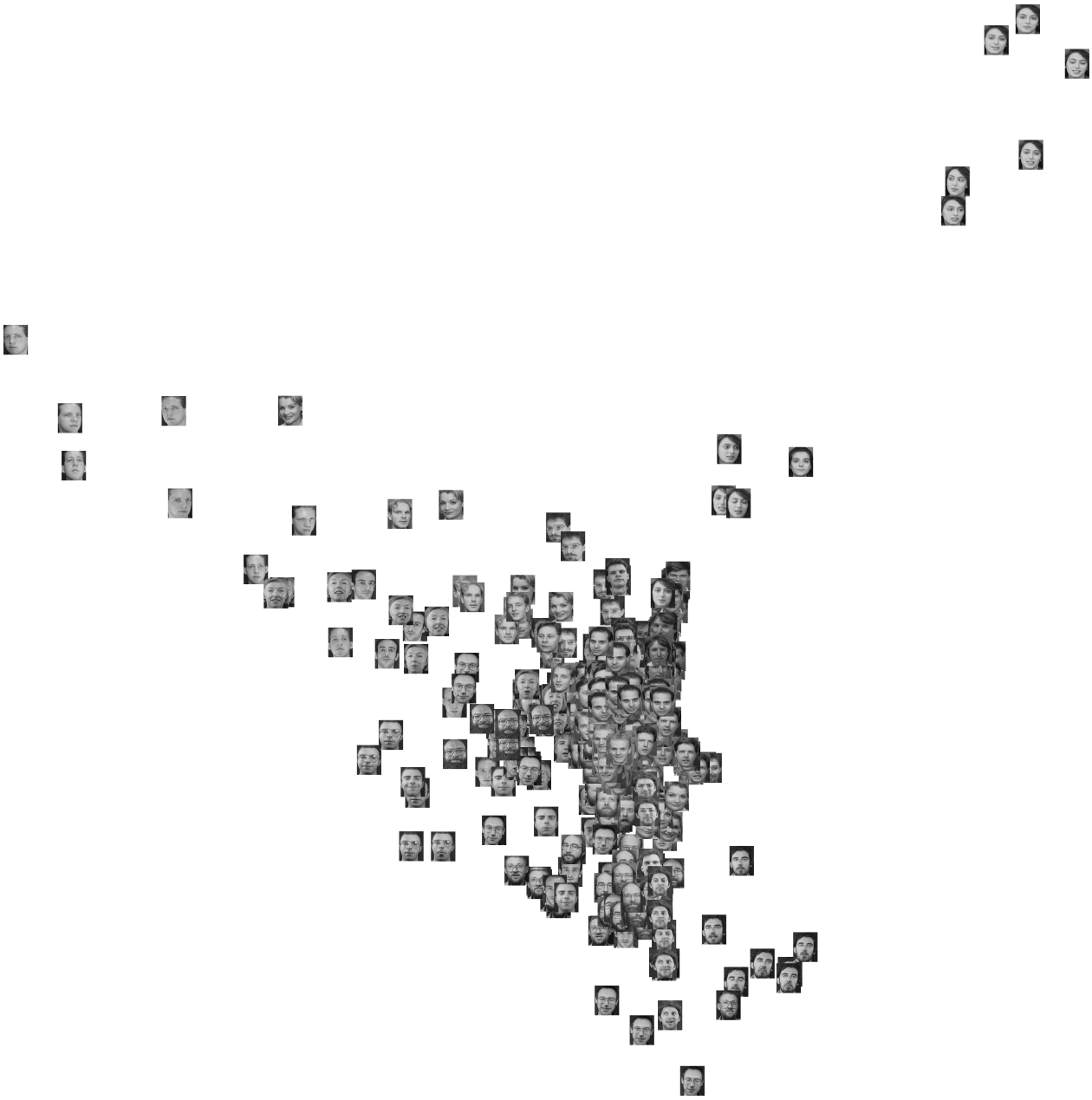

Figure 15: Visualization of the Olivetti faces dataset produced by LLE.

Figure 16: Visualization of the Olivetti faces dataset produced by Laplacian Eigenmaps.

Figure 17: Visualization of the COIL-20 dataset produced by t-SNE.

Figure 18: Visualization of the COIL-20 dataset produced by Sammon mapping.

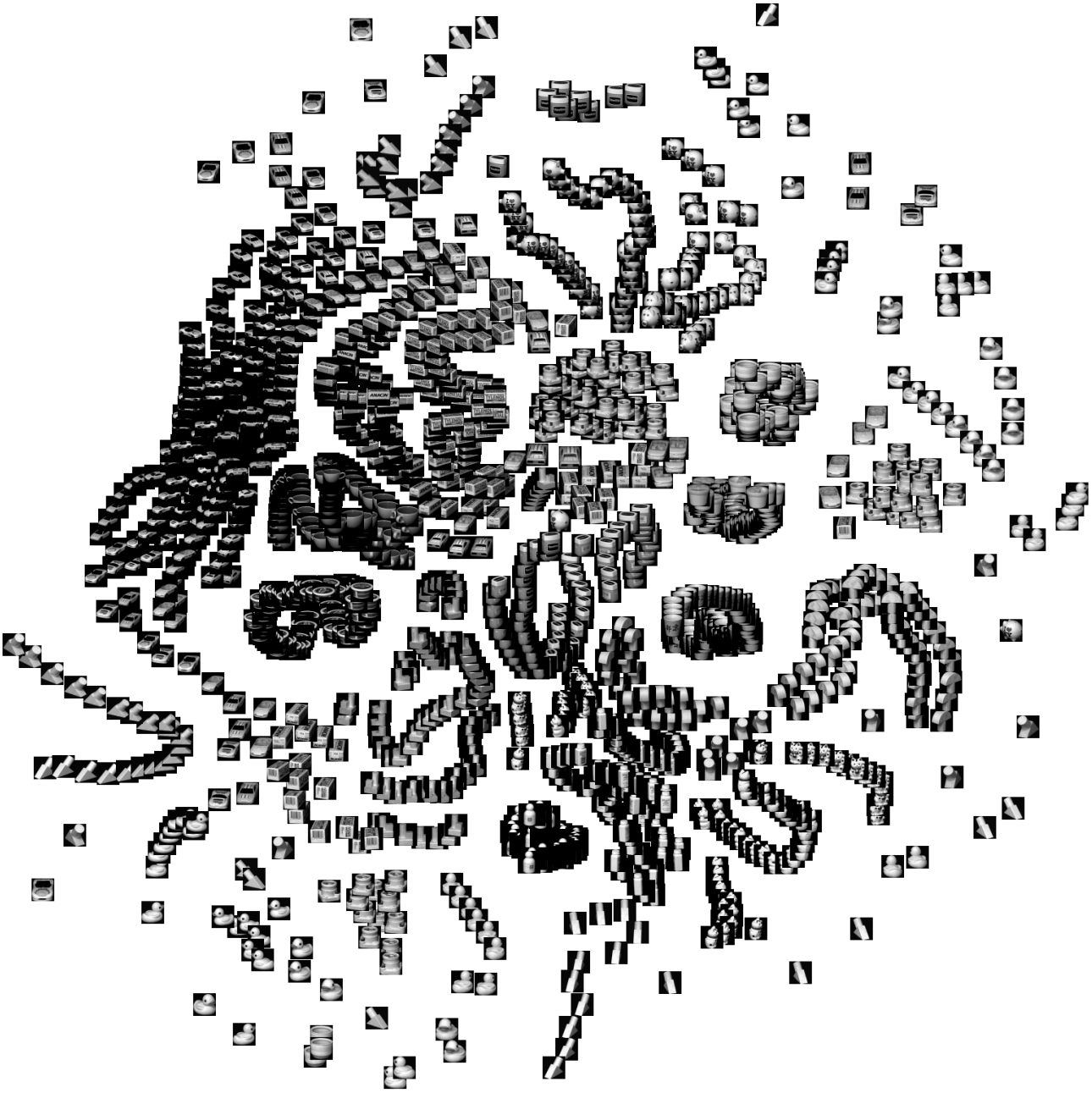

Figure 19: Visualization of the COIL-20 dataset produced by CCA.

Figure 20: Visualization of the COIL-20 dataset produced by SNE.

Figure 21: Visualization of the COIL-20 dataset produced by Isomap.

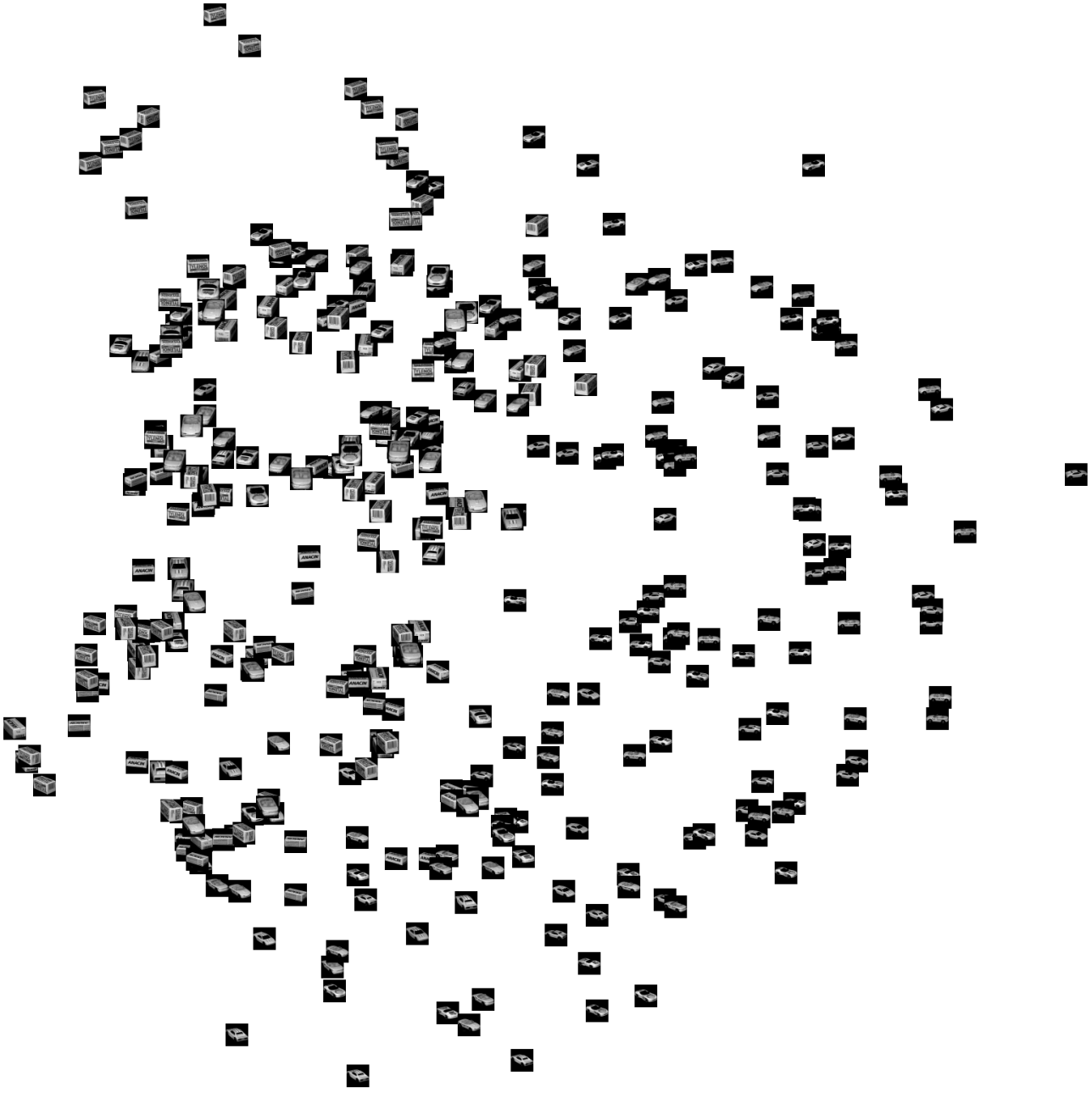

Figure 22: Visualization of the COIL-20 dataset produced by MVU.

Figure 23: Visualization of the COIL-20 dataset produced by LLE.

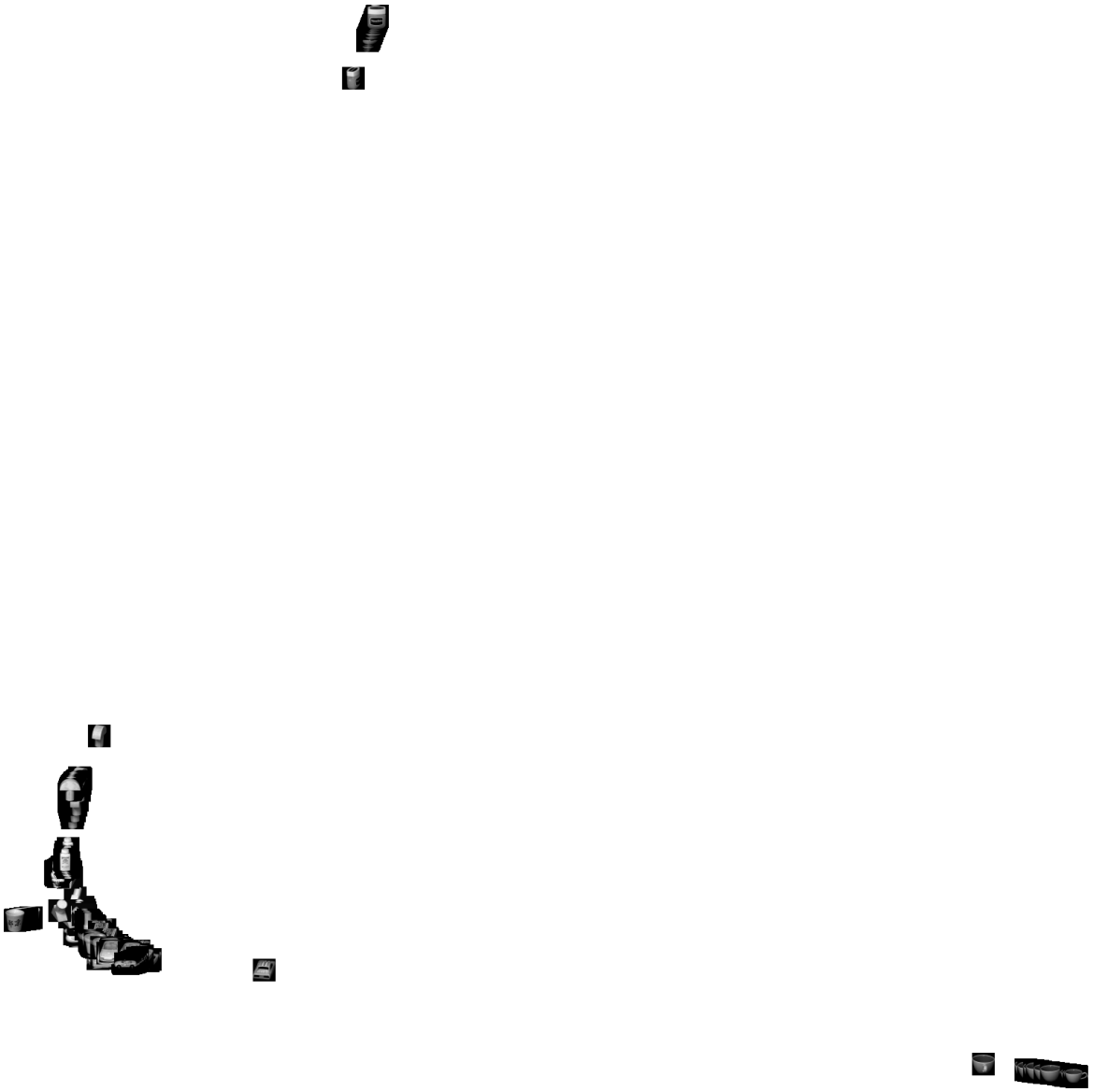

Figure 24: Visualization of the COIL-20 dataset produced by Laplacian Eigenmaps.

Figure 25: Visualization of the word-features dataset produced by t-SNE.

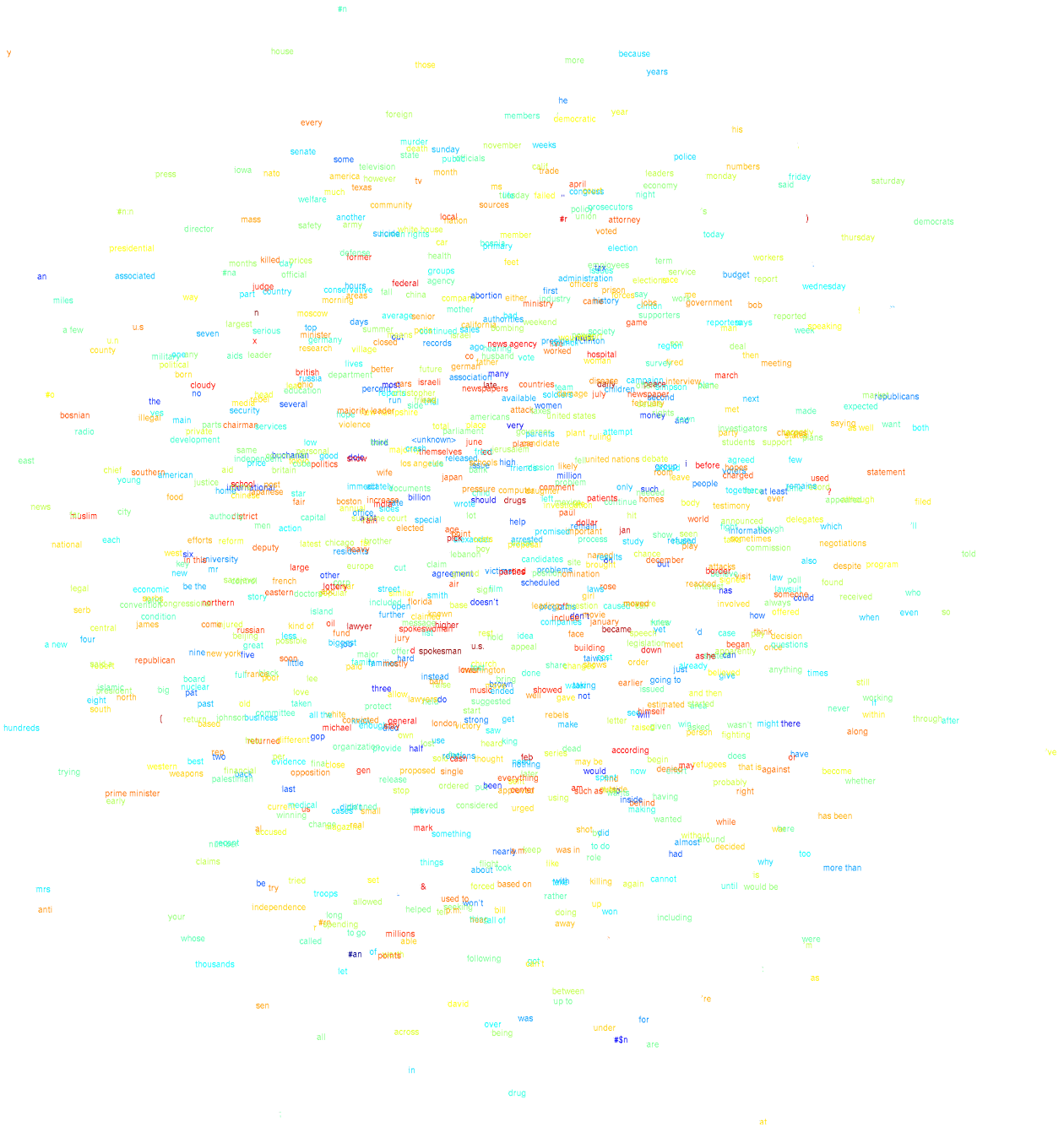

Figure 26: Visualization of the word-features dataset produced by Sammon mapping.

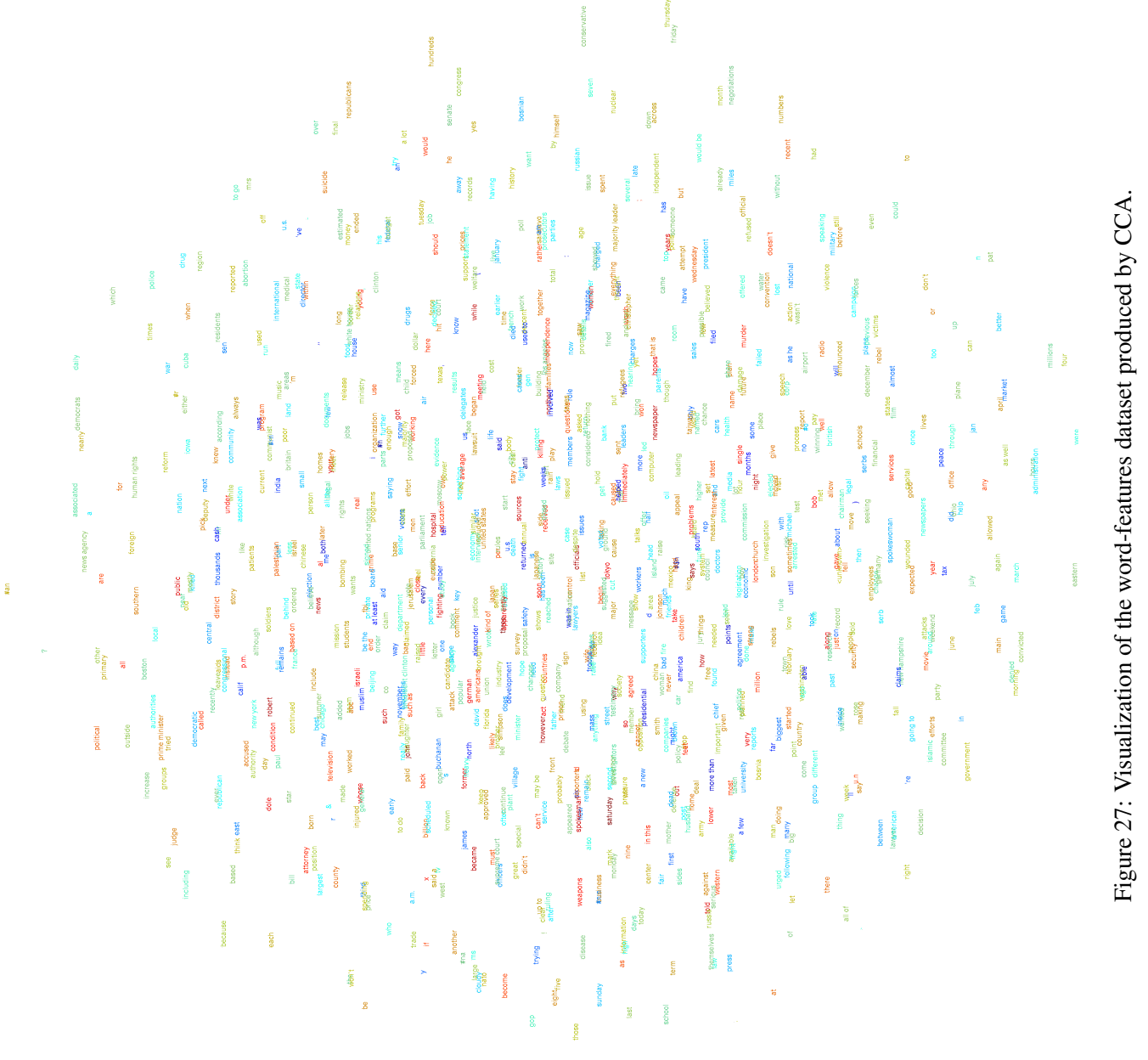

Figure 27: Visualization of the word-features dataset produced by CCA.

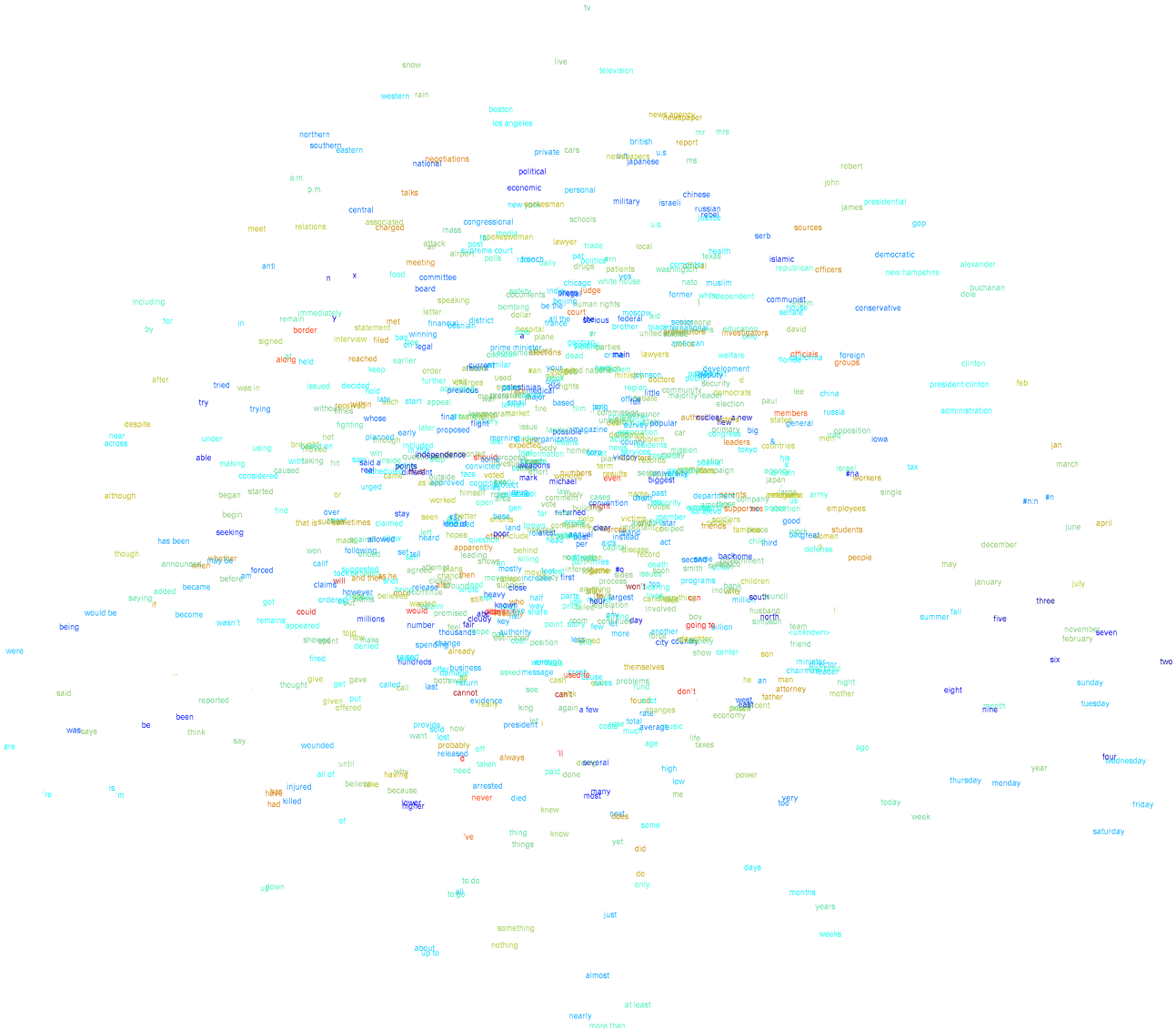

Figure 28: Visualization of the word-features dataset produced by SNE.

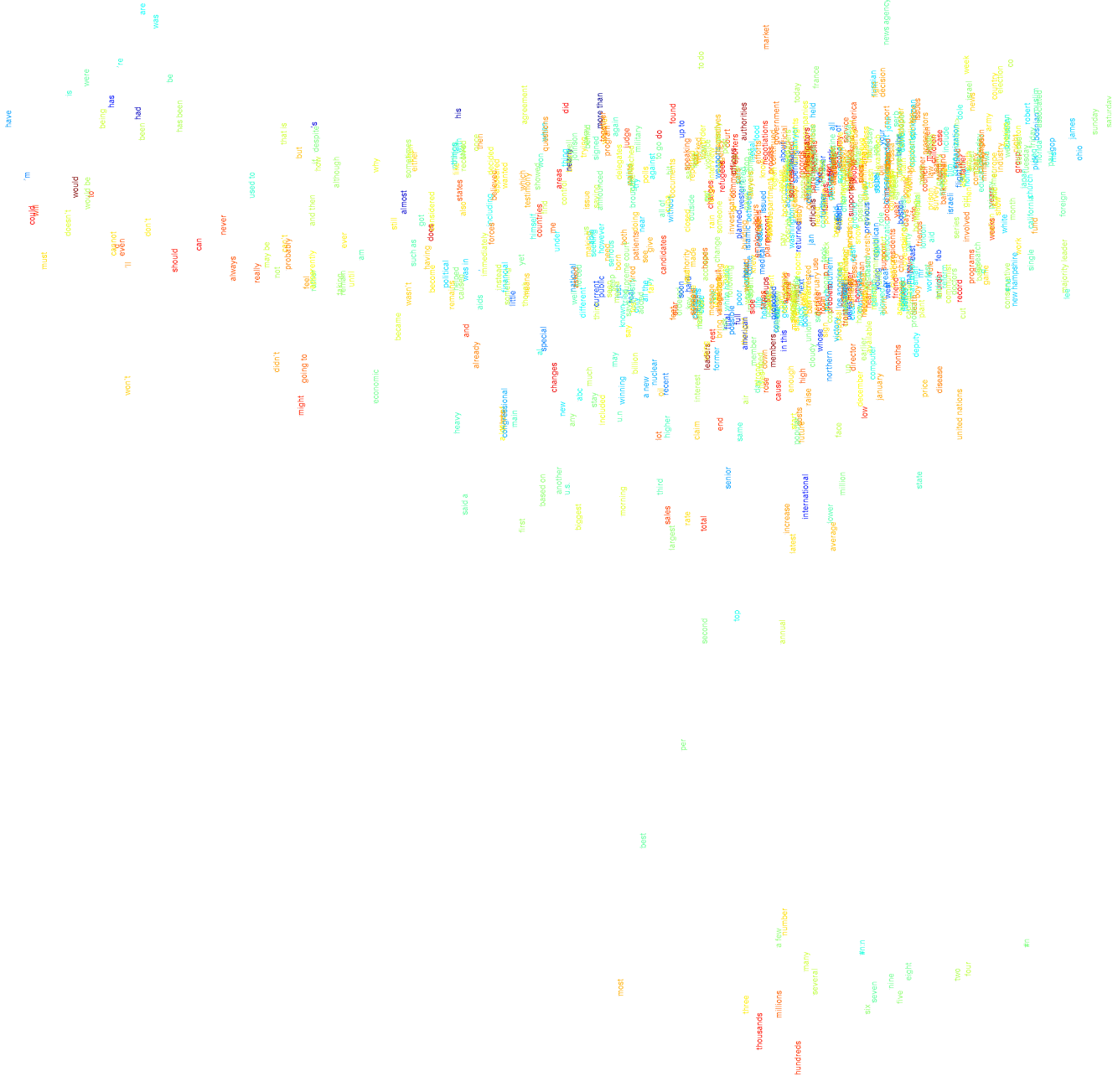

Figure 29: Visualization of the word-features dataset produced by Isomap.

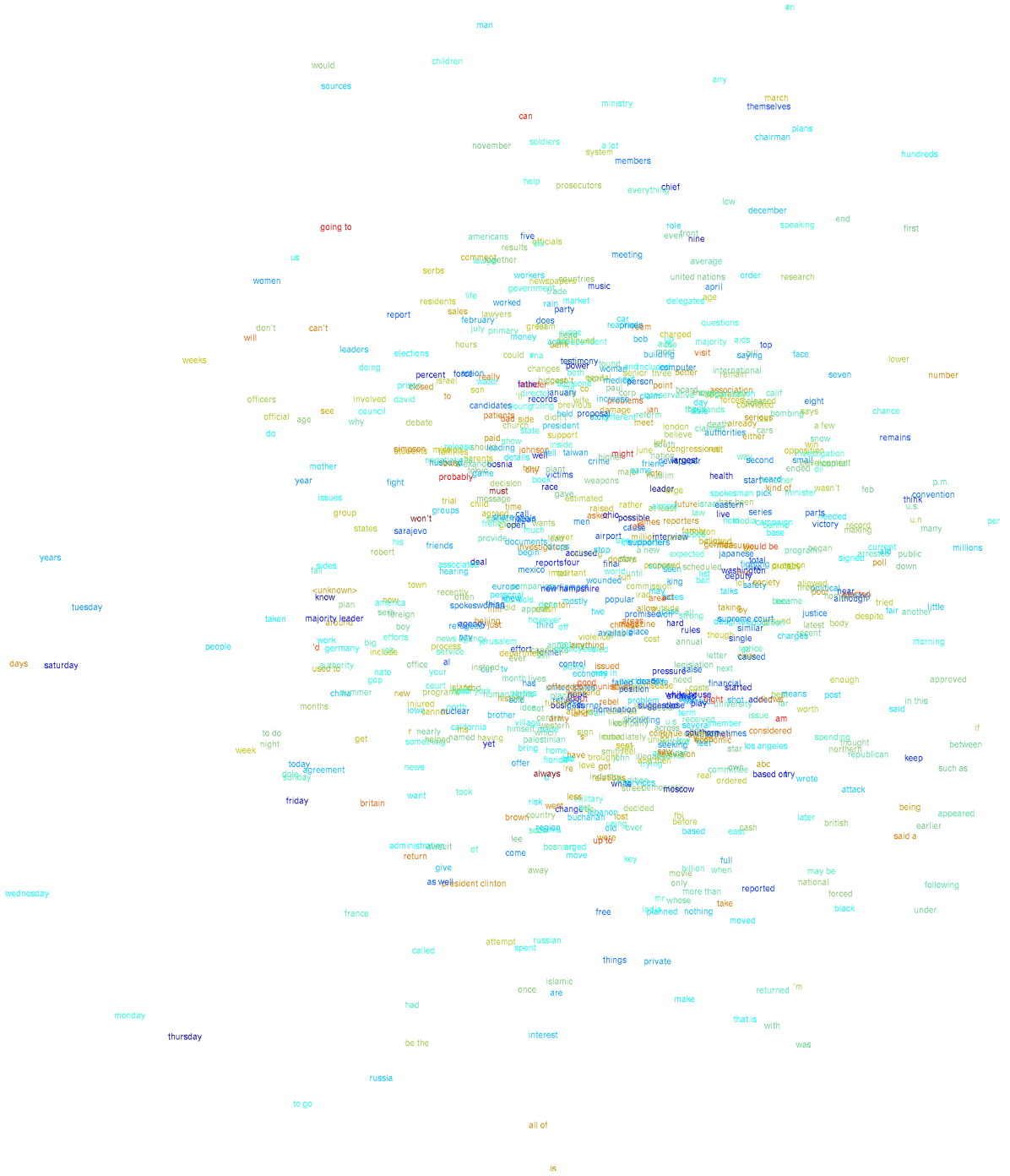

Figure 30: Visualization of the word-features dataset produced by MVU.

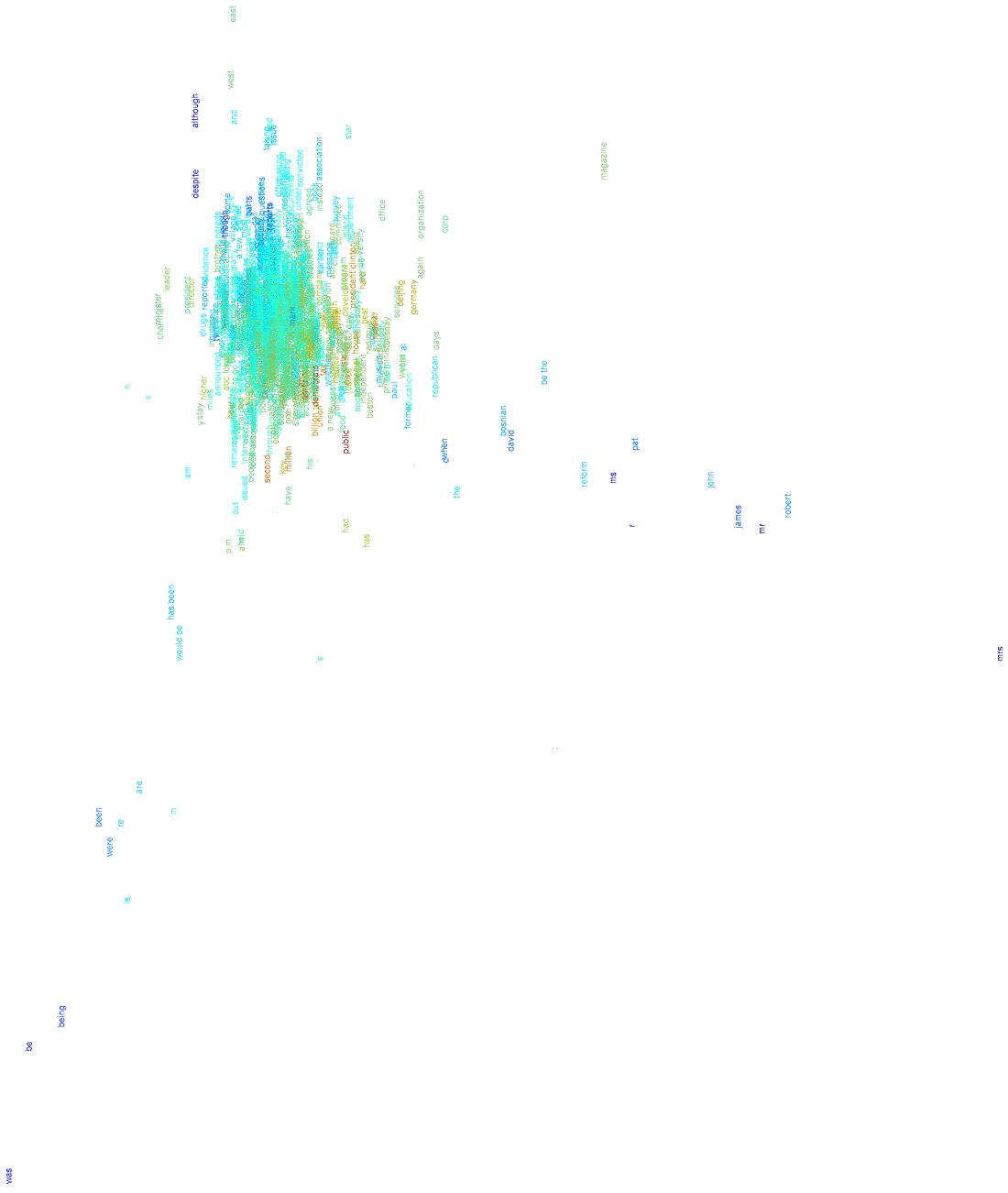

Figure 31: Visualization of the word-features dataset produced by LLE.

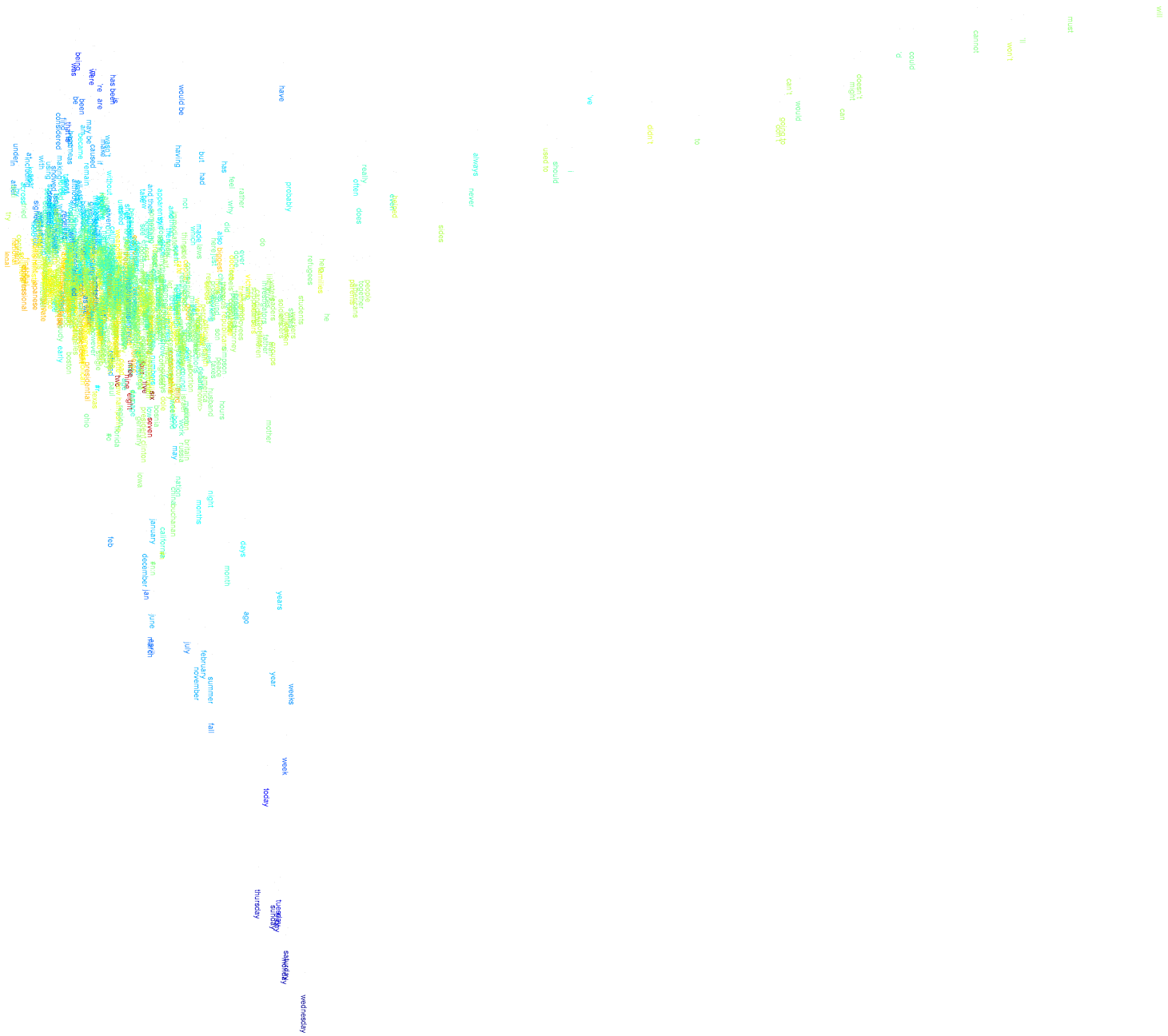

Figure 32: Visualization of the word-features dataset produced by Laplacian Eigenmaps.

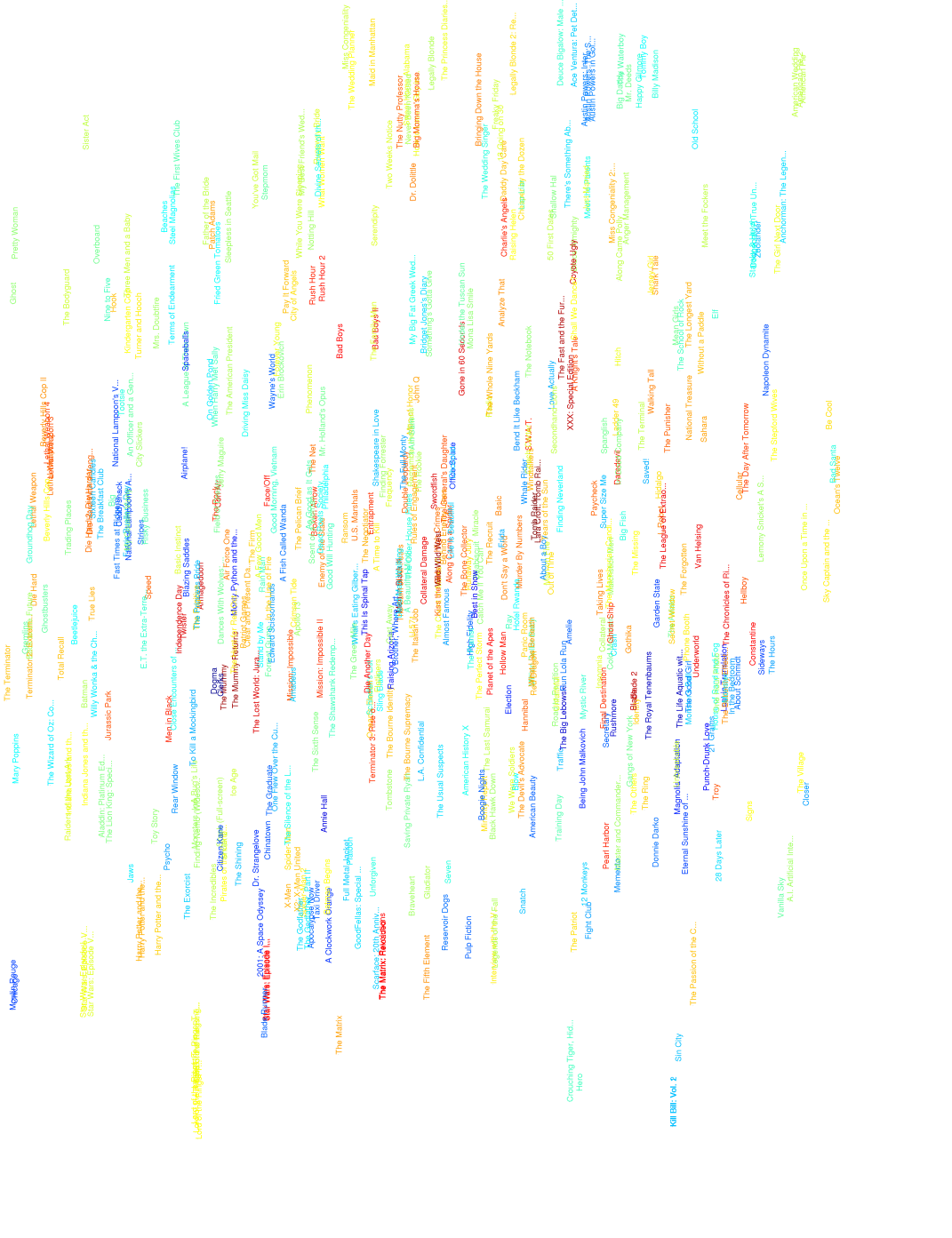

Figure 33: Visualization of the Netflix dataset produced by t-SNE.

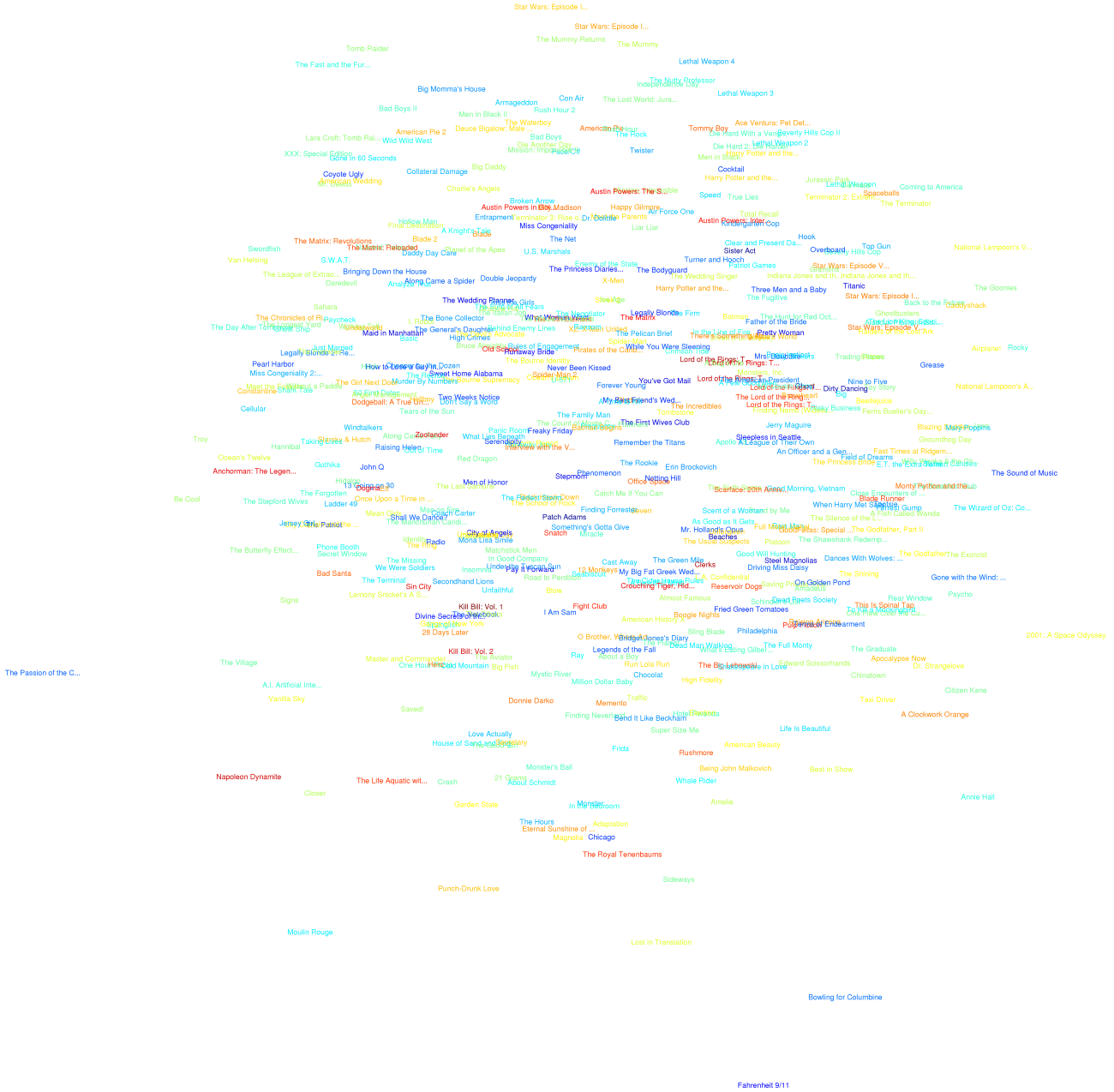

Figure 34: Visualization of the Netflix dataset produced by Sammon mapping.

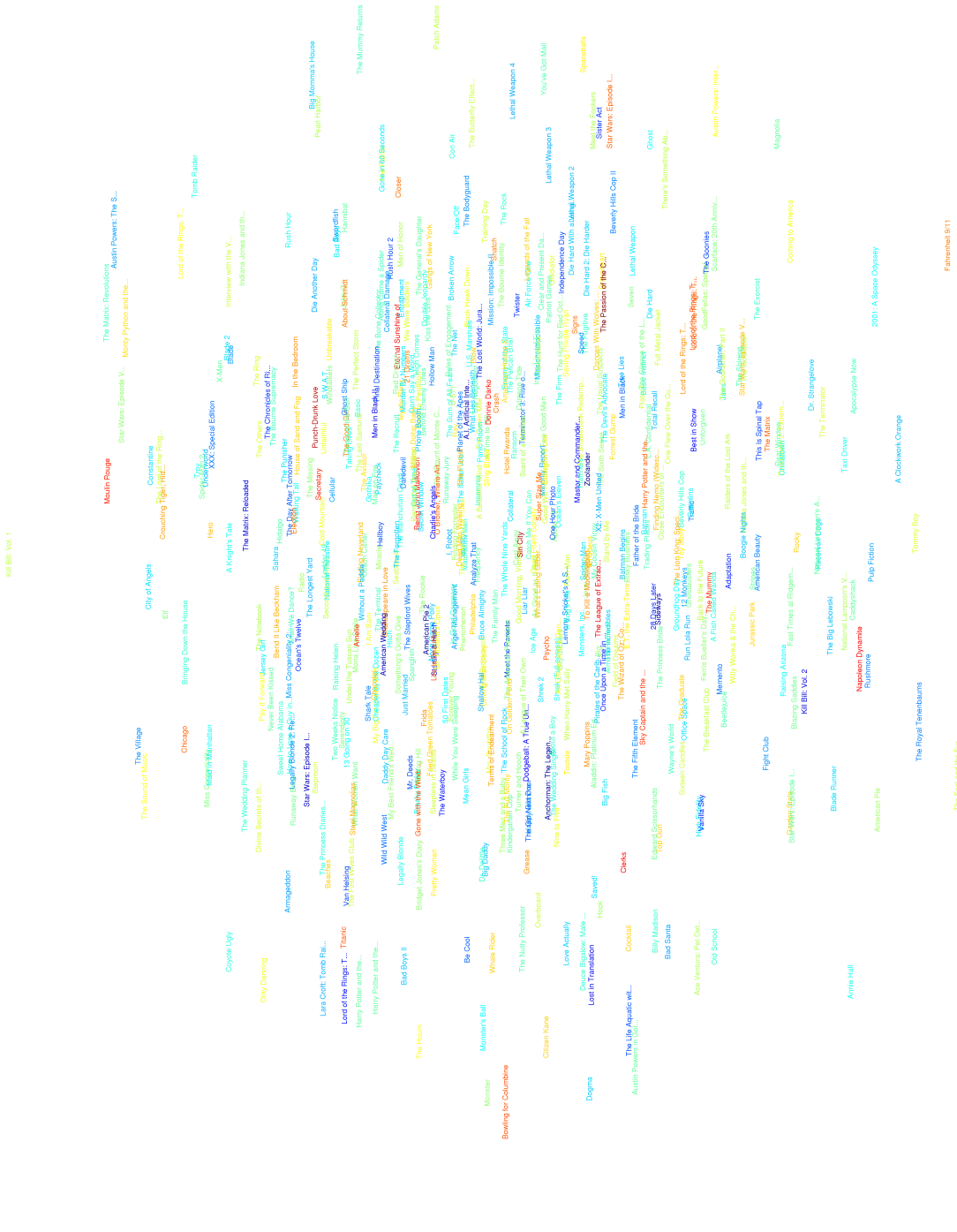

Figure 35: Visualization of the Netflix dataset produced by CCA.

Figure 36: Visualization of the Netflix dataset produced by SNE.

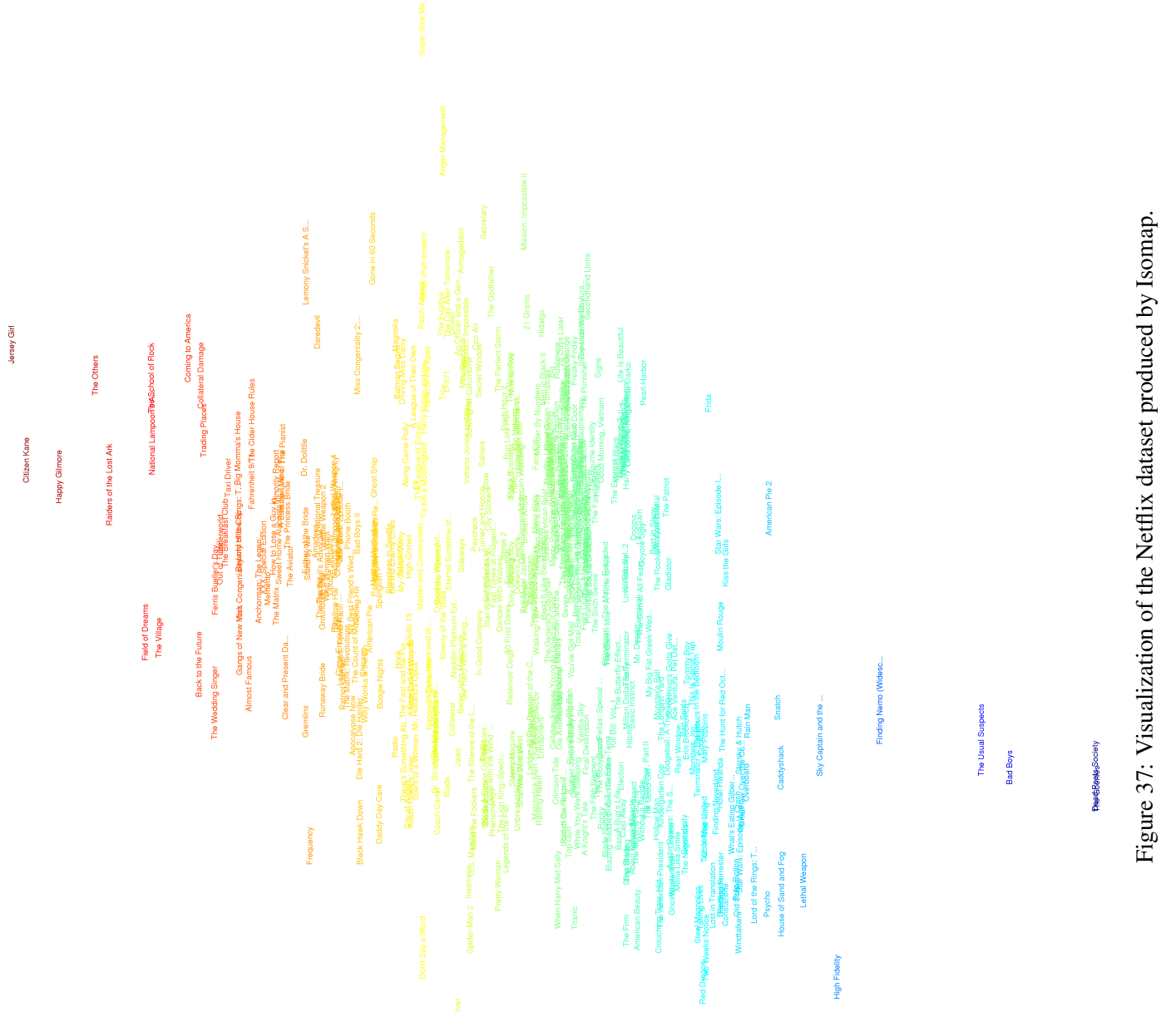

Figure 37: Visualization of the Netflix dataset produced by Isomap.

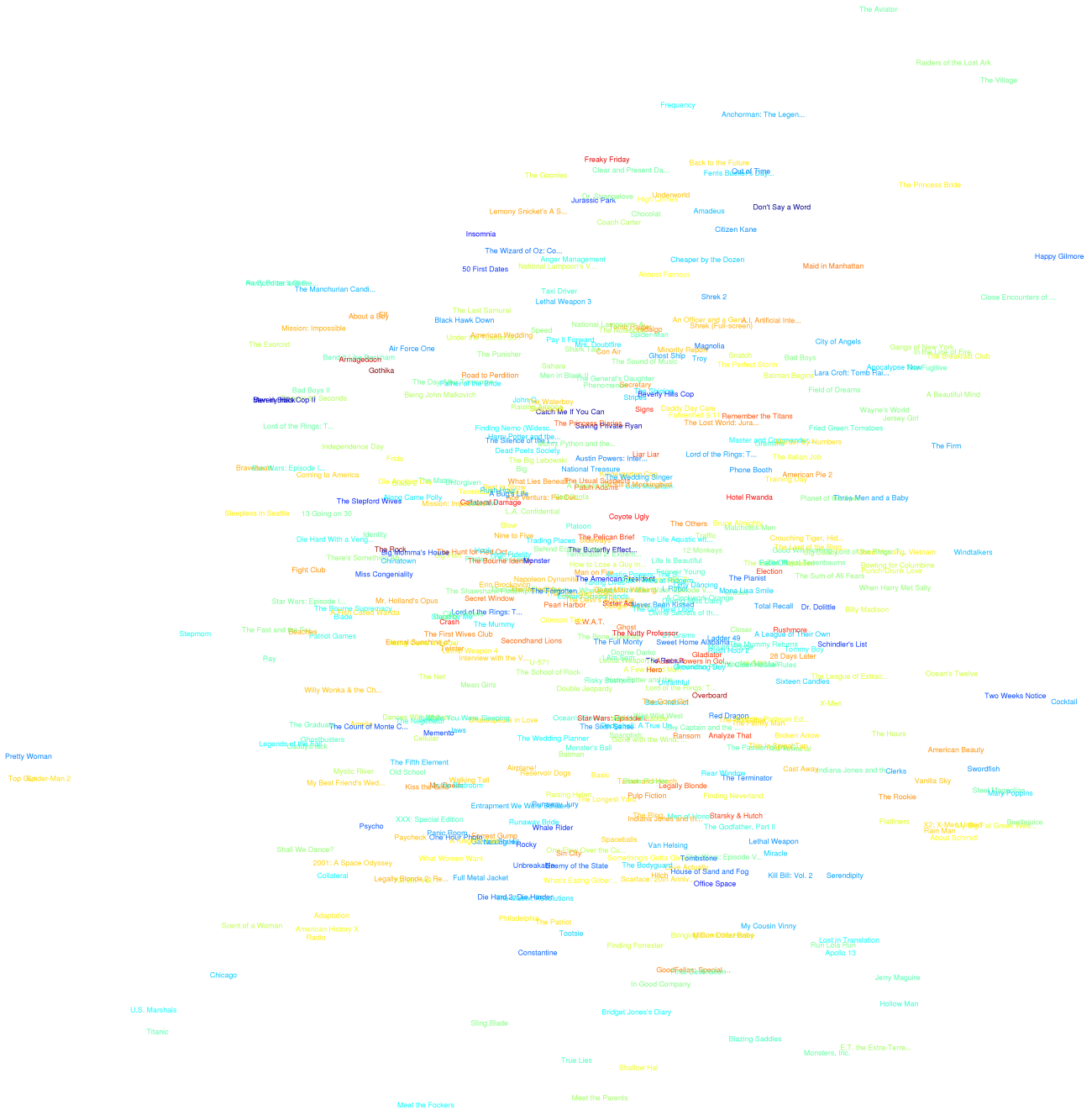

Figure 38: Visualization of the Netflix dataset produced by MVU.

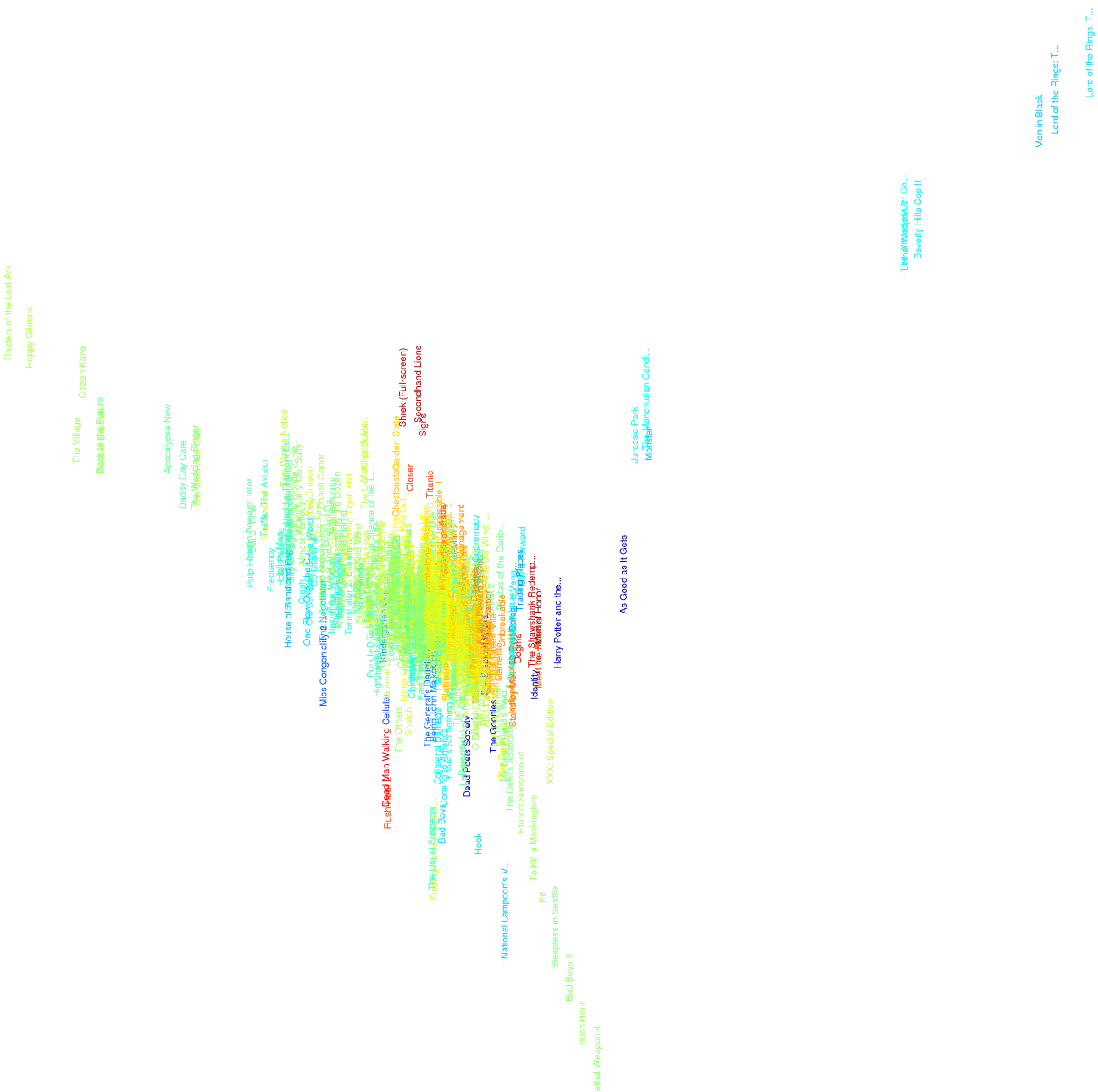

Figure 39: Visualization of the Netflix dataset produced by LLE.

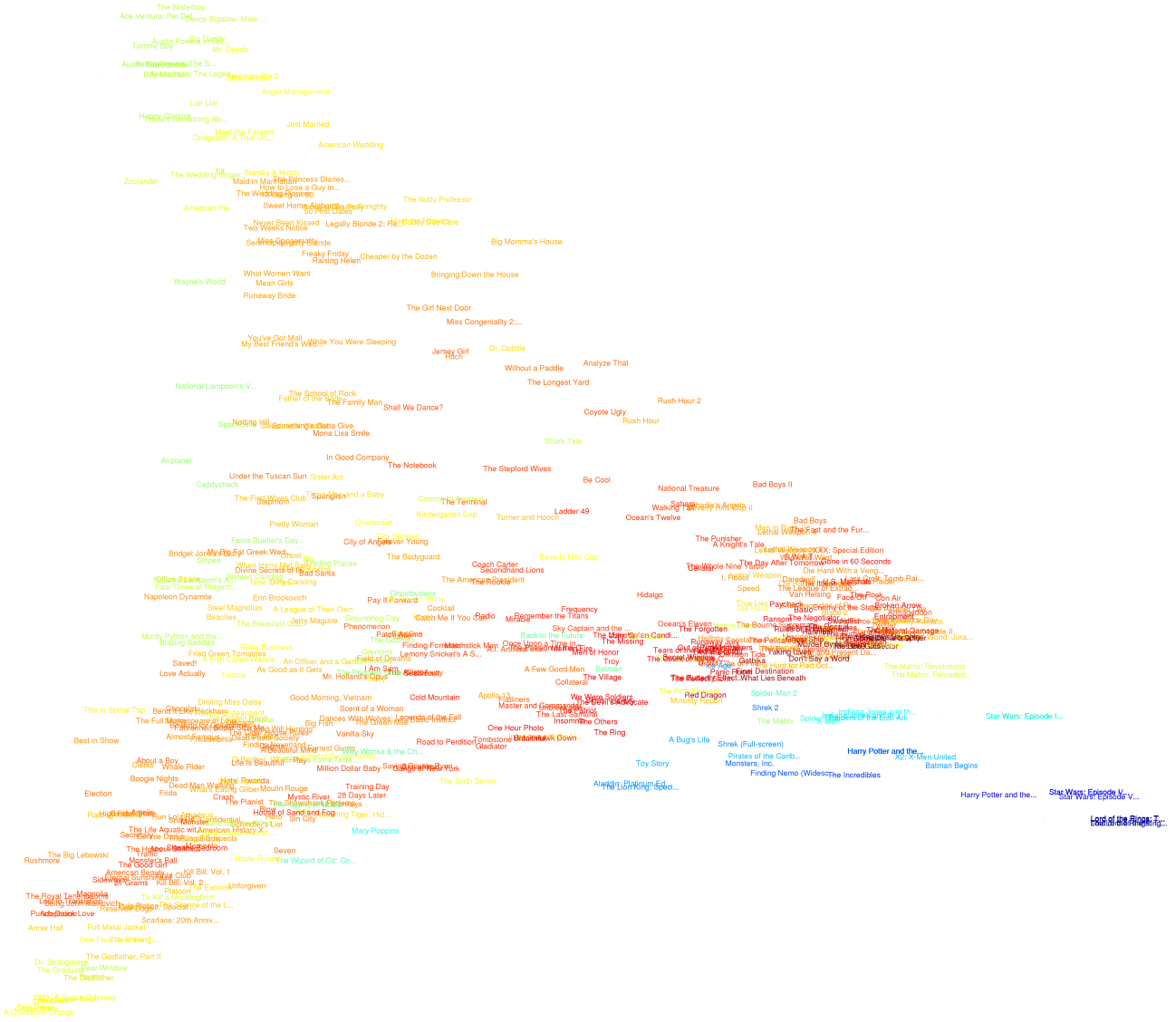

Figure 40: Visualization of the Netflix dataset produced by Laplacian Eigenmaps.

Acknowledgements

The authors thank Andriy Mnih for supplying the word-features dataset, and Ruslan Salakhutdinov for help with the Netflix dataset. We thank Guido de Croon for pointing us to the analytical solution of the random walk probabilities.
Laurens van der Maaten is supported by NWO-CATCH, project RICH (grant 640.002.401), and cooperates with RACM. Geoffrey Hinton is a fellow of CIFAR and is supported by grants from NSERC and CFI and gifts from Google, and Microsoft.

References

A. Mnih and G.E. Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the* 24^{th} *International Conference on Machine Learning*, pages 641–648, 2007.

S.A. Nene, S.K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). Technical Report CUCS-005-96, Columbia University, 1996.

R.R. Salakhutdinov, A. Mnih, and G.E. Hinton. Restricted Boltzmann Machines for collaborative filtering. In *Proceedings of the* 24^{th} *International Conference on Machine Learning*, pages 791–798, 2007.

K.Q. Weinberger, F. Sha, Q. Zhu, and L.K. Saul. Graph Laplacian regularization for large-scale semidefinite programming. In *Advances in Neural Information Processing Systems*, volume 19, 2007.