Bielefeld University

Project Report

Title Clusters in Vector Space

Clustering of Artwork Titles via Word Embeddings

Max Harder, 2919411
max.harder@uni-bielefeld.de

Module 23-TXT-BaCL6 Project Module
230020 Projektseminar (S) (WiSe 2019/2020)

Mr Nikolai Ilinykh
nikolai.ilinykh@uni-bielefeld.de

March 28, 2020

Contents

Abstract

Assuming that artwork titles try to condense the content of the corresponding artworks, I apply clustering algorithms to word embeddings to group a set of artwork titles from *The Tate Collection* into emergent categories. First, I represent the semantic qualities of each title via word vectors. Then, I cluster similar titles in a three-dimensional vector space. Finally, I examine the qualities of each title cluster. Therefore I follow a two-track approach utilising word vectors trained on the data with Word2Vec as well as GloVe's pre-trained vectors, which are then clustered using the algorithms of three different clustering methods. Showing the limits of the algorithms applied, this process reveals that, on the whole, two clusters can be formed; one of them encompasses the titles of one specific artist.

Keywords: GloVe, Word2Vec, k-means, agglomerative clustering, DBSCAN

1 Introduction

Dividing data into distinct sets, clustering and classification are widely used techniques in the field of Natural Language Processing (NLP) to retrieve information from large amounts of text data. While classification processes entail assigning pre-defined labels to existing classes, clustering does not require prior knowledge of these classes. Being able to deal with unlabelled data, the unsupervised learning of clustering algorithms proves useful in cases in which labelled data is either not available or not affordable.

This report describes my approach of applying clustering algorithms to word embeddings, i.e. to the distributional representation of words in a space of dense, real-valued vectors (Corrêa Jr. et al. 2017), to group titles of a structured collection of artworks without pre-defined class labels into emergent categories. Determining common features in the collection and regrouping the data accordingly, I also identify recurring descriptors and patterns in titles of prestigious artworks. In the context of the dataset used for clustering artwork titles, *The Tate Collection*, which encompasses the metadata of about 70,000 artworks owned by the art museum network Tate, I assume that each title tries to condense the content of its corresponding artwork. To group the titles according to their commonalities, it is necessary to first represent the semantic qualities of each title as a vector. In the next step, similar titles can be clustered in a three-dimensional vector space. Finally, it is possible to examine the qualities of each title cluster and evaluate the results. Showing the limits of the algorithms applied, this process reveals that, on the whole, two larger clusters can be formed; one of them encompasses the titles of one specific artist.[1]

[1] The scripts of this project are available on request: max.harder@uni-bielefelde.de.

In a similar way, Choi and Kim (2019) exemplified statistical semantic models and word embedding techniques in the context of large-scale text analysis. The authors implemented an approach based on a Word2Vec model of documents' nouns which were clustered using the k-means method. They performed topic modelling on each of the clusters, utilising a bag-of-words model and latent Dirichlet allocation (LDA) to reveal latent topics clusters and their relationships.

Likewise, Butnaru and Ionescu (2017) developed an approach for text classification based on clustering word embeddings and inspired by the bag-of-visual-words model. They used Word2Vec's pre-trained word embeddings to represent each word in a collection of documents in a vector space and applied a k-means algorithm to the embeddings of each document to obtain equally sized clusters. The cluster centroids were then interpreted as super word vectors and put into the document's "bag of super word embeddings" to finally train a classification model with the frequencies of the designated super words.

This paper describes an exploratory approach of using Word2Vec and GloVe embeddings and k-means, agglomerative clustering, and DBSCAN algorithms for the task of clustering artwork titles. In the following section, I describe the dataset and its key characteristics in a more detailed way. The third section gives an overview of different approaches to represent words in NLP, focusing on the two distributional representations mentioned above. Moreover, it outlines the three clustering algorithms used in this project. The subsequent section provides the analysis of the dataset, based on the aforementioned methods. Finally, a discussion of the results and considerations for future work are included in the fifth section.

2 Dataset

Tate, also known as the National Gallery of British Art or Tate Gallery, is a network of four art museums in the United Kingdom. The dataset in a repository called *The Tate Collection*[2], last updated in October 2014, presents the metadata of around 70,000 artworks owned by Tate, as well as metadata for around 3,500 associated artists. In this project, I focused on the titles of artworks included in the collection, but also integrated additional information about artists and years of origin.

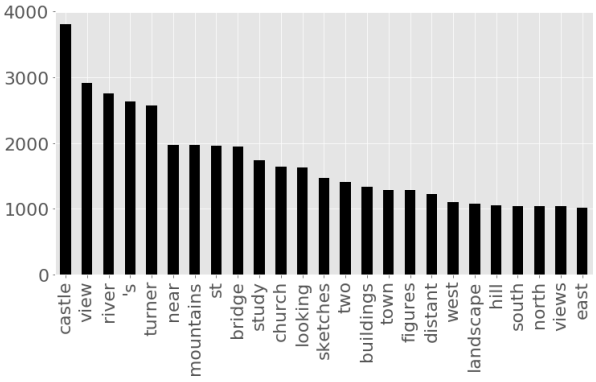

Figure 1: Frequencies of the 25 most common words (without stop words)

After a basic preprocessing and first analysis of the data, several key characteristics of the collection could be identified. Most importantly, the titles of about 13,000 included artworks are labelled unknown, thus the number of usable titles reduced to a maximum of less than 57,000. Moreover,

[2]The dataset is available for download on GitHub: https://github.com/tategallery/collection.

non-English words like the French "de" (519 occurrences) or the German "burg" (501 occurrences) indicate the presence of titles in multiple languages and underline the necessity to either identify and differentiate between titles in different languages or to use methods which can handle multilingual text data. Beyond that, titles of 3,188 different artists are part of the collection, the most frequent one being J. M. W. Turner, who has a share of 58.3 percent with 33,106 artworks. The works originated in 359 different years and cover a period from 1545 to 2012. Most of the artworks, in numbers 2,757, were created in 1819, almost exclusively by Turner. The year of origin is unknown in 2,814 cases.

On the whole, the 56,788 titles consist of 329,059 words, what gives an average length of 5.8 words per title. The vocabulary includes 22,421 words, out of which 49.0 percent are so called hapax legomena, words which occur only once in the whole collection. With 19,279 word types (lexemes), the type-token ratio of 5.9 percent indicates a very low lexical variation. Out of all tokens, 50.8 percent are nouns, 7.1 percent are adjectives, and 5.5 percent are verbs. In contrast to this, the proportion of prepositions and subordinating conjunctions and of determiners is very high, accounting for 27.5 percent altogether.

The application of a list-based stop word filter with 153 entries resulted in a total value of 31.0 percent stop words, which include 136 different types. Stop-word-filtered, the average title length reduced by a value of 1.8 to 4 words. In addition, 45 titles were filtered out, such as "This, That And The Other", as they consist only of stop words. Tagged as non-word are 55,183 tokens of 11,362 types, which means that 16.8 percent of the tokens and 50.7 percent of the vocabulary are (in its lemmatised form) not part of Unix' word corpus, which comprises more than 236,000 words used in the English language. To a large extent, the non-words are place names and numbers, but also numerous French, German, and Italian expressions. The 25 most frequent words, illustrated in Figure 1, encompass nouns describing landscapes, e.g. "castle", "river" or "mountains", words specifying locations, e.g. "near", "distant" or the four cardinal points, and expressions from the field of arts, e.g. "turner", "study" or "sketches".

3 Methodology

The following sections provide an outline of the two basic methods used for the clustering task: the representation of words through feature vectors with Word2Vec and GloVe and the subsequent algorithmic clustering of those word embeddings via k-means, agglomerative clustering, and DBSCAN.

3.1 Word Representation

Different approaches have been developed to represent words in NLP. Simple approaches like the dictionary lookup, one-hot encoding or term frequency–inverse document frequency (tf–idf) based distributional representations entail fixed word representations which are easy to use, but not only do they require a relatively large memory, they also fail to include word meanings into representations (Rakhmanberdieva 2018a). More advanced distributed word representations like Word2Vec or GloVe do this by representing

words as feature vectors (Rakhmanberdieva 2018b). Similar words are modelled in such a way that they are close to each other. This makes it possible to exploit semantic linear substructures through vector differences, for example, and to find nearest neighbours via Euclidean distance or cosine similarity.

3.1.1 Word2Vec and GloVe

The first approach, Word2Vec, published in 2013 and patented by Google, makes use of the skip-gram model, which represents words through their neighbours in a local context window, and tries to find word representations which can predict the context of a word (Mikolov et al. 2013; Rakhmanberdieva 2018b). In contrast to this, the GloVe (Global Vectors) model, which followed in 2014 as an open-source project at Stanford, works with overall co-occurrence statistics of words from a collection of texts (Pennington et al. 2014). With a count-based method, it constructs a high dimensional context matrix of co-occurrences and their conditional probabilities on which it applies a global log-bilinear regression model to capture linear substructures (ibid.). For both methods, pre-trained word vectors are available for download, but embeddings can also be trained on a separate corpus. Yet, Word2Vec and GloVe cannot provide vectors for out-of-vocabulary words, vectors for rare words can have a questionable validity, and multiple meanings of a single word cannot be represented in one embedding (Rakhmanberdieva 2019). Beyond that, the methods are not designed to handle multilingual data.[3]

3.1.2 Title Representation

For the task of representing artwork titles as feature vectors, each word in a title was first embedded in a vector space by means of Word2Vec's training algorithm as well as through GloVe's pre-trained word vectors. This allowed me to draw a comparison of the two approaches while gaining from their differing strengths: Training embeddings directly on the relevant data circumvents the problem of out-of-vocabulary words. In the case of rare words, however, one runs the risk of low-quality vectors with low informative value. This problem is bypassed with the pre-trained word vectors of GloVe, which include 400,000 word representations trained on English Wikipedia and newswire text data (Gigaword 5) from 2014 and 2011, respectively. To obtain the final title vectors, the word vectors of all words in a title were summed up and divided by the number of words in a title to calculate its average meaning. Inspired by Corrêa Jr. et al. (2017), the initial plan to use weighted embedding vectors, embeddings multiplied by the tf–idf of the word which the embedding represents, was rendered impossible by the fact that no meaningful tf–idf values could be calculated, as the artwork titles are not arranged in coherent documents.

3.2 Clustering

The process of clustering is a multi-faceted step spanning a variety of methods, which generate divergent results (Wang et al. 2017). The most common methods are partitioning, hierarchical, and density-based ones. On the

[3]For the results of popular language detectors, e.g. TextBlob or LanguageDetector, proved deficient in the case of the title's short text samples, this problem could not be solved completely.

whole, partitioning methods like k-means, k-medoids or CLARANS are good at forming spherically shaped clusters but they require the number of clusters to be specified in advance; most hierarchical methods, like BIRCH or Chameleon, avoid the problem of determining the number of clusters altogether but they generally fail to perform effectively when datasets are large; density-based methods can handle arbitrary shapes and some, e.g. DBSCAN or OPTICS, also sidestep the issue of choosing the optimal number of clusters. In the following sections, I sketch out the three clustering algorithms applied in this project, each belonging to one of the aforementioned methods: k-means (partitioning), agglomerative clustering (hierarchical), and DBSCAN (density-based). Then, I confront the problem of finding the correct number of clusters in a dataset and outline two methods which try to tackle it: the elbow and the silhouette method.

3.2.1 k-Means

The k-means algorithm aims to cluster data by partitioning samples into k clusters. With the number of clusters *(k)* as a required parameter, k-means initialising step is to randomly choose cluster centroids. By calculating mean values, it then assigns each sample to its nearest centroid. Thirdly, it defines new centroids based on the mean value of all samples of each centroid. From then on, the algorithm loops between the last two steps until the difference between the old and the new centroids is no longer significant (scikit-learn developers n.d.[a]). To address the issue of sub-optimal clustering caused by an unfavourable initialisation, the algorithm is often run several times, with the final clusters being the best output in terms of inertia (scikit-learn developers n.d.[b]).

Beyond that, scikit-learn's implementation of k-means includes an improved initialisation algorithm (k-means++) which spreads out the initial centroids by choosing the second centroid and subsequent centroids with a probability proportional to the squared distance from the closest existing centroid (Arthur et al. 2007). All in all, the k-means algorithm can scale to a very large number of samples and its use case is a medium number of convex shaped clusters with even sizes and a flat geometry (scikit-learn developers n.d.[a]).

3.2.2 Agglomerative Clustering

Agglomerative clustering belongs to a larger set of hierarchical clustering algorithms which seek to build a hierarchy of clusters. It represents the bottom-up approach in which each observation starts in its own cluster and, moving up the hierarchy, pairs of clusters are successively merged together (until a predefined number of clusters is reached, if applicable) (ibid.). The applied merge strategy is determined by a linkage criterion specifying the distance to be used between observations. A common criterion is Ward, which requires the merging cluster pairs to minimise the sum of squared distances within all clusters. Suitable for Euclidean distances, Ward leads to the most regular cluster sizes and is robust to noisy data.

Generally, agglomerative clustering can scale to a large number of samples and its use case is a large number of clusters of any

shape (depending on the linkage criterion used), with possible connectivity constraints and non Euclidean distances in the data (for its ability to detect non-flat manifolds) (scikit-learn developers n.d.[a]). A major drawback, however, is that the algorithm cannot scale to a very large number of samples, as is the case in this project.

3.2.3 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm which groups together samples with many nearby neighbours, thereby creating clusters in areas of high density. Based on a minimum number of samples within a radius of a neighbourhood, DBSCAN classifies samples as core samples of a cluster, as non-core samples at the edge of a cluster, and as outliers (noise) (ibid.). Not asking one to specify the number of clusters in advance, scikit-learn's implementation of the method only requires a minimum number of samples in the neighbourhood of a core point *(min_samples)* and their maximum distance from it *(eps)* as input parameters.

Altogether, DBSCAN can scale to a very large number of samples and its use case is low-dimensional data with a medium number of clusters of any shape, uneven cluster sizes, and a non-flat geometry (ibid.). Moreover, the algorithm's notion of noise makes it robust to outliers, but its inability to effectively cluster data with large differences in density is a drawback of the method.

3.2.4 Elbow and Silhouette Method

Clustering algorithms like k-means confront the user with the problem of choosing the correct number of clusters in a dataset. The elbow method and the silhouette method are tools which help to solve this task using statistical calculations to specify the required input parameter. Considered a rough rule of thumb, the elbow method is a heuristic approach which often yields ambiguous results (Mahendru 2019). Yet, it can be used well together with the more refined silhouette method, which allows for a more reliable comparison as well as a validation of clustering results (Rousseeuw 1987).

By plotting the percentage of variance explained by the clusters against the number of clusters, one ideally obtains a curve shaped like an arm with an elbow indicating a drop in the marginal gain. The trade-off between maximising the explained variance and minimising the number of clusters is located at this point, which points out the optimal number on the abscissa (Wikipedia 2019). A variance of this method applied in this project uses the within-cluster sum of squares (WCSS)[4] to similarly find the balance between minimising the WCSS and the number of clusters.

The silhouette method employs the mean within-cluster distance and the mean nearest-cluster distance for each sample to calculate the silhouette coefficient over all samples. The silhouette scores can range from minus one to plus one; positive values indicate that a sample is well matched with its cluster, val-

[4]The WCSS, or within-cluster variation, is calculated from the sum of squared distances of samples to their closest cluster centre.

ues near zero indicate overlapping clusters and negative values indicate that a sample would be better matched with a different cluster. Hence, many low or negative values suggest that the selected number of clusters is suboptimal or that it is hardly possible to form clusters from the data, while high values validate the consistency within clusters (scikit-learn developers n.d.[c]).

4 Results

The following sections provide the analysis of the dataset based on the methods described above. First, I focus on the title representations created with Word2Vec's training algorithm and successively discuss the clustering results obtained with the k-means, agglomerative clustering, and DBSCAN methods. I then repeat my procedure with regard to GloVe's pre-trained word embeddings and the resulting title vectors. In any case, the title vectors' number of dimensions was immediately reduced to three by means of a principal component analysis (PCA), which increases the efficiency of the clustering process and allows for plotting the data in a three-dimensional coordinate system. Further, only stop-word-filtered titles were represented, since the resulting point clouds appeared more differentiated compared to those of unfiltered titles.

4.1 Trained Vectors: Word2Vec

The Word2Vec model was trained on the full set of 56,743 stop-word-filtered titles with a total vocabulary of 22,285 words, using 100 dimensions and 100 iterations. As no minimum frequency of words to be included in the model was set, the approach circumvented the problem out-of-vocabulary words. However, given the large number of hapax legomena, it ran the risk of low-quality vectors. That said, figure 2 shows the PCA-transformed title embeddings. At large, the resulting data points can be grouped into two merging clouds, each with a core of high density; one is more compact and the other covers a larger area of lower density.

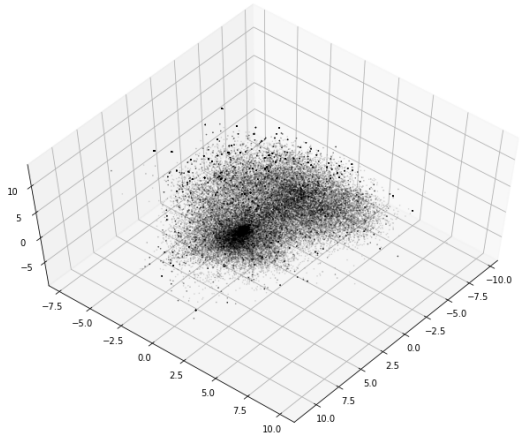

Figure 2: 56,743 Title Vectors (Word2Vec)

4.1.1 k-Means

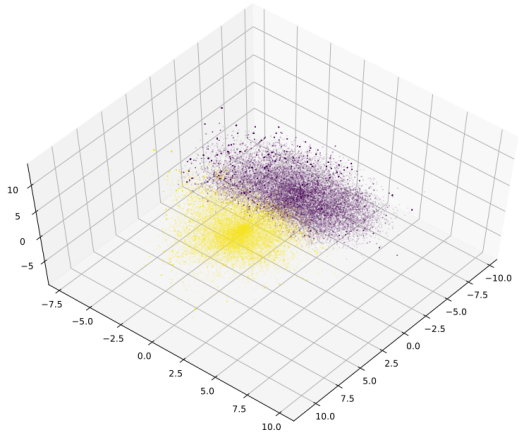

Figure 3: k-means, 2 clusters (Word2Vec)

In the course of the calculation of the op-

timal number of clusters for k-means, the elbow method proposed a number of two. Further on, the most promising results of the silhouette method were also obtained with two clusters. Exceeding the scores achieved with scikit-learn's implementation of k-means, the algorithm of the Natural Language Toolkit (NLTK) with a cosine distance metrics resulted in a maximum of 0.83, a minimum of -0.02, and a mean score of 0.61, thereby validating the clustering. Figure 3 shows two evenly sized clusters, separated as predicted but with sharply cut, straight edges which seem overly artificial.

By use of the centroids' nearest neighbours, the cluster centres can be represented by (1) J. M. W. Turner's artwork "cliffs beyond combe martin harbour" as well as by (2) Bernard Leach's "tile". Coming from only 27.4 percent of the collection's artists, 87.5 percent of the titles in the first cluster belong to artworks by Turner. Contrary to this, including works from 93.4 percent of the artists, the second cluster has a Turner share of only 30.8 percent.

Beyond that, the first cluster makes use of only 36.8 percent of the total vocabulary with a high share of non-words, while the second cluster uses 81.9 percent of the vocabulary and less non-words than average. In addition, the average title length in the first cluster is with 4.5 words about one word longer than in the second one. Furthermore, the most common words of the first cluster describe landscapes and buildings (e.g. "castle", "view", "river"), the ones of the second cluster, however, encompass expressions from the field of arts (e.g. "turner", ["'s",] "study", "inscriptions"). As Turner's titles, for the most part, seem to constitute a separate cluster, it can be assumed that they have a specific structure and choice of words, which distinguish them fundamentally from titles of other artists.

4.1.2 Agglomerative Clustering

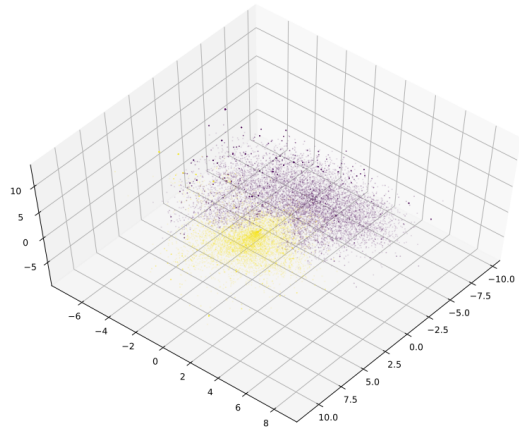

Figure 4: Agglomerative clustering, 2 clusters (Word2Vec)

The algorithm used for agglomerative clustering proved incapable of processing the full data set. I discovered a limit of about 20,000 data points and used a correspondingly large random sample from the set of title vectors. For technical reasons, the elbow method was not applied in the case of agglomerative clustering. As before, plotting the sample led to the assumption that a cluster number of two is the correct choice. The silhouette scores with two clusters had a maximum of 0.61, a minimum of -0.44, and a mean score of 0.33. Marginally better values were scored with four clusters, however, only achieved through the creation of a very small cluster. In any case, the method could not validate the clustering, for the low mean score and very low minimum indicate noisy data with a low density. As shown in Figure 4, the

method puts out a well defined small cluster, next to a larger one with a cleaved edge and very low density. On the whole, the clusters are similarly separated as in the case of k-means and have almost identical characteristics which are not worth further discussion in this section.

4.1.3 DBSCAN

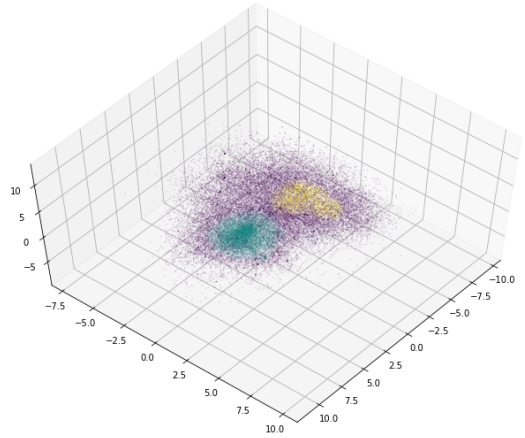

Figure 5: DBSCAN, 2 clusters (Word2Vec)

In the case of DBSCAN, duplicate points had to be removed to reduce memory and computation time, which reduced the number of samples to 41,427. The best results were obtained with an *eps* of 0.66 and a *min_samples* value of 202. This assigned 28.2 percent to a large first cluster, and 7.8 percent to a second one of smaller size, while 64.1 percent of the data points were labelled as noise. Figure 5 shows that the larger cluster covers the presupposed compact cloud almost entirely, while the smaller cluster is centred in the large cloud of low density. Most interestingly, the method highlights the very large number of outliers which are almost impossible to cluster.

In contrast to k-means, the clusters cannot be represented by titles, for DBSCAN does have the notion of centroids. The overall results, however, are even more differentiated: The first cluster includes only 12.0 percent titles by Turner, while the second one has a Turner share of 82.8 percent. Additionally, the average title length of the second cluster is exactly two words longer than in the first one. As could be expected, the titles of first cluster are mostly made up of words describing landscapes and buildings (e.g. “castle”, “near”, “view”), while those of the second one include expressions from the field of arts (e.g. “study”, “blue”, “sketch”), but also a notable number of non-words (e.g. “’s”, “ii”, “5”).

4.2 Pre-Trained Vectors: GloVe

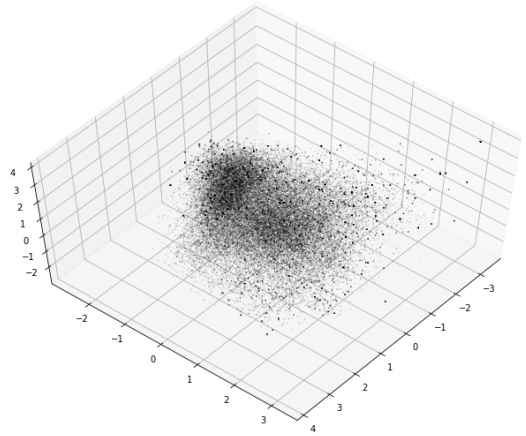

Figure 6: 49,471 Title Vectors (GloVe)

As the GloVe model with 300-dimensional vectors contains a limited set of 400,000 word embeddings, the artwork titles had to be filtered for out-of-vocabulary words before they could be represented as vectors. To ensure a valid representation, I decided to filter out any title containing at least one

word which is not included in GloVe's vocabulary. This reduced the number of usable titles to 49,471, with a total vocabulary of 16,517 words. Figure 6 shows the PCA-transformed title embeddings. Again, the data points can be grouped into two merging clouds with relatively dense centres; one compact cloud and another widely scattered one.

4.2.1 k-Means

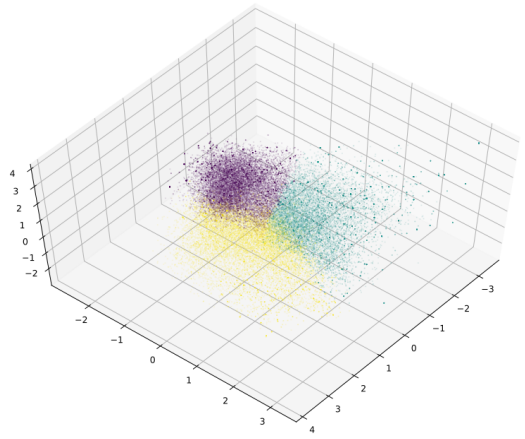

Figure 7: k-means, 3 clusters (GloVe)

The elbow method suggested three clusters and the best results of the silhouette method were also obtained with the same number of clusters. Using the NLTK tool with a cosine distance metrics, the best scores resulted in a maximum of 0.83, a minimum of -0.11, and a mean score of 0.53, which can be interpreted as a modest validation of the clusters. Illustrating k-means aim to form evenly sized clusters, Figure 7 shows the respective results: a point cloud cut into three parts of almost the same size.

In this case, the cluster centres could be represented by (1) Albert Richards' "[the] landing h hour minus 6 [in the] distance glow [of the] lancasters bombing battery [to be] attacked" as well as by J. M. W. Turner's (2) "[a] fort [on a] cliff [by the] sea" and (3) "[the] piazza castello turin". Further analysis revealed that the first cluster encompasses the highest share of artists (47,1 percent), but the lowest proportion of Turner's titles (33,1 percent). The second cluster includes the lowest share of artists (15,5 percent) but the highest proportion of titles by Turner (85,8 percent). Including almost exactly one third of the data points (33,2 percent), the remaining cluster is remarkably average: its share of artists is with exactly four percentage points marginally above average (37,3 percent) and the Turner proportion is only 0.6 percentage points below-average (56,1 percent).

Apart from that, the titles of the second cluster consist of a very small vocabulary, accounting for only 22.1 percent of the total vocabulary. Furthermore, the proportion of non-words in the vocabulary of the first cluster is well below-average (23.7 percent), while the proportion in the third cluster is above-average (44.7 percent). Ranging from 3.5 to 3.7 words, the average title length does not significantly differ in all three clusters. The most common words of the first cluster encompass expressions from the field of arts (e.g. "study", ["'s",] "figures", "two"), the ones of the second cluster describe landscapes and buildings (e.g. "river", "castle", "mountains"), and those of the third are from both categories (e.g. "turner", "engraved", "sketches" and "castle", "st", "view", respectively). This undergirds the assumption that the title's choice of words determines the clustering results and, further, that the vocabulary can be divided into two broad

categories; one descriptive set focused on landscapes and buildings and another more academic one specialised in the description of artworks themselves.

4.2.2 Agglomerative Clustering

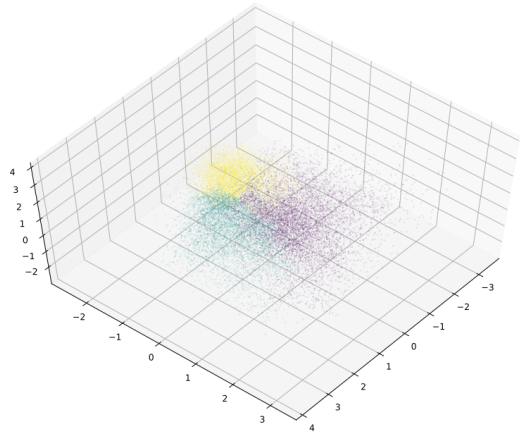

Figure 8: Agglomerative clustering, 3 clusters (GloVe)

As the algorithm of the agglomerative clustering method could not handle the total number of title vectors, I used a random sample containing 20,000 data points. Again, the plotted sample led to the assumption that a cluster number of two is the correct choice. The silhouette method, however, scored similar means for all numbers of clusters from two to six. The most promising, but also most extreme results were achieved with three clusters, which produced a maximum of 0.64, a minimum of -0.48, and a mean silhouette score of 0.26. Given the low mean score and the very low minimum, once again the method could not validate the clustering. Figure 8 shows that the methods put out clusters similar to those of k-means, but with more nuanced cluster edges. On the whole, the cluster's characteristics are largely identical again.

4.2.3 DBSCAN

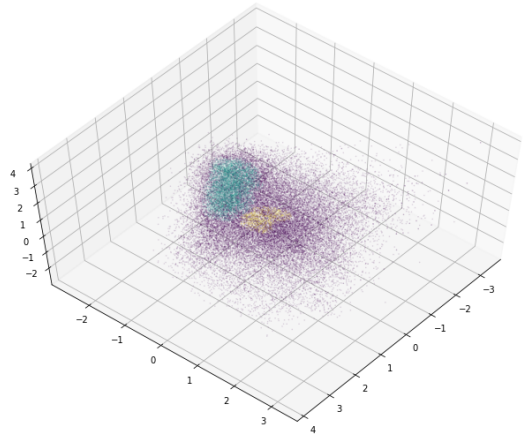

Figure 9: DBSCAN, 2 clusters (GloVe)

After filtering duplicate points for DBSCAN, 35,140 samples were left for clustering. The best results were obtained with an *eps* of 0.21 and a *min_samples* value of 129. This assigned 19.5 percent to a large first cluster, and 2.8 percent to a second one of smaller size, while 77.7 percent of the data points were labelled as noise. Figure 9 shows that the larger cluster covers the presupposed compact cloud almost entirely, while the smaller cluster is centred in the large cloud of low density. The number of outliers which cannot be clustered is even higher than with the trained vectors.

The first cluster includes 27.2 percent titles by Turner, while the second one has a Turner share of 72.2 percent. Same as with the Word2Vec model, the average title length of the second cluster is exactly two words longer in comparison to the first cluster. As expected, the titles of the first cluster encompass expressions from the field of arts

(e.g. ["'s",] "study", "figures", "two"), while those of the second one are mostly made up of words describing landscapes and buildings (e.g. "view", "looking", ["'s",] "rome").

5 Conclusion and Future Work

In summary, the two sets of title vectors, one based on embeddings trained with Word2Vec and the other grounded in the pre-trained embeddings of GloVe, produced similar results in a three-dimensional coordinate system: two merging point clouds with differently spread cores of high density. In both cases, one more compact cloud and another cloud covering a larger area of lower density could be interpreted as clusters. Each of the three clustering algorithms, which all belong to different families of clustering methods, demonstrated their individual strengths and weaknesses and proved to be suitable for the specific task to varying degrees.

For the two cases under consideration, k-means produced clusters which could be validated by the silhouette method. Its inherent logic of producing clusters of even size, however, does not seem appropriate for the relevant data. Considering the overall shape of the plotted title vectors, it produced overly artificial cluster shapes with clear-cut edges.

Agglomerative clustering, in contrast, produced more nuanced transitions in similarly shaped clusters. Nevertheless, due to the method's major drawback, its incapability to handle a large number of samples, agglomerative clustering has proven to be unsuitable for the task of clustering the total number of titles included in the dataset.

With regard to its results, DBSCAN, which stands out because of its notion of noise, is hardly comparable to the other methods applied in this project. Large differences in density complicated the task of finding appropriate values for its parameters, but once found, the method reliably detected areas of high density, which it interprets as clusters. Thereby, DBSCAN separated outliers, so that the actual essence of clusters could be found.

On the surface, the results of both embedding methods differ only insofar as that remarkably more non-words (in the actual sense) are found in one of DBSCAN's clusters when using the trained vectors. This can be traced back to the conditions under which the models were created: While GloVe was trained on a very large corpus with a vocabulary of 400,000 words, the Word2Vec model learned from a limited number of artwork titles, which usually have a very specific structure and vocabulary, as well as a high number of hapax legomena.

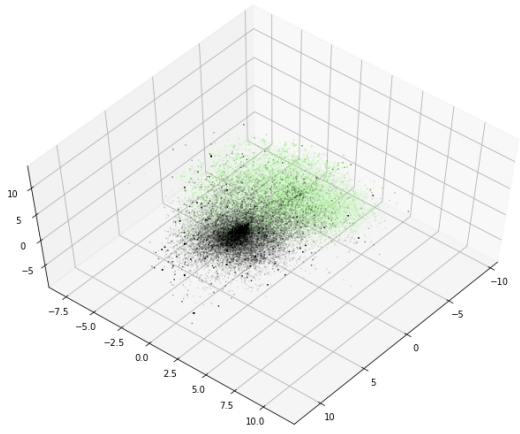

Figure 10: Title vectors by J. M. W. Turner coloured in green (Word2Vec)

Nevertheless, as figure 10 and 11 illustrate, both models yield similar scatter plots of two merging clouds, which can be separated best when titles by J. M. W. Turner are

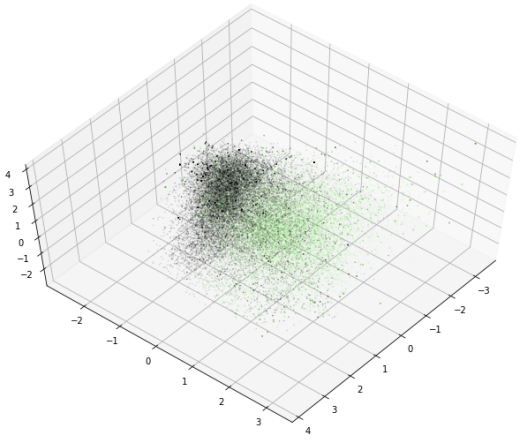

Figure 11: Titles vectors by J. M. W. Turner coloured in green (GloVe)

highlighted (in green). Hence, it can be assumed that Turner's titles have a special structure and choice of words, with the result that they form a cluster of their own.

Further, I found that the vocabulary can be divided into two broad categories: One set is largely descriptive and focused on landscapes and buildings, while the other is more academic and specialised in the description of artworks themselves. The descriptive vocabulary constitutes most of the titles by Turner, but the exact link between the choice of words and a broader structure of the titles remains unclear. The title length, however, seems to be an important influence on the clustering results.

As *The Tate Collection* includes a substantial number of titles from Turner, future work could experiment with datasets which are more balanced with regard to the artists and artworks included. Besides, an approach focusing on collocations and metaphores, for example, could reveal interesting aspects of artwork titles which remain hidden in this project. Beyond that, more recent techniques of word representation, which are able to build vectors for unseen words, have the potential to increase the informative value of the title vectors, especially when confronted with a large number of hapax legomena. Models like ELMo and FastText, for instance, even use the character and morphological structure of words to embed them in a vector space (Rakhmanberdieva 2019). Apart from that, the application of topic modelling techniques, such as LDA, to clustering results could reveal the latent topics of clusters – an issue omitted in this project, owing to its limited scope, but left for further research.

References

Arthur, David and Sergei Vassilvitskii (2007). "k-means++: The Advantages of Careful Seeding". In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics Philadelphia, pp. 1027–1035. URL: https://web.archive.org/web/20060209181757/http://www.cs.umd.edu/~mount/Papers/kmlocal.pdf.

Butnaru, Andrei M. and Radu Tudor Ionescu (2017). "From Image to Text Classification: A Novel Novel Approach Approach based based on Clustering Word Embeddings". In: *Procedia Computer Science* 112, pp. 1783–1792.

Choi, Won-Joon and Euhee Kim (2019). "A Large-scale Text Analysis with Word Embeddings and Topic Modeling". In: *Journal of Cognitive Science* 20.1, pp. 147–187.

Corrêa Jr., Edilson A., Vanessa Queiroz Marinho, and Leandro Borges dos Santos (2017). "NILC-USP at SemEval-2017 Task 4: A Multi-view Ensemble for Twitter Sentiment Analysis". In: *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, pp. 611–615.

Mahendru, Khyati (June 2019). *How to Determine the Optimal K for K-Means?* Medium (accessed March 22, 2020). URL: https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb.

Mikolov, Tomas et al. (Sept. 2013). "Efficient Estimation of Word Representations in Vector Space". In: *Cornell University*. URL: https://arxiv.org/abs/1301.3781.

Pennington, Jeffrey and Christoper D. Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: https://nlp.stanford.edu/pubs/glove.pdf.

Rakhmanberdieva, Nurzat (2018a). "Word Representation in Natural Language Processing Part I". In: *Towards Data Science*. URL: https://towardsdatascience.com/word-representation-in-natural-language-processing-part-i-e4cd54fed3d4.

— (2018b). "Word Representation in Natural Language Processing Part II". In: *Towards Data Science*. URL: https://towardsdatascience.com/word-representation-in-natural-language-processing-part-ii-1aee2094e08a.

— (2019). "Word Representation in Natural Language Processing Part III". In: *Towards Data Science*. URL: https://towardsdatascience.com/word-representation-in-natural-language-processing-part-iii-2e69346007f.

Rousseeuw, Peter J. (Nov. 1987). "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20, pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7.

scikit-learn developers (n.d.[a]). *Clustering*. scikit-learn (accessed March 2, 2020). URL: https://scikit-learn.org/stable/modules/clustering.html.

scikit-learn developers (n.d.[b]). *KMeans*. scikit-learn (accessed March 24, 2020). URL: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans.

— (n.d.[c]). *Silhouette Score*. scikit-learn (accessed March 22, 2020). URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.

Wang, Jiang et al. (Dec. 2017). "From Partition-Based Clustering to Density-Based Clustering: Fast Find Clusters With Diverse Shapes and Densities in Spatial Databases". In: *IEEE Access* 6, pp. 1718–1729. DOI: 10.1109/ACCESS.2017.2780109.

Wikipedia (2019). *Elbow method (clustering)*. Wikipedia (accessed March 27, 2020). URL: https://en.wikipedia.org/wiki/Elbow_method_(clustering).

Written Insurance

I hereby certify that I have prepared this written project report independently. All passages which are taken from the wording or the meaning of other works (including electronic sources) have been clearly marked in each individual case with precise indication of the source.

Bielefeld, March 28, 2020

_____

(Max Harder)