Large-Scale Interactive Visualizations of Nearly 12,000 Digital Games

James Owen Ryan[1,2], Eric Kaltman[1], Andrew Max Fisher[3], Timothy Hong[3],
Taylor Owen-Milner[3], Michael Mateas[1], and Noah Wardrip-Fruin[1]
[1] Expressive Intelligence Studio, [2] Natural Language and Dialogue Systems Lab
[3] Department of Computer Science
University of California, Santa Cruz
{jor, ekaltman, michaelm, nwf}@soe.ucsc.edu
{anmfishe, thon6, towenmil}@ucsc.edu

ABSTRACT

We present three large-scale interactive visualizations of nearly 12,000 digital games. These were built using techniques from natural language processing and machine learning, namely latent semantic analysis, clustering analysis, and multidimensional scaling. In this paper, we briefly describe these visualizations and some of the insights that they offer. All three are hosted online as interactive web apps.

1. INTRODUCTION

There is a growing body of work in which techniques from natural language processing (NLP) and machine learning are applied to large collections of text about digital games. In [3], we review this promising literature and find that it exhibits a major shortcoming: machine learning models are hard to interpret except through visualization or interaction, but the models produced by this body of work can only be engaged through the prose of their respective publications. In this paper, we present three interactive visualizations that each give unique insight into a complex model that we have built using text describing 11,829 digital games. Beyond clarifying our model, we find that these visualizations stand on their own as unique ontological expressions of the videogame medium.

2. LATENT SEMANTIC ANALYSIS MODEL

All of the visualizations we present here derive from a *latent semantic analysis* (LSA) model trained on Wikipedia articles describing videogames[1]. LSA is an NLP technique by which words are attributed vectorial semantic representations according to their contextual distributions across a large collection of text [2]. From such a corpus, a *co-occurrence matrix* of its words and documents is built; this matrix specifies which words occurred in which documents (and thereby which documents words occurred in). The columns and rows in this matrix can be thought of as vectors that represent the meanings, in an approximate sense, of the words and documents that they correspond to—this is called a *vector space model* of semantics. LSA is an example of such a model, but its hallmark is that it reduces the dimensionality of these vectors by a matrix factorization algorithm. Remarkably, doing this allows the model to infer semantic associations that are not encoded in the full co-occurrence matrix [2]. Having an LSA model, one can easily calculate how semantically related any of its documents are by taking the cosine between their LSA vectors. In corpora in which each document pertains to a specific individual concept, these relatedness scores can reasonably be utilized as a measure of the relatedness of the concepts themselves. Relying on this notion, we trained an LSA model on a corpus comprising Wikipedia articles for 11,829 videogames. By this model, which has 207 dimensions, we can quantify how related any two of these games are by taking the cosine between their LSA vectors.

[1]We invite the interested reader to consult [3], in which we discuss LSA and our model at much greater depth.

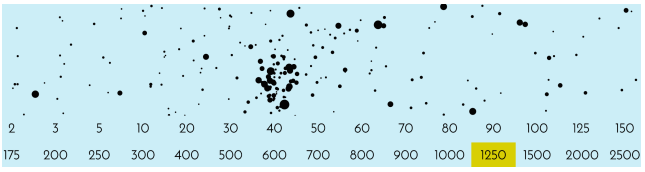

Figure 1: Detail from a GameGlobs clustering.

3. VISUALIZATIONS

Briefly, we will describe the three visualizations we have developed so far, though we encourage the reader to also try them for herself by following the link given in Section 5.

GameGlobs. GameGlobs is a two-dimensional visualization of various clusterings of the games in our LSA model. A user selects how many clusters (groups of related games) she would like to see the 11,829 games partitioned into and is presented with such a clustering, as shown in Figure 1. Each cluster is drawn as a circle that can be clicked to display the games it contains, which are stylized as hyperlinks to their entries in a tool we have built called GameNet [3]. The clusterings themselves were derived by applying the classic *k-means* [4] algorithm to the games' LSA vectors. GameGlobs includes clusterings using several values for k (number of clusters) spanning between 2 and 2500 and utilizes two key visual cues: clusters with more games appear larger, and clusters are positioned semantically, such that clusters whose games are more similar are nearer one another. To achieve the latter effect, we used a technique called *multidimensional*

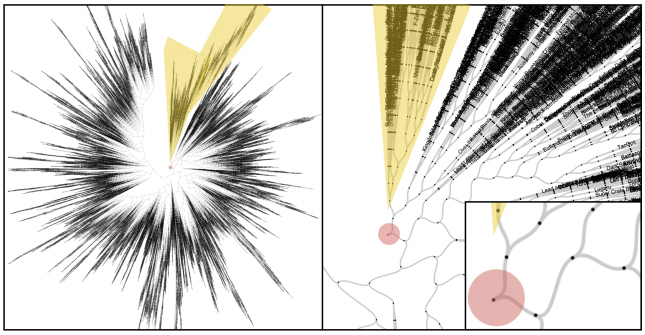

Figure 2: GameTree from afar and closer up (root node emphasized in red; racing branch in yellow).

scaling (MDS), which is a way of building low-dimensional visualizations of high-dimensional data [1]. This technique is represented by a suite of algorithms; we submitted the LSA vectors of our cluster centroids to a variant called *locally linear embedding* (LLE) [1] to derive their 2D coordinates.

GameTree. GameTree is a massive two-dimensional visualization of a hierarchical taxonomy of the games in our LSA model. The underlying representation is a tree that was built bottom-up by submitting the games' LSA vectors to an algorithm called *hierarchical agglomerative clustering* [4], which works as follows: each of a set of objects is initialized to be its own cluster; on each iteration, the two clusters whose centroids are most similar are merged into a higher-level cluster, whose centroid gets set as the mean of those two centroids; this repeats iteratively until a root node is formed by merging the last two remaining clusters. Figure 2 shows the visualization, which is a *radial tree*.

GameSpace. GameSpace is an explorable three-dimensional ontological space in which each of our LSA model's 11,829 games is represented as a data-rich star whose positioning is semantically meaningful; Figure 3 shows its most central portion. Specifically, games are placed in the space such that their most related games are nearby. Three-dimensional coordinates for the games were derived by submitting their LSA vectors to LLE, as in GameGlobs. The user can fly freely through the space using conventional 3D game controls, and upon encountering a game she can engage it (by clicking it) to explore data about it: its title and year of release; an embedded YouTube player with a *Let's Play* video preloaded; an embedded pane displaying its Wikipedia page; another exhibiting images of the game; and one more showing its entry in GameNet, our LSA-fueled game-discovery tool that we have described elsewhere [3].

4. DISCUSSION

Like most machine learning models, our LSA model is by itself largely uninterpretable; it is too high-dimensional to visualize and its 207 dimensions are themselves obscure linear formulas that characterize complex statistical phenomena. Because we cannot look at the thing itself and understand it, we have to build tools and visualizations to get at what insight the model can give. But as it *is* so high-dimensional and complex, interpretable interfaces to the native model will only ever approximate it, or faithfully depict some facet of it. This is the impetus for building multiple visualizations of a single complex model: if they each work to express discrete aspects of the model, together they may work to afford a fuller understanding of the entire thing.

We built our LSA model to operationalize a notion of

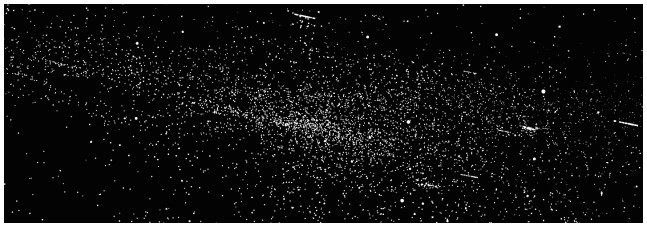

Figure 3: A wide view of central GameSpace; its denser core is made up of combat-oriented games.

game relatedness that proceeds bottom-up according to how games are actually described, and the above visualizations express this notion in complementary ways. GameGlobs shows how games might be *discretely* categorized by our model, as in a conventional genre typology. GameTree is also made up of discrete clusters, but in a way that expresses a more continuous notion of game classification. If its clusters are thought to represent individual game genres, GameTree represents a *genre hierarchy*, interestingly. That is, its components constitute genres at varying granularities, from the finest possible (genres that each comprise a single game) to the coarsest possible (one genre comprising every game), and everything in between. (We informally observe that conventional notions of game genre seem to correspond to nodes at around the third highest level of the tree; for instance, the protrusion that extends directly upward from the root of the tree, shown in Figure 2, is composed almost entirely of racing games.) GameSpace, then, is fully continuous in its game classification—games get clustered, messily, inasmuch as cluster-like formations might emerge naturally from sets of related games being placed near one another. Another interesting artifact of GameSpace's particular continuous spatial representation (and the algorithms that derived it) is that obscure games reside largely at the perimeter of the space, while more conventional games are central (its denser core is composed mostly of combat-oriented games).

Beyond clarifying our model, we believe these visualizations stand on their own as unique ontological expressions of the videogame medium; each features its own affordances and, as we have just discussed, lends its own insights. Moving forward, we plan to explore the use of these visualizations as tools for game discovery, perhaps in the context of library and archival collections.

5. LINKS

Try the visualizations at `http://gamecip.soe.ucsc.edu/projects`.

6. ACKNOWLEDGMENTS

This project was made possible in part by Institute of Museum and Library Services grant LG-06-13-0205-13.

7. REFERENCES

[1] T. Cox and M. Cox. *Multidimensional scaling*. 2000.

[2] S. T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 2004.

[3] J. O. Ryan, E. Kaltman, M. Mateas, and N. Wardrip-Fruin. What we talk about when we talk about games: Bottom-up game studies using natural language processing. In *Proc. FDG*, 2015.

[4] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *Proc. KDD Workshop on Text Mining*, 2000.