Abstract Data Visualization Using Game reviews

Magnus Håkon Petersen

Aalborg University Copenhagen

Supervisor: George Palamas

Aalborg University Copenhagen

Aalborg Univeristy Copenhagen.
A.C. Meyers Vænge 15, 2450.
Copenhagen SV, Denmark.
Secretary: Lisbeth Nykjær.
Mail: lny@create.aau.dk
Phone: 99 40 24 70

Semester: 6^{th} semseter, Medialogy (BSc)

Title:

Abstract Data Visualization
Using Game reviews

Project period: June 23^{th}, 2020 - September 24^{th}, 2020

Semester theme: Master Thesis

Supervisor: George Palamas

Group members:

Magnus Håkon Petersen

Deadline: September 24^{th}, 2020

Pages: 42

Abstract

This paper explore abstract data visualization and the relationship between games, by using game reviews. Sentiment analysis and topic embedding is used to ascertain relationships between games, in the sense of the similarities in game-play, themes, audience and anything before undiscovered. Game reviews for games are shown to be very similar over a great variety of games, when using sentiment and topic embedding as the primary factors for comparison. However in order to understand the data better, an abstract visualization is made and tested, this visualization reveals beforehand hidden relationships in the data. However the number of test participants and the method of abstraction puts the validity of the findings in question.

Table of content

List of Figures

1 Introduction

Data analysis especially when dealing with large complex data sets can be a laborious and highly specialized task. Several data plots and visualization might be required to ascertain useful information. Large part of these data visualizations deal with dimension reduced data, where understanding the relationship in the data require understanding of the dimension reduction technique. This limits data analysis to a certain and rather small group of people. This is can become a problem when the analyzation of data might require a pair of fresh eyes or radical ideas. Further more people who are unable to read and understand the data, is very much still affected by the data and the decisions made about the data. In order to try an cooperate more people of any work pipeline dealing with large data, abstract data visualization is explored in this paper. First is needed an understand of why we perceive classes and relationships in data visualization. Then the ways data can be analysed, in this case the data in question text. Here sentiment analysis and topic embedding is explored as ways of creating objective values from textual data. Then dimension reduction is explored, the most common ways and how they work. lastly a look at how other people have made data visualization in abstract and concrete ways. A large data set of Steam game reviews is explored and a method for abstract data visualization is then created, with the goal of finding otherwise overseen relationships in data. The abstract data visualization is then tested on people of different backgrounds and professions.

2 Analysis

2.1 Gestalt Theory

When analyzing data it is important to understand how and why some data are similar to other, why we consider some data to be part of a distinct group or to be random. When just looking at raw data all you see is numbers, you could scroll through page after page of numbers and when tasked to group the data into 4 classes you would be at a loss. This is when data visualization and gestalt principles plays a role in the human understanding of data, if each set of numbers were to be visualized as a dot on a 2d grid suddenly the task of grouping the data into classes becomes much easier as the dots might cluster together in shapes or you could draw clear lines between them [20].
The world is full of similar objects and objects made of many smaller objects, every object having distinct parts and features. yet the human brain can determine what belongs together and what is one entity in relation to another. The understanding of how human's perceive the world is called Gestalt Theory, gestalt is a German word for shape or form, the field of research was first introduced by German psychologists back in the early 1920s [20].

2.1.1 Gestalt Principles

When understanding Gestalt Theory, the concept is surmised in principles called the Gestalt principles or Gestalt laws. Each principle dealing with a way the human brain will group visual information[20].
The first principle often not even considered a Gestalt principle is figure-ground articulation, this is when one shape is consistent throughout clearly standing out from its surroundings with differing color. This principle is rarely used in data visualization as data rarely gets visualized in consistent shapes, instead opting for dots and graphs which are easier to make visually distinct. it should however not be dismissed as it is the most obvious form of grouping and processing data into

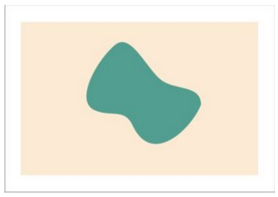
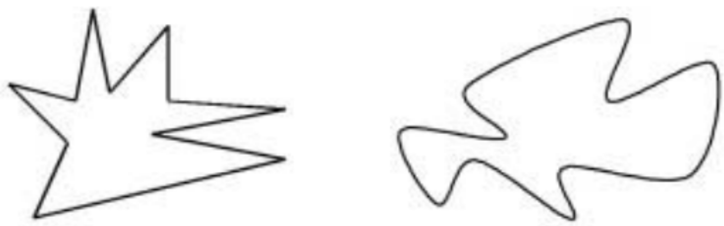


figure-ground articulation [20]

such a shape might yield interesting human responses [20].
Another principle is the proximity principle, objects close to each other are perceived to belong to each other. This principle is especially prevalent when looking at scatter plots, where the data is represented as dots on a graph[20].
The last principle important for this paper is the similarity principle, which states objects of similar shape or color is considered to belong together[20].
The gestalt principles is important when creating new ways of visualizing data. Does the new visualization present similar data to correspond to the principles, people will be able to spot these differences and find a use for these findings. Just using data to create random shapes and colors would be no good.

2.1.2 Bouba/Kiki Effect

A phenomenon in psychology is the Bouba/kiki effect orginally discovered by Köhler in 1929[9]. The bouba/kiki effect describe a common grouping method ingrained into the human mind. When presented with a jagged shape and a more rounded shape, then asked which is Bouba and which is Kiki 95 percent of people will classify Kiki as the jagged and Bouba as the rounded [17]. Ramachandran[17] argued the classification stemmed from the evolution of language, as kik is spoken in short distinct syllables which relates to the sharp edges, while Bouba is spoken softer and so the shapes match articulation.



Bouba Kiki shape example [17]

This effect can be taken into account when creating visualization. Because the human brain already has classified the shapes as fitting together they are a good option for representing similar data.

2.2 Natural language Processing

Natural language processing (NLP) is a broad research field concerning the understanding, manipulation and formatting of language, speech or text. NLP often is used to create computer programs that interact with the language in a plethora of useful applications. Such applications include but are not limited to machine-readable dictionaries, speech synthesis and recognition and cross-language information retrieval[4].
NLP is in other words anything to do with the processing of language. In this paper the focus will be natural language understanding, such a task is multilayered. Many approaches will often begin at the word level.

2.3 Word2Vec

In 2013 a group of researcher[15] made a way to turn words into vector space. The technique is called topic embedding. The vectors is created in such a way that if the vectors were plotted in multidimensional space similar words would be close together. Similar in the sense related to their meaning to humans, such as the words rat and rodent would be close. The capability of Word2Vec did not stop there as when the vectors were put together the resulting vectors would be close to words the 2 words meaning together would hint at, such as capital and Russia put together would be close to the word Moscow. Alongside with their paper, they released the code named word2vec. The word2vec code is publicly available and are still under continuous update and improvement.
Of course it is not the first time words have been turned into vectors, as the notion was first introduced back in 1986 [19] yet over the years the method has been developed and Word2vec is the current state of the art with publicly available code.
In 2003 [2] these vectorized words was popularized in the field of machine learning as a effective neural network language model was made. This was done with a feedforward neural network with a linear projection layer and a non-linear hidden layer. The network would create vectors for each words as to better analyse them. The problem it

tackled was that every word simply has too many variables in its uses in order to be efficiently reduced to a few numbers.
Afterwards many approaches and improvements was made to field of langauge processing through machine learning. One problem persisted, that the words resulting vectors had very little to do with the meaning of the words or the relevance of the words unless of course it was read by a specifically trained neural network.
This is what makes word2vec so revolutionizing, that the created vectors has intrinsic meaning. This is especially useful regarding the topic of data visualization. As topics could be read and transoformed into visuals.
Word2vec is based on the Skip-gram model [14] which in turn is based on the Neural Net Language Model (NNLM) [2] developed in 2003. The NNLM consist of a projection layer divided in constant n size for each word into a hidden layer and finally a output layer. The skip-gram model uses no hidden layer and the projection layer is collapsed to a single layer and thereby shared between multiple words called lexicon. text[15].

2.3.1 Doc2vec

Doc2vec is an extension to word2vec where words are replaced with documents. Documents meaning arrays of single words, referred to as phrases in the paper[10][11]. In their first paper Mikolov Et Al.[15] explained a way to create vector embeddings based on phrases instead of words, here they proposed a way of combining words that frequently appeared together into single tokens, which then could become part of the lexicon, effectively learning the vectors of those phrases[15]. This was done to circumvent the problem of having a n sized input space, that no computer would be able to handle. however in 2014 [11] they came up with a solution to the problem created an efficient model that used entire phrases instead of words.
Doc2vec has since been criticized by subsequent researchers struggling to replicate the results originally produced in Mikolov paper[11]. This is because the in order for Dov2vec to work, a suffciently large and varied data set is required[11]. Several improvements and fixes has then since been proposed [11].
[11]

2.4 Sentiment Analysis

Text data such as reviews, comments or social media posts, usually get shared with certain sentiment. The problem that then occurs is how to format extract the sentiment in way that get any useful statistical information. After a brief contemplation of this problem the most obvious answers would be occurrence of words or symbols, maybe an average count of words and such. Yet quickly it is clear such approaches, might have its applications, but will hardly give any intrinsic meaning the same way a human reading the texts would. This is when sentiment analysis steps in, to convert the humans understanding into numbers[12].

When dealing with language in the current age, the nature of social media with short comment like updates and the reliance on slang and abbreviations has warped the written language and increased the sheer volume of data. This mean that the efficiency and the process of sentiment analysis has become an increasing challenge[12].

many sentiment analysis methods rely on sentiment lexicons, this means long lists of words with labels describing whether they impact the meaning of a sentence positively or negatively or even appealing to different emotions. These lexicons sheer volume of words and information means they are hard to manufacture. The traditional way of creating these lexicons has been by hand, where human has gone in manually rating and listing each word, needless to say this is a very resource demanding task. This means when a lexicon is first made it is usually reused plenty of times and slowly revised to keep up with the changing language[12].

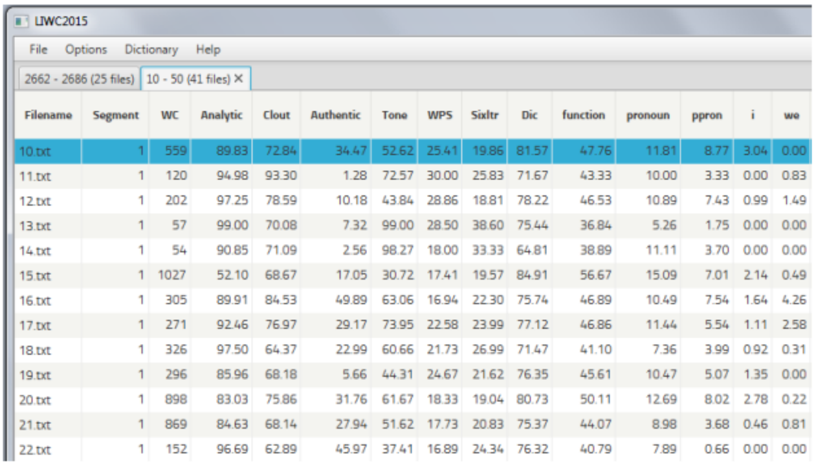
With the revolution in machine learning lexicon creation has changed as machine learning algorithms can help create and rate lexicons. The challenge that occurs is how to make a machine learning application that can perform as good a job as a human or better when creating lexicons. Often the machine learning is at least used in tandem with manpower and has become a stable in state of the art of lexicon creation[12].

One challenge sentiment analysis propose is the sentiment intensity, "good food" is less good food than "the best food", often sentiment analysis classify word either good or bad not taking into consideration that some words are more intense in their meaning.

This does not mean sentiment analysis with sentiment intensity is not useful, the entire process can become more streamlined and it has produced beneficial results such as predicting political indications or predicting depression. Yet it stands to reason a system with sentiment intensity can do the same and create further data[12].
Another challenge in sentiment analysis is context, often a word can be either good or bad according to context. An example being "catch" saying "he's a catch" has positive indication while saying "but there's a catch" has negative. Many sentiment analysis methods often have trouble with coverage of all context and some ignore it completely. Context awareness also incorporates sentiment intensity, as adjectives can increase or decrease sentiment intensity. Therefore state of the art sentiment analysis need to have some kind of context awareness[12].

2.4.1 Linguistic inquiry and word count

Linguistic inquiry and word count (LIWC) is a dictionary based text analysis software, first published in 2001[16]. One of the big players in sentiment analysis has been LIWC, which tallies up the words of text data and in accordance to a dictionary determines the valence, amount of used words and other factors for the text[16].



Filename	Segment	WC	Analytic	Clout	Authentic	Tone	WPS	Sixltr	Dic	function	pronoun	ppron	i	we
10.txt	1	559	89.83	72.84	34.47	52.62	25.41	19.86	81.57	47.76	11.81	8.77	3.04	0.00
11.txt	1	120	94.98	93.30	1.28	72.57	30.00	25.83	71.67	43.33	10.00	3.33	0.00	0.83
12.txt	1	202	97.25	78.59	10.18	43.84	28.86	18.81	78.22	46.53	10.89	7.43	0.99	1.49
13.txt	1	57	99.00	70.08	7.32	99.00	28.50	38.60	75.44	36.84	5.26	1.75	0.00	0.00
14.txt	1	54	90.85	71.09	2.56	98.27	18.00	33.33	64.81	38.89	11.11	3.70	0.00	0.00
15.txt	1	1027	52.10	68.67	17.05	30.72	17.41	19.57	84.91	56.67	15.09	7.01	2.14	0.49
16.txt	1	305	89.91	84.53	49.89	63.06	16.94	22.30	75.74	46.89	10.49	7.54	1.64	4.26
17.txt	1	271	92.46	76.97	29.17	73.95	22.58	23.99	77.12	46.86	11.44	5.54	1.11	2.58
18.txt	1	326	97.50	64.37	22.99	60.66	21.73	26.99	71.47	41.10	7.36	3.99	0.92	0.31
19.txt	1	296	85.96	68.18	5.66	44.31	24.67	21.62	76.35	45.61	10.47	5.07	1.35	0.00
20.txt	1	898	83.03	75.86	31.76	61.67	18.33	19.04	80.73	50.11	12.69	8.02	2.78	0.22
21.txt	1	869	84.63	68.14	27.94	51.62	17.73	20.83	75.37	44.07	8.98	3.68	0.46	0.81
22.txt	1	152	96.69	62.89	45.97	37.41	16.89	24.34	76.32	40.79	7.89	0.66	0.00	0.00

Example of LIWC collected data [16]

LIWC has been prolifically and successfully used in many interesting research projects over the years. In 2010[21] researchers used LIWC to predict the outcome of the

German federal election. This was done by analysing more than 100,000 Twitter messages, tallying up every mention of a political party and using the sentiment of the tweet to get the person's political affiliation[21].
Despite it's success LIWC is a paid program and quite old. LIWC effectiveness is quite dependent on fine prepossessing of the used data. It also does not take into consideration context or intensity. Furthermore it requires comprehensive understood dictionary and as such is out of the scope of the project.

2.4.2 Valence Aware Dictionary for sEntiment Reasoning

Valence Aware Dictionary for sEntiment Reasoning (VADER) is a simple rule-based model for general sentiment analysis introduced by Hutto and Gilbert in 2014[6]. VADER were created with the sentiment structure of social media in mind and in the area been tested to outperform previous state of the art applications. Compared to LIWC VADER includes consideration for sentiment intensity and VADER accounts for many symbols and expressions previous sentiment analysis simply ignored such as acronyms, initialisms, emoticons, or slang, which can be important factors in sentiment analysis[6].
First a new dictionary were created using information from other dictionaries, this was needed in order to encompass the new features used in text such as emoticons and abbreviations. Then the new dictionary were tweaked and modified using human workers. In addtion they took large batches of tweets extracted the sentiment using know methods, then scrutinized the tweets by individually evaluating them. With this approach they found 5 rules where small changes in a sentence would create different senitment, such as saying "ins't" before great and using capitalized words and punctiation[6].
For the challenge of context VADER uses publicly available python package made by Kathuria Pulkit which uses word sense disambiguation. The author themselves grant that context analysis could be improved, yet with the package they achieved good test results[6].
At last they tested their new method against 11 other established sentiment analysis

methods and outperformed them all. This proven effectiveness in addition to their code being publicly available to use, makes VADER a perfect method for sentiment analysis for the purposes of this paper.

2.5 Dimension reduction

Often data has many more dimension than are possible to visualize in a 2D or 3D graph so humans can understand and analyze it. Certainly plotting 2 dimension on a graph is possible to do even if the data has thousands of data points, the problem is that 2 dimensions out of a complex dimensional data often does not represent the data as a whole. The problem is then how to reduce the dimensions, while retaining data that can summarize the overall data. Several techniques lies in actively removing data as to reduce dimension, figuring out what data is repetitive or low impact, removing dimensions with missing values, removing dimension below or above a certain threshold when comparing to variance and correlation. In this section the most common and best fitted for the purposes of this paper dimension reduction methods will be explored.

2.5.1 Principal Component Analysis

Principal Component Analysis(PCA) is a quite popular dimension reduction technique, which has been around since the 1930s and recently gained new relevance with the rise of big data and machine learning, often used to extract features from large data set. PCA reduces any amount of dimensions down to another smaller number of dimensions called principal components. The principal components are ordered in such a way that the first component determines most of the data variance and the subsequent components less and less in order of their number[23].
PCA is linear combination of the values. In order to calculate PCA the mean is found for every dimension, then the covariance matrix of the whole dataset is cacluated. From there the eigenvectors can be found, the eigenvalues then determine wich eigenevector to put as first column in a matrix the length of the desired amount of principal components each column is then the principal components[24][5].

PCA work in such a way that all the points are mapped to a single line, the line the data is mapped to is the one that best symbolises the data's overall structure. This linear structure however means that data points that could be otherwise be lying far apart could show up together, this is the problem with PCA. If the data is too complex PCA will not always be able to show the data points separated correctly. In order to visualize more complex data a non linear approach might be needed. However the PCA is so reliable than even in very complex data, it often still can separate the data into some resemblance of classes even if not all desired classes are present. Furthermore PCA unlike more complex algorithms are computationally very light and can take seconds to calculate the principal components even for high dimensional data with more than half a million points[24][5].

2.5.2 t-Distributed Stochastic Neighbor Embedding

T-distributed stochastic neighbor embedding or t-SNE is another popular dimension reduction method. T-SNE in opposition to PCA is a nonlinear dimension reduction method. T-SNE was first developed in 2008 by Maaten and Hinton as a variation of stochastic neighbor embedding(SNE) and has since become widespread in machine learning. This is because the produced data are significantly better for visualizing the data structure of large data sets than other widespread dimension reduction methods [13]. This could be seen in the fact t-SNE were able to visually separate the data for the MNIST digit dataset into their respective digits, were no other previous method could. In order to understand t-SNE it is important to understand stochastic neighbor embedding (SNE), as t-SNE differ only by having the embedded points move in accordance to a t-distributed curve instead of normal distributed one. Stochastic means random, neighbor refers to closely related points through euclidean distance in any dimensional space and embedding refers to the fact of taking one point to a lower dimensional space. SNE creates a random lower dimensional point for each data point, then over iterations it moves these points a little away or towards the other points depending on the relations of the original points, e.g. if the original points are close the new points move closer and opposite if the originals are further apart[13].

SNE will measure the euclidean distance to every other point, these distances is then converted to conditional probabilities that represent similarities probabilities, which is a fancy way of saying a number from 0 to 1 that determines how similar 2 points are to each other. In order to get the distance down to a scale from 0 to 1 the points are mapped to a Gaussian curve centered at the first point. This is done for both the original points and their lower dimensional counter parts. Then in order to relate the 2 distributions of probabilities he Kullback-Leibler diverge[13] is used. The summ of Kullback-Leibler diverge is minimized using gradient descent, where the cost function designed in such a way that the cost increases rapidly for far divided points, this means in turn that the local structure of closely related points will be maintained between the 2 probability maps[13].
The width of the Gaussian curve used for the original points is found through a binary search determined by a perplexity parameter. The relation is such that the perplexity directly determines the amount of neighbours each point has. This means the perplexity should be adjusted depending on the sparsity of the data points, are the points sparsely separated the perplexity should be higher and the perplexity should be lower if the points are clustered closer together. Furthermore the size of the dataset should be considered as well, normal ranges of perplexity according to Maaten and Hinton are from 5-50, yet other numbers have been recommended by google brain employee Wattenberg who goes up to even a 100[22][13].
In order to upgrade to t-SNE as mentioned earlier t-distributed curve most be used instead of a Gaussian.

2.5.3 Reading t-SNE

The nature of t-SNE is such that if not understood, used or read correctly then their might be concluded to be patterns or relations where there in reality is none or otherwise patterns and relations that exist might no be seen[22]. This is because t-SNE is highly reliant on its hyper-parameters, such as perplexity and step count. too few steps might mean that the algorithm never converges and the points are not moved correctly together with their neighbours. If this is the case their might still seem to be

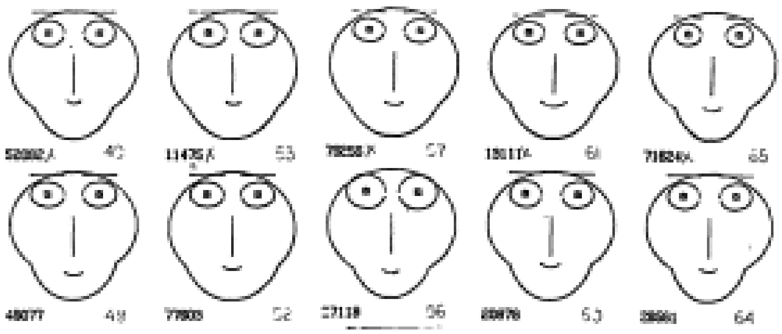
clustering and patterns, yet they are not good visualizations of the whole. This might mean that the step count should always be set high, and as perfect visualization goes that is correct as when the algorithm converges extra steps do not matter, yet t-SNE is not a light algorithm and extra steps might mean hours of extra computational time, if used on high dimensional data it is made for, and as the perfect hyper parameters is hard to know without trial and error setting the steps too high might lead to much wasted research time and computational time. If too low a perplexity is chosen data points within clear clusters in the original data might be separated away from each other as they no longer is seen to be relatively close together and as t-SNE does not care how far away a point is to another as long as they are at the edge of the t-distributed curve, close points might move just as much away from each other as far away points meaning otherwise clear classes are scattered in the lower dimensional plot. On the opposite side a too high perplexity might lead to clear data clusters to be scrambled together as all points are seen to be neighbours even if far away and the resulting map will not accurately depict the higher dimensional data[22].
T-SNE also has other pit holes to lookout for even if the perfect hyper parameters is found. This is because t-SNE might divide up otherwise non related points into clusters if there is no real relation in the data or if the data is dominated by perceivable random data. Furthermore even if the points are highly random, the resulting points will be gathered in clouds with higher density in the middle. suggesting that the data might be normal distributed where no such distribution is present. Also seemingly complex patterns can occur where points cluster together in way that might seem significant here it is important to remember such patterns may form in completely random data. last it is important to note that t-SNE, if used on the same data multiple times will not create the same low dimensional plot. Overall clusteres and such will be preserved if the data has clear enough classes, they might just be position shifted or the entire plot rotated. If the data is random enough totally different looking plots might happen after each run of t-SNE, otherwise the end plot will useally just be rotated or ordered differently while keeping the same clear clusters. This variance in the end plot is important to take note of, it is because as mentioned earlier the length between the dimension reduced points

does not matter only their relative position to other points[22].

2.6 State of the art

2.6.1 Chernoff faces

In a not so recent article from 1973 reseacher Chernoff[3]. explored data visualization of high dimensional data in the form of faces. This abstract yet concrete data visualization still has an impact today. Taking advantage of the human perception bias towards faces, makes it easier for the people reading the visualization to grasp small variations and tendencies in larger data sets[3].

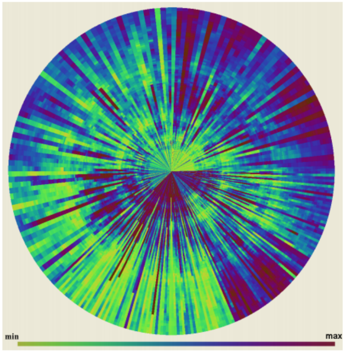


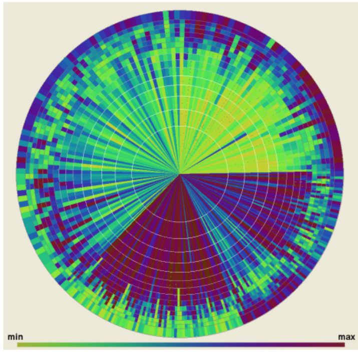
Example of Chernoff faces [3]

The concept of data usually displayed as points on coordinate system, as a spectrogram, a graph or other concrete visualization, suddenly displayed in new abstract yet familiar way, is testament to data visualization need for different approaches. This has to be taken into account for the purposes of this paper.

2.6.2 Scalable Pixel based Visual Data Exploration

In data analysis it is important to understand the data, this understanding often a staged process, of analysing data on different processing levels. When visualizing the data afterwards the problem occurs that the display of the data is not comprehensive and missing important tendencies lost in previous stages. one solution is proposed by Keim, Schneidewind and Sips[8], to create a scalable multi-resolution visualization.





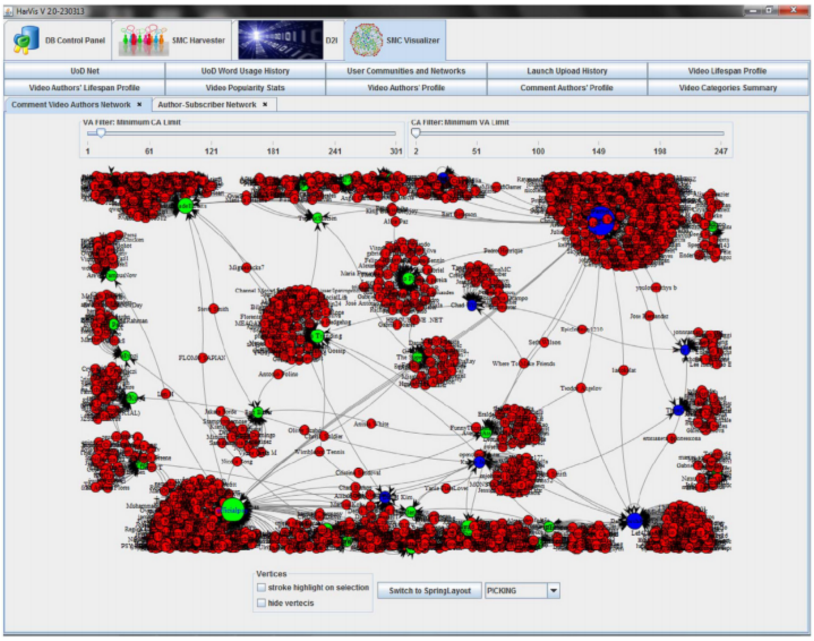
CircleView[7] visualization and Muliti resolution visualization[8]

To do this the researchers created a method which divided the data up into a hierarchical layers, where the first layer had the most important data, while having data point in the layer relating to multiple data-points in a lower layer, so to show all the data at the same time compressing less impotant data[8]. Using the Circleview[7] method they plotted their data in a know way, while displaying more data with the same method.

The idea of having data of different level visualized together in single, but coherent plot is interesting and indeed is part of the inspiration for the puposes of this paper.

2.6.3 Harvis

Harvis is a framework for topic analysis targeting YouTube content. It was first presented in 2016. Harvis include a unique node system for visualizing complex connected data from Youtube. The resulting node network creates a visualization, much more layered than any traditional method can do[1].



Harvis node visualization[1]

Harvis were a promising framework, but were limited by data acquisition which have proved cumbersome to collect from Youtube[1]. In any case the papers node system is an interesting and novel way of visualizing data which can be taken into consideration for the purposes of this paper.

2.6.4 CosMovis

CosMovis present a framework for a node based recommendation system based on sentiment analysis with added graphics for cognitive interpretation. One of the problems the paper tackles is the random positioning created by force-directed layout networks, where only proximity of nodes are relevant[1].

Harvis node visualization[1]

They solve this issue using heat maps created with sentiment. The relevance to this paper is their final visualization. The use of data and more concrete images together is interesting.

2.7 Problem Statement

From this analysis following problem statements can be formulated.

Ideally analysing game reviews using NLP, relationships between games can be established.

From that following hypothesis will be tested.

Creating an abstract way of visualizing complex data can lead to before unseen relationships in the data.

3 Methods

In this section the method of testing the proposed system and the elements of the system will be described.

3.1 The Test

This section will describe the elements and the purpose of the test. The data chosen for this experiment was steam game reviews, it is a data set containing reviews from 48 different games, were data from 9 of the games were used. The data were used to create abstract images.

The goal of the test is to asses the created images and see if they are similar enough for grouping. The test is taken in the browser in the environment of the participant's own choosing, the test conductor will not be present.

The participant is presented with grey screen with several images. Each image can be picked up using the mouse and placed anywhere else in the screen. Similar to familiar drag and drop methods widely used in many popular programs. When a picture is moved groups are made, text is shown in front of the image describing its current group and how many images belong to the group, this is to ensure the participant is certain the pictures are grouped together correctly. The grouping are not limited in any way and could be any number of groupings including all in one or all separated, but the participant is not able to continue the test if they have not moved at least 1 picture and by effect made at least 1 group.

The participants is prompted to group the images in any way they see fit. When the participant has decided the pictures are grouped as they want, they click the large button at the bottom of the screen. Then they are zoomed in on the first of the groupings and asked to look at the pictures. They can look at the pictures any duration determined by themselves with no time restraints. Then when they have looked at the group they are asked to describe the images in the group and why they grouped them together, this is done directly in a text box which is part of the environment. Then the next group is assessed, until all the groups have been described and the test is then over.

The data saved from the test is the descriptions of each group and the associated pictures in the group. With that data, the pictures can be evaluated in several ways.

3.2 Target group and Sampling

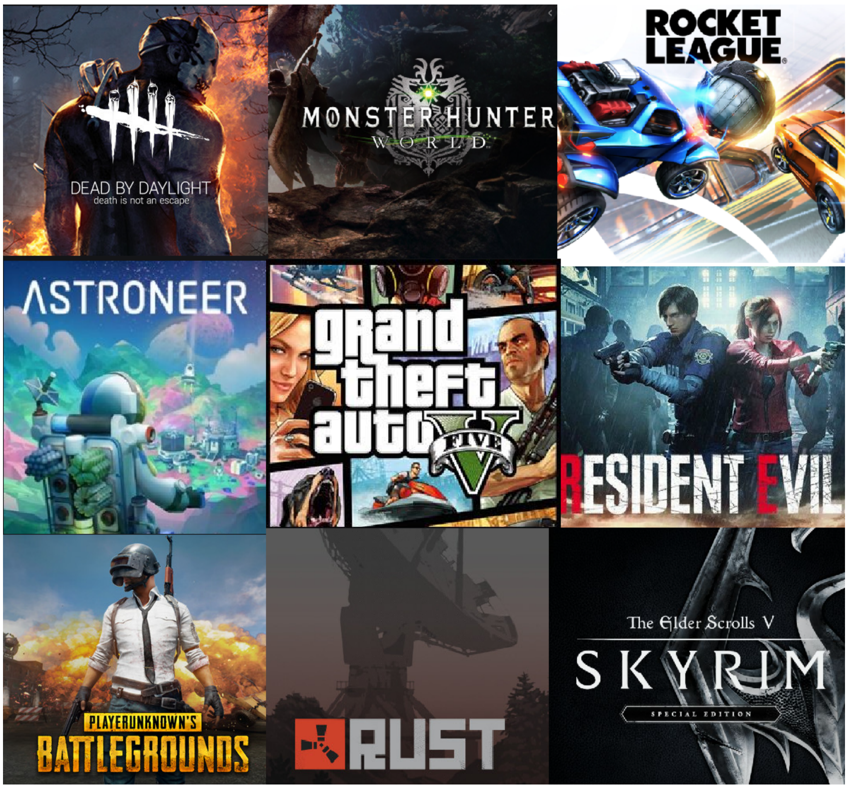
This section will describe the target group and the sampling of the test participants. While often several experiments require a quite strict target group, the nature of this test, require a target group as broad as possible. This is because visualizing data in an abstract way can potentially benefit an entire work chain populated not only by data analyst and other such technical oriented personnel. Therefore the target group is limited to any person of a working age from 18 and up. The sampling used were convenience sampling with as broad an audience as possible. The test were created so that it could be taken online broadening the pool of potential participant.

4 Results

In this section the quantitative and qualitative results of the test will be described, any significant data will be described in further detail.Then the results from the preliminary data exploration will be summarized as well. Any conclusions and assumptions derived from the results will be described in the evaluation chapter for any findings and in the discussions will asses biases and improvements.

4.1 Test Results

The test had 19 participants who divided the images into 2.84 groups each. The following nine games were used for the images, Dead by Daylight(DD), MONSTER HUNTER: WORLD (MHW), Rocket League®(RL), ASTRONEER(Astroneer), Grand Theft Auto V (GTAV), RESIDENT EVIL 2 / BIOHAZARD RE:2(REB2), PLAYERUNKNOWN'S BATTLEGROUNDS(PB), Rust and The Elder Scrolls V: Skyrim Special Edition (Skyrim). every game used had more than 1000 reviews.

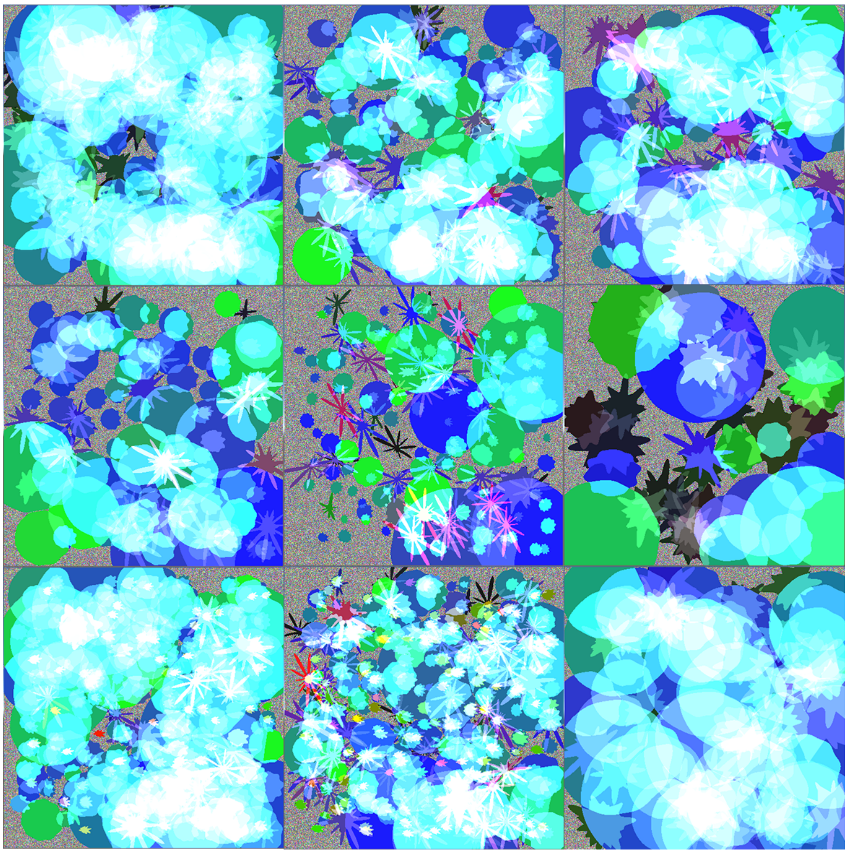


Snippets of each game's promotional poster, ordered as they are mentioned

Name	Number
Dead by Daylight	22221
MONSTER HUNTER: WORLD	18412
Rocket League®	67907
ASTRONEER	2661
Grand Theft Auto V	99956
RESIDENT EVIL 2 / BIOHAZARD RE:2	1385
PLAYERUNKNOWN'S BATTLEGROUNDS	145685
Rust	71088
The Elder Scrolls V: Skyrim Special Edition	1473

The games used and the number of reviews each game had

Each game had a abstract visualization made created by several splotches of color, in differing shapes. These were what the participants saw. The images can be seen on the picture below.



Images of the games in order from left to right then top to bottom: ASTRONEER, Dead by Daylight ,Grand Theft Auto V, MONSTER HUNTER: WORLD, PLAYERUNKNOWN'S BATTLEGROUNDS, RESIDENT EVIL 2 / BIOHAZARD RE:2, Rocket League®, Rust and The Elder Scrolls V: Skyrim Special Edition

The games were grouped together several times, every grouping were recorded. For each pair of games grouped together the result were tallied up and a table over the groupings were made.

	Dead by Daylight	Monster Hunter: World	Rocket League®	ASTRO-NEER	Grand Theft Auto V	RESIDENT EVIL 2 / BIOHAZ-ARD RE:2	PLAYERUN-KNOWN'S BATTLE-GROUNDS	Rust	The Elder Scrolls V: Skyrim
Dead by Daylight	0	11	7	6	9	8	11	14	6
Monster Hunter: World	11	0	7	6	9	9	14	10	6
Rocket League®	7	7	0	18	9	6	7	6	17
ASTRO-NEER	6	6	18	0	10	7	6	7	18
Grand Theft Auto V	9	9	9	10	0	9	6	9	11
RESIDENT EVIL 2 / BIOHAZ-ARD RE:2	8	9	6	7	9	0	9	8	7
PLAYERUN-KNOWN'S BATTLE-GROUNDS	11	14	7	6	6	9	0	11	6
Rust	14	10	6	7	9	8	11	0	7
The Elder Scrolls V: Skyrim	6	6	17	18	11	7	6	7	0

All the games and the amount of times they were grouped together

When looking at the data it is clear several games have very similar images as they have been grouped together almost every time. Most significant is Astroneer and Skyrim which were grouped together a whole 18 times, as well as Astroneer and RL with the same amount of groupings. Not only are they grouped together separately all 3 games have been grouped together 17 times and as such clearly show a tendency.
These 3 games have widely different amount of reviews, with Astroneer and Skyrim

having similiar numbers, RL have more then 10 times theirs combined. This means the image creation method does not seem to be biased towards amount of reviews and it is the content of the reviews or other factor that matter.

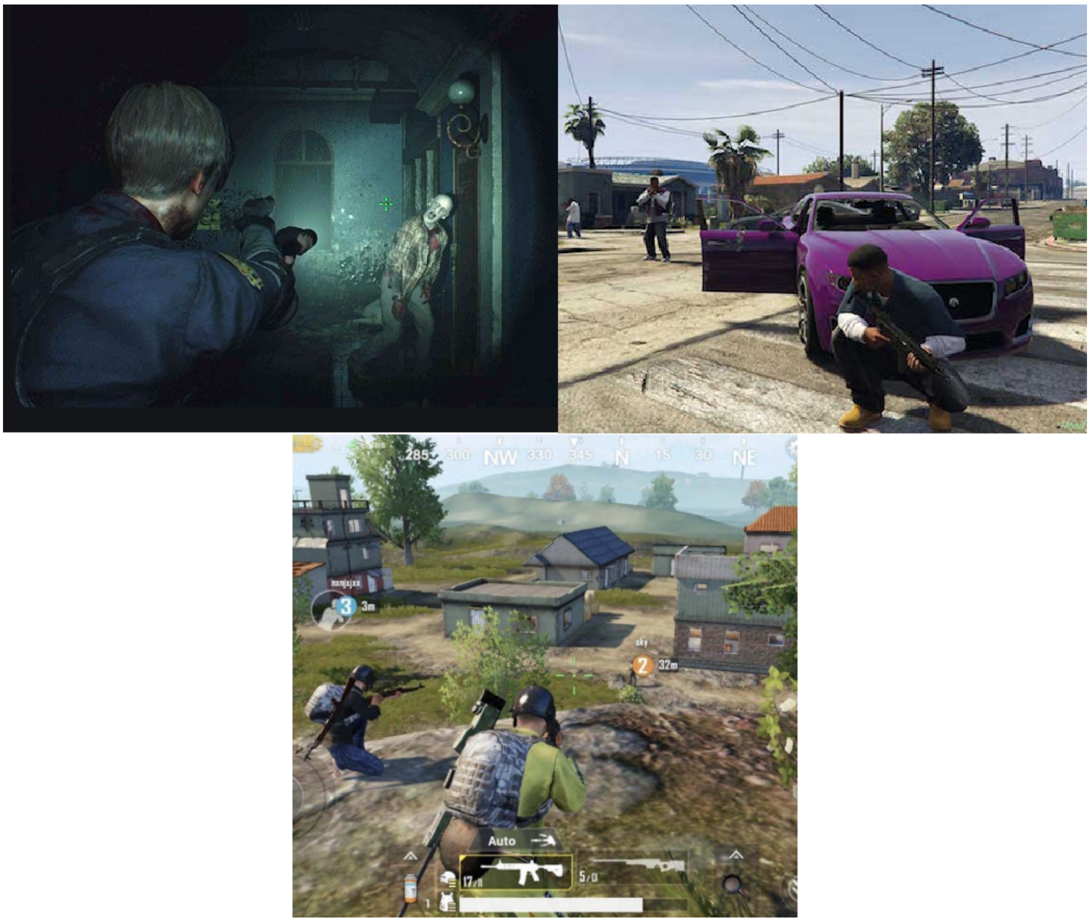
Images of Astroneer, RL and Skyrim in order

Another factor than similar groupings, is the amount of times a game have been grouped in group by itself and as such would standout to not follow the trend of patterns set in the other images.

Game	Single groupings
MONSTER HUNTER: WORLD	0
Rocket League®	0
Rust	2
PLAYERUNKNOWN'S BATTLEGROUNDS	1
Dead by Daylight	1
ASTRONEER	0
Grand Theft Auto V	3
Skyrim	0
RESIDENT EVIL 2 / BIOHAZARD RE:2	7

Several games have not been grouped by itself a single time, while other games such as REB2 has been grouped alone seven times. When looking at the images REB2 is the only game with visible dark areas and as such does stand out. This also lines up with peoples reasoning for grouping it alone, from the questioning.

One thing to note is the amount of reviews REB2 has which is the most similar to Skyrim which never stood by itself. Another important note is the most common pairing with REB2 is PB which has 1 single grouping, MHW which has no single groupings and GTAV who has the second highest of single groupings with 3.



Images of REB2, GTAV and PB in order

Another game with relatively high single grouping is Rust with 2, which is mostly grouped with DD who has only 1 single grouping.

This suggest that images might have been grouped together, if they had no obvious space to go.

In addition to the quantitative data created by the groupings, the participants also described their groups yielding qualitative data. Most of the data is none specific lending most to phrases like "they look alike", other data does yield a little insight to the participants criteria for grouping.

When looking at the trinity grouped together 17 times of Skyrim, RL and Astroneer. Words like large, big and bright circles are repeated.

When looking at a game like REB2 which had the most single groupings and only looking on the data from those groupings, words like gray and black colors in the

background are predominant. When looking at data where it was grouped with other images, the data suggest the participants had deiced on a single factor for their grouping choices such as the amount of cyan, ignoring the other factors.
The words color were used 10 times, cyan or blue 11 times, while words like pattern, surface, shape and similar were no where near as predominant.
The qualitative data also reveals, at least 2 people have grouped the image not because their likeness, but instead they were trying to create a pattern with the images themselves. Such as ordering the images in chequered formation or other.
Several participants didn't write anything at all. Where 4 of these in addition to not writing grouped all the images in one large group.

4.2 Data Exploration

One of the first objectives when dealing with large data sets is to explore them. In this section some of the initial exploratory results will be described. These result are needed for creating and evaluating the data visualization.

The data used had initially 6 features, which are listed below1.

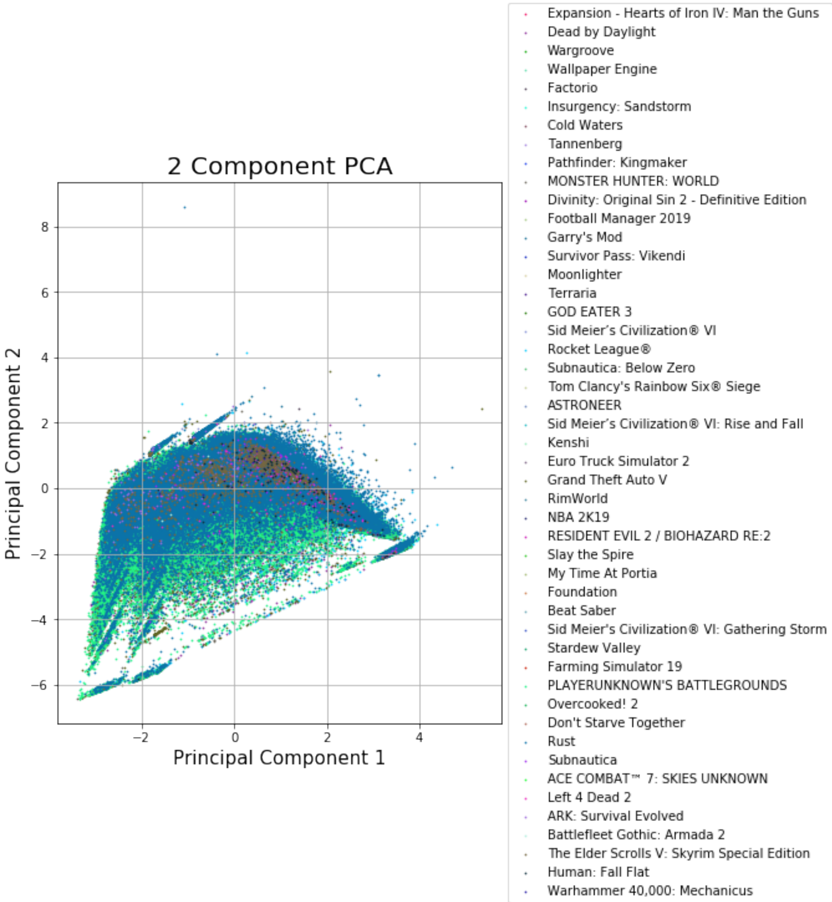
Amount Funny	Amount Helpful	Hours Played	Early Access	Recom-mended	Text review

Table 1 The 6 features of the exploratory data

Amount funny or helpful denotes how many people found the specific review such and hours played denotes how many hours the reviewer has played the game before writing the review. Early access denoted whether the game is in early access at the point of the review and Recommended denotes whether the reviewer would recommend the game.

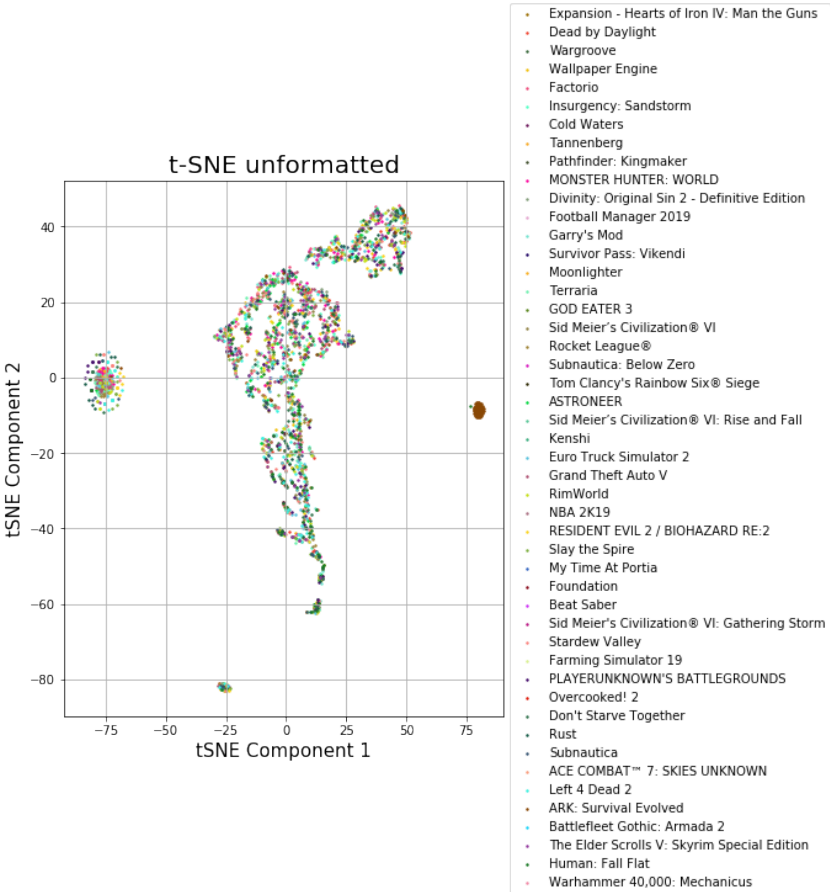
For initial exploration these features seemed interesting and were added to any feature vectors extracted from the text review, when doing NLP.

The firsy interesting plot to show is all the features, with 4 VADER sentiment values instead of the text review. The features were standardized and in the case of string data like recommended and early access they were set to 0 or 1 respectively before standardization.



Scatterplot of 2 component PCA using all the data, text converted into values using Vader, colorized depending on game

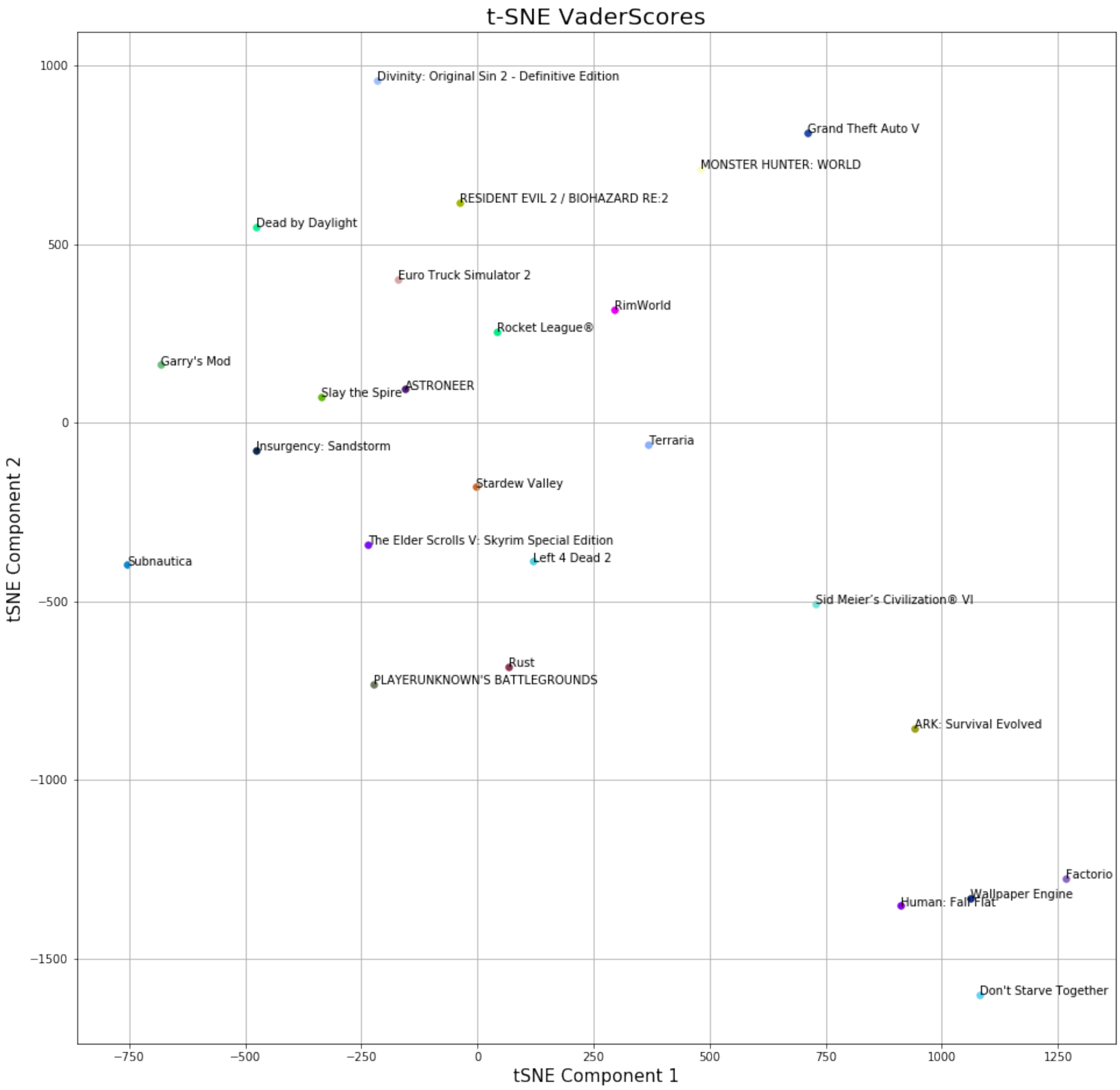
From this exploratory plot it is clear that reviews from one single game does not seem to cluster together as all the game reviews seem to be very randomly clustered within the rather complex shape. Of course it is clear the data is dominated by points of a few games which has an over representation in the fact of number of reviews.
Several more plots and variation were made with the same features including t-SNE of multiple perplexity and coloration of data depending on recommendation instead of games and 3D plotting. All with the same conclusion.
The games were plotted using only vader scores and still no clear groupings were found. Then plotted using only positive score, plotted with only a few hundred reviews for each and other approaches. All images created can be found in associated GitHub link.



50 perplexity t-SNE of 100 positive reviews from each game using only Vader scores as features

To only show a single plot of many in the exploratory analysis. Plot of t-SNE of isolated

Vader features on only 100 positive reviews from each game. This plot actually has a clear separation of 1 game, the brown smudge to the right. The game is ARK: Survival Evolved and had only very few reviews in the data set, what was more the game had undergone a reviewing bombing due to server problems, meaning the amount of positive reviews were small and likely not representative of game as a whole, even when positive. Then in order to not be overloaded by information each game were drawn as a single dot in the plot. Again following the trend set by the prior plots every game clumped together. Yet t-SNE with very low perplexity yielded some useful plots.



2 perplexity t-SNE of every game as long vector the first 200 reviews using Vader scores as features

This plot 4.16 of the Vader score show some relation between games could be seen such as ASTRONEER AND Slay the spire seemed to have similar sentiment in their reviews. Yet with so low perplexity and only a fraction of the data taken into account any conclusions are hard to make.

Another plot were made of with Doc2vec scores as features, yet it wasn't useful.

5 Implementation

The section will describe the code, logic and tools used for the setting up of the test, processing the data and creating the visualizations. This will ensure that similar implementation could be created and the test be replicated. While all the code can be found on github for the test environment for the jupyter code. Some logic and implementation has been described in more detail.
GitHub link:
`https://github.com/sennepsild/Master-Thesis-Data-Visualization.git`
In order to dimension reduce, visualize and format the raw data, python 3.7 was used. For this the code was written in Jupyter Notebook 6.0.1, which allowed for dividing and running the code in sections all within the same document.
In order to apply PCA, t-SNE and standardize the data the implementations provided by the Scikit-learn library was used.
In order to do VADER sentiment analysis the code provided by C. J. Hutto[6] GitHub which was published along the paper under the MIT license was used.
In order to do Doc2Vec paragraph to vector conversion, implementation from the Gensim library was used [18].
The plotting was done using Matplotlib and the data was formatted using Pandas.
The test environment was made using game engine Unity version 2019.2.9f1 and hosted online using unity's web game share site unity connect.
The code made to create the abstract visualizations used for the test was initially created in shadertoy for prototyping, then converted to Unity shader code for use in the final test.

5.1 Data processing for the visualization

First all the text data was tokenized, replacing every capital character with lowercase, removing any symbols and replacing spaces by ordering every word as its own string, this was done using Gensim's preprocess function. In addition to tokenization every sentence were tagged with a unique number, the number is used for initial classification

in the Doc2vec network.
The entire tokenized dataset is then used for the Doc2Vec training. Using Gensim's Doc2Vec model. Several Doc2vec settings were tried, hierarchical softmax were used as supposed to noise words. Vector size were set to 50, no minimum of single word repetition were used. trained with 4 workers and a window of 2. When the model had been trained the weights were saved, so every game could be evaluated with the exact same model.
Then the text data were divided into individual games and any game with less that 1000 reviews were discarded.The game with lowest amount of reviews used had 1385 and the one with the most had 145685. Each game's doc2vec vectors were then dimension reduced using t-SNE, several perplexity settings were tried out before a perplexity of 20 were used. The number of iterations used were 3000, it consistently converged the algorithm. Then the dimension reduced data were classified using DBSCAN, with epsilon of 2.2 and a varying minimum sample depending on the number of reviews for each game, in order to get a most uniform distribution over all the data. Minimum sample used were 20 for the highest number of review games 10 for the average amount and 5 for the games with only a few thousands reviews.
All the text were evaluated for sentiment without prepossessing using Vader, as Vader takes into account the use of capital letters and symbols. For each class created with DBSCAN, the average sentiment score provided by Vader were calculated for that class. These scores were then saved in csv file to be read in another program to create the basis of the abstract images.

5.2 The Shader

In order to create the visual from the processed data a shader were created. Shaders use the GPU and are designed to create visuals and so were an ideal choice for the implementation. The shader's purpose were to convert the classes created by DBSCAN into visuals, in this case every class related to a blob. To create a blob, first a radius were determined depending on the size of the class, with a radius a circle could be create by checking every pixels distance to the center. TO modifiy the circle atan were used, which gives a number from negative pi to pi depending on the uv postion around the center, this is then used to add to radius so a sin wave distortion could be created around the edge, creating almost a flower shape structure. The power of the distortion were then change depending on the vader score, distorting negative classes more than positive. In addition to strength of the distortion the frequency of the wave were increased by multiplying atan, also in accordance to the vader score. In order to create seamless transition the absolute value of atan were used. To not get very distorted blobs to fill to much the entire scale of distorted blobs were decreased accordingly. Furthermore to not have uniform distortion the strength were multiplied by the atan value as well, with a random offset for each blob. In order to create some hard edges for a Kiki effect, two triangles were layered on top of each other. To create a triangle atan were used again, this time with the floor function. An offset atan divided by the fraction of 2 Pi, depending on the number of edges, then timed by the same double pi fraction -atan, useing floor on the result, create a hardedged shape with amount of edges equal to the fraction of double pi. This shape could then be scaled by multiplying by any number, this was again done in accordance to the size of the blob. Ensuring that the positive blobs would not actually show the triangles by the virtue of obscuring them.

```
d = cos(floor(.5+a/r)*r-a)*length(uv*3./size);
```

Hard edge shape logic, were a is atan and r is fractional double pi

In any area without a blob each the color channels were assigned a random number each between 0.3 and 1. To create random numbers used in any part of the shader hashing were used.

6 Evaluation

This section will describe the projects findings.
In the exploratory analysis of the data, relating games proved hard to accomplish. This were because of the sheer amount of data and overall possibilities. Using every Feature available did not yield any useful plots, as the data for each separate game seemed too similar. Focusing on the text data alone proved to be the best option. Plotting the games separately or as single dots created the most useful plots. Furthermore separating sentiment and topic vectors when analysing the text reviews proved useful as well, as they created too much noise put together.
The knowledge of exploration were then put into creating a way of visualizing the topic and sentiment in way that they wouldn't compete among each other. This was done and using classes identified by DBSCAN on Doc2vec data, dimension reduced by t-SNE. These classes where then related into visuals using additional VADER sentiment analysis. Bouba/Kiki and other gestalt theory were taken into use.
then test environment were set up to see if any useful information could be derived from the new way of visualizing the data.
The test confirmed that the data were hard to separate, as many of the games were related consistently with at least 6 pairings to every other game.
Yet pairings of games were found as several games pairing stood out to be consistent. This is in stark contrast to exploratory analysis where every game seemed to relate to a new game depending on the approach or were indistinguishable from other games.
It has to be stressed that the goal were not to relate games in any distinct way and such any observations that can be drawn are useful for setting up new test in order to confirm relations between the games.
As such the relations created in the results will be evaluated on several level, such as gameplay, goal of the game, graphics, audience, demographic of the player and other trends that can only be shown through reviews.

Images of Astroneer, Rocket League (RL) and Skyrim in order

Here the attention falls on the consistently grouped games Astroneer, RL and Skyrim. On the surface none of these games are similar, but when considering the before mentioned deeper levels, they suddenly have commonalities. Astroneer and RL is both indie games and as such have to rely on unique innovations to stand out and become popular as they have done. People who play these kind of games and as proxy their reviewers, might be more cautious in their game choices, not falling for hype of the next big franchise release, instead taking it upon themselves to find the games they are looking for. Skyrim stands out as it does not on the surface follow the same pattern, as it is a big franchise game. Yet Skyrim does have similar appeal in the fact of player tenaciousness, Skyrim is almost 10 years old and still have thousands of consistent players, this is a testament to some unique factor that the players have not found in another game, which is the same thing games like Rl and Astroneer is sold upon.

Image of Resident Rvil 2 Biohazard re:2 (REB2)

Applying same logic to REB2 there is nuances unseen, which might need exploration. Looking over the games with regular goggles the closest game to REB2 would be DD which has same horror themes, dark color palette and overall objectives. Yet there is no apparent trend in the grouping of the 2 games. Here it might vital to understand the type of player for each of the games. REB2 is remake of an older game, a tried and true recipe for players who know what they and what they will get. While DD is a new spin on the genre, letting the monster be a player. People who goes for these games might overlap, but are indeed distinct.

Image of Dead by Daylight (DD)

These relations might seem weak and biased, because similarities are forced to be created. This is not necessarily untrue, but it proves that there is several more angles to be explored before dismissing the findings as incidental.

In addition to the new relation found between the games, the fact every person almost consistently made at least 3 groups suggest, that is is a lot easier to pair games using an abstract visual approach. As 3 clear separate groups were never found during any approach in the exploratory analysis. This finding with additional research to validify the findings could prove that the best way of exploring complex large data-sets is through abstract visualization, where each feature has a clear abstraction.

7 Discussion

This section will discuss the topics brought up in previous sections, as well as asses biases and explore potential fixes to any method used.

The overall test seemed efficient enough, it was very simple yet engaging. It still however could have used some tweaks, as 2 people were unsure how the test work a

different introduction could be needed.
There was a problem in the code such all the qualitative data were not recorded, overall the data sending could have used revisions. Test the test, is a poplar phrase very much applicable here.
One thing to note is the qualitative data, which seemed to trend towards people grouping the images according to color in Opposition to shape or pattern. This however could be due to the nature of the images, with no real patterns implemented and could be interpreted to have weak shape language. Therefore more and different visualizations should have been tested.

8 Conclusion

Finding similarities however vague between games using thousands of reviews about the games, does not seem like a fruitful task. Vader sentiment analysis alone of text reviews are too similar to draw concrete conclusion, even considering very different games of several genres. Doc2vec topic embedding vectors alone, also can not predict the kind of game or similarities between the games. However using Doc2vec looking at reviews of single games, several small classes can be found in the data, too many to have any meaningful impact to an observer using conventional methods. Using these classes as a input for creating abstract visualizations combined with sentiment analysis data however might yield more useful, as this were done with more success. However the finding of the visualizations require more explorations and data to be confirmed in any meaningful way, but lays ground for new research.

9 Future works

This section will describe how this project would progress if more time and resources were permitted. First more human test would need to take place, however useful the test has been it can only be improved with more participants.
The relations created by the groupings in the test would have to be explored in depth.

And new test set up to validate them.
Using the data visualization method on data with clear tendencies could be useful to see if the visualization reflect the data and in which way. Then several iteration would have to be made, depending on the new findings.
The way qualitative data was extracted and formulated could be improved. Now when there is a thread in what participants think about when being asked general questions about groups, the acquired knowledge could be used to formulate better and more direct questions useful for evaluation.
The visualization would need to be tested, so that every nuance of the shape and color is equally weighted.

References

[1] Ahmad, U., Zahid, A., Shoaib, M., and AlAmri, A. Harvis: An integrated social media content analysis framework for youtube platform. Information Systems 69 (2017), 25–39.

[2] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. A neural probabilistic language model. Journal of machine learning research 3, Feb (2003), 1137–1155.

[3] Chernoff, H. The use of faces to represent points in k-dimensional space graphically. Journal of the American statistical Association 68, 342 (1973), 361–368.

[4] Chowdhury, G. G. Natural language processing. Annual review of information science and technology 37, 1 (2003), 51–89.

[5] Dubey, A. The mathematics behind principal component analysis, Dec 2018.

[6] Hutto, C. J., and Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eighth international AAAI conference on weblogs and social media (2014).

[7] Keim, D. A., Schneidewind, J., and Sips, M. Circleview: a new approach for visualizing time-related multidimensional data sets. In Proceedings of the working conference on Advanced visual interfaces (2004), pp. 179–182.

[8] Keim, D. A., Schneidewind, J., and Sips, M. Scalable pixel based visual data exploration. In Visual Information Expert Workshop (2006), Springer, pp. 12–24.

[9] Köhler, W. Gestalt psychology (liveright, new york).

[10] Lau, J. H., and Baldwin, T. An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv preprint arXiv:1607.05368 (2016).

[11] Li, P., Liu, Y., Sun, M., Izuha, T., and Zhang, D. A neural reordering model for phrase-based translation. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (2014), pp. 1897–1907.

[12] Liu, B. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies 5, 1 (2012), 1–167.

[13] Maaten, L. v. d., and Hinton, G. Visualizing data using t-sne. Journal of machine learning research 9, Nov (2008), 2579–2605.

[14] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).

[15] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (2013), pp. 3111–3119.

[16] Pennebaker, J. W., Francis, M. E., and Booth, R. J. Linguistic inquiry and word count: Liwc 2001. Mahway: Lawrence Erlbaum Associates 71, 2001 (2001), 2001.

[17] Ramachandran, V. S., and Hubbard, E. M. Synaesthesia–a window into perception, thought and language. Journal of consciousness studies 8, 12 (2001), 3–34.

[18] Řehůřek, R., and Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks (Valletta, Malta, May 2010), ELRA, pp. 45–50. http://is.muni.cz/publication/884893/en.

[19] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. nature 323, 6088 (1986), 533–536.

[20] Todorovic, D. Gestalt principles. Scholarpedia 3, 12 (2008), 5345.

[21] Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. Predicting elections with twitter: What 140 characters reveal about political sentiment. In Fourth international AAAI conference on weblogs and social media (2010), Citeseer.

[22] Wattenberg, M., Viégas, F., and Johnson, I. How to use t-sne effectively. Distill (2016).

[23] Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. Chemometrics and intelligent laboratory systems 2, 1-3 (1987), 37–52.

[24] Zou, H., Hastie, T., and Tibshirani, R. Sparse principal component analysis. Journal of computational and graphical statistics 15, 2 (2006), 265–286.