Computer Science Department

B.Sc. Computer Science
Computer Science Department

Machine Learning Approaches for the Analysis of Mythology and Folklore

Candidate

Michael Hills

Student ID 700013809

Supervisor

Dr Chico Camagaro

University of Exeter

Academic Year
2022/2023

Abstract

Myths are traditional narratives that serve as the foundation of a culture, allowing us to interpret the world and our existence within it. They set idealistic values and behaviours for the way we conduct our lives and define our cultural identity. There is therefore growing interest in the study of mythology. Technological developments, such as machine learning, enables the automated analysis of large digitised datasets, providing a greater insight into the similarities and difference between myths of different cultures. This project investigated the use of data mining, natural language processing and clustering methods to analyse the myths recorded on the Myths and Folklore Wiki that contains over 3000 pages. A total of 107 well-defined and meaningful clusters were created that successfully separated myths based on their description on the Wiki. The analysis of these clusters allowed for cultural patterns between cultures to be identified. The project therefore successfully demonstrated that natural language processing and machine learning methods are able to quickly and effectively identify similarities between myths of different cultures.

	Yes	No
I certify that all material in this dissertation which is not my own work has been identified.	☑	☐
I give the permission to the Department of Computer Science of the University of Exeter to include this manuscript in the institutional repository, exclusively for academic purposes.	☑	☐

Contents

1 Introduction

Culture is defined as the ideas, customs and social behaviour of a particular group or society and is integral to our lives as it shapes our identity, beliefs and values. Through culture, we develop a sense of belonging within a community by the creation of shared values that help us to relate to and respect one another [1]. Therefore, to create a society with equitable and fair values requires us to understand and embrace different cultures. It is only by acceptance of cultural diversity that we can create an inclusive world where creativity, innovation and prosperity can thrive.

Myths are traditional narratives that serve as the foundation of a culture, providing a means by which people can interpret the world and their existence within it [2, 3]. They reflect the ideal characteristics of human and social behaviour and set idealistic values for the ordering of society [3]. The sharing and spread of myths throughout generations is fundamental to the development of cultural identities. Therefore, to understand a particular culture and how it identifies itself in relation to others, it is necessary to understand the mythology that lies at its core [4].

Historically, these myths have been preserved through a variety of methods, such as oral storytelling, art and ancient manuscripts. However, with recent technological advancements, particularly the internet, there is an increased interest in the development of large digital collections of myths [5]. This opens up an exciting research opportunity to study and analyse these collections and provide a much improved understanding of the mythologies that lie at the soul of our cultures.

1.1 Project Specification and Aims

The Myths and Folklore Wiki is an open, easy access website for the cataloguing and study of mythological and folkloric tales and traditions and currently contains over 3000 pages. The aim of this project is to extract information from each page to highlight the similarities and differences between the mythologies of different cultures through the use of natural language processing and clustering algorithms, with the objective of answering the following research questions (RQ):

RQ_1: How is the dataset distributed over different cultures?

RQ_2: Can myths be separated into meaningful clusters?

RQ_3: Are there common tropes between myths?

RQ_4: How do myths vary between different cultures?

There are a number of functional requirements (FR) that must be met to answer these research questions:

FR_1: Exploration and collection of myths on the Wiki

FR_2: Semantic analysis of myths using SBERT

FR_3: Dimensionality reduction and clustering using UMAP and HDBSCAN

FR_4: Analysis of the topics found within clusters

FR_5: Visualisation and evaluation of the cultural spread within topics

The project will be considered successful if all the functional requirements above are fulfilled, leading to justified conclusions for each of the research questions. However, it is important to note that for these conclusions to be valid, there must be evidence to show the generated clusters are well-defined. This is achieved through the use of cluster evaluation metrics such as silhouette coefficient and density-based cluster validation.

1.2 Project Motivation

The Wiki represents only a very small proportion of the available content on the internet relating to mythology and folklore. With such a vast quantity of available material, it is almost impossible to closely analyse even a fraction of the content by hand. Such analysis, without any form of automation, leads to a selection bias in the myths that are studied, often omitting some of the lesser known myths and folk tales [6]. However, the use of data mining and machine learning reduces selection bias as all myths are collected efficiently and analysed equally, allowing for new unseen connections between cultures to be uncovered.

In addition, myths between different cultures are often interrelated, yet at first glance there are no apparent similarities. It is only with further study that their common elements and internal connections can be identified [7]. Dimensionality reduction is therefore a particularly useful tool as it highlights the common elements between myths. Subsequent clustering can then efficiently sort myths into appropriate categories based on the identified common elements [8]. Furthermore, the use of data mining and machine learning tools to cluster myths between cultures not only highlights the similarities of myths but also their differences. By understanding and acknowledging these similarities and differences, we are able to gain a greater appreciation of the origins of different cultures leading to a society that is much more accepting and inclusive.

Although similar projects exist that automate the analysis of mythology, they often focus on a single genre of mythology, such as the analysis of Dutch folklore or the case of Greek songs [5, 9]. In addition, other published projects looking at a wider range of mythology often have shallow analysis that fail to offer conclusions with meaningful information [10]. These gaps in knowledge open up an exciting research opportunity to improve on the research previously undertaken.

2 Design, Methods and Implementation

2.1 Data Collection

The Mythology and Folklore Wiki is based on the principle of impromptu crowd science, by which all users have the ability to create and edit pages [11]. Since each page is written by different users, their structure and format varies significantly. However, the majority of pages have the same broad characteristics, with a brief overview of the myth and a list of categories at the top of the page, as well as a summary style table of its key features. These common characteristics allow for the use of automatic parsing to iterate through each page and collect the important information of each myth.

In this project, the automatic parser was implemented through the use of the Beautiful Soup python library to collect all relevant information (i.e. main text, categories and tables) from within the HTML of each page. This is achieved by the navigation of parse trees created by Beautiful Soup from which

the HTML tags containing this information are identified and their content extracted [12].

The main texts for each page are stored within paragraph and list tags within a parent division tag of class "mw-content-text". This allows for easy identification of all text relevant to a myth and ignores any text outside the division tags, such as the "fan feed" and "user discussion" sections. However, it is important to note that not all text collected about a myth is useful, with sections such as the Gallery, Sources and Citations providing no relevant information to the final clustering. The text within these sections is therefore discarded, allowing for more meaningful clusters to be created. In this way only the important information is used within the natural language processing and clustering of pages. To discard the irrelevant text, each collected HTML tag is passed through a series of 'if statements' to remove all tags that are part of a class, table or have parent header tags with names of sections that are deemed unimportant to collect. The removal of tags with classes is due to a large number of pop-up advertisements that include text that is not relevant to the myth and should therefore not be collected. Additionally, all tables are removed as these are collected separately to the main text. The code for removing text with unimportant headers is shown below (the full series of 'if statements' is provided in the Appendix).

```
page = requests.get(url)
soup = BeautifulSoup(page.content, 'html.parser')
content = soup.find_all(id = 'mw-content-text')[0]
text = content.find_all(['p','li'])

notCollect = ['relations','family','see also','external','gallery','sources','citations',
                'references','film','video Games','tv shows']

for item in text:
    headers = []
    #Find headers of paragraph tags
    headers.append(item.parent.find_previous_sibling(['h2','h3']))
    #Find headers of list tags
    headers.append(item.find_previous_sibling(['h2','h3']))

    if (len(headers) != 0):
        for header in headers:
            if header != None:
                if (any(notCol for notCol in notCollect if (notCol in header.get_text().lower()))):
                    add = False
```

Code 2.1: Code snippet for discarding text with unuseful headers

The collection of data within the summary tables was undertaken as follows:

1. Set the parent tag to the aside tag of class "portable-infobox pi-background pi-border-color pi-theme-wikia pi-layout-default"
2. Collect all division tags of class "pi-item pi-data pi-item-spacing pi-border-color" within the parent tag. These represent a row in the table, containing a label and a value (e.g. a label of 'Species' with value 'Demon')
3. Collect the label value pairs within each row by finding the h3 tag of class "pi-data-label pi-secondary-font" for the label and division tag of class "pi-data-value pi-font" for the value.

The labels and values were then stored in a list of dictionaries for later use. However, again there are some sections common within the summary tables that provide no useful information (e.g. Languages, providing the names of the myth in non-English languages). These are discarded using the same technique as with the main text by identifying the parent headers. Finally, to collect the categories defined for each page, all anchor tags within the division of class "page-header__categories" are collected and stored in a list for later use.

The methodologies described above enable the collection of key information from an individual page. However, before any natural language processing can be performed the key information must be extracted and stored from every page. This is achieved by using the "All Pages" section of the Wiki displaying approximately 150 pages at a time, starting from a specified myth found in the URL. For example, `https://mythus.fandom.com/wiki/Special:AllPages?from=Angeoa` shows myths alphabetically, starting with Angeoa. Initially, the method to collect all pages was implemented as follows:

1. Collect all links present on the page
2. Collect the main text, categories and summary table for each link and store in a list containing all currently collected myths
3. Append list to a JSON array
4. Reload the page with the starting myth as the next myth to be collected (change the ?from= in the URL to the next myth)
5. Repeat until every myth is collected.

It is important to ignore the collection of links in either of the following circumstances as they provide no meaningful information to a cluster:

i. Any link tag that is a member of a class, as these represent redirects to pages with alternate names for a myth. This approach avoids the myth being collected multiple times
ii. Any link containing the string "Gallery" or "disambiguation", as gallery pages contain only images and disambiguation pages list articles of the same title.

The initial collection method outlined above proved to be inefficient. This was due to the appending of a JSON array that requires the parsing of the whole array before adding new elements and resulted in a large overhead and slow collection of myths. For this reason, the implementation was altered to use JSON Lines, in which each line in the file is a standalone JSON document. This allows for efficient appending to a file without the need to parse through the whole array each time and resulted in a significant speed up in the collection of myths [13]. Additionally, a simple JSON file was added to keep track of the most recently stored myth, so that collection can be run over multiple sessions and is not impacted by potential network errors.

2.2 Word and Sentence Embeddings

2.2.1 Word Embeddings

Word embeddings are a learned representation of a word as a high dimensional vector, by which words with similar semantics create vectors of similar values. As discussed in the literature review, there are many methods to achieve this with CBOW and Skip-Gram being two of the most common. Both methods use neural networks with a single hidden layer to learn the probabilities of words appearing in the same context. However, the key difference between them is the part of the text being predicted. CBOW uses the surrounding words as the input to predict a central word, whereas Skip-Gram uses one central word to predict the surrounding words [15]. This is illustrated in Figure 2.1. Both methods were implemented in this project by using Gensim's Word2Vec library, with the aim of finding similarities between the summary tables of each myth.

Gensim provides many pretrained Word2Vec models trained on a variety of text corpora such as the Google News and Twitter datasets that contain vectors for 3 and 1.2 million words respectively.

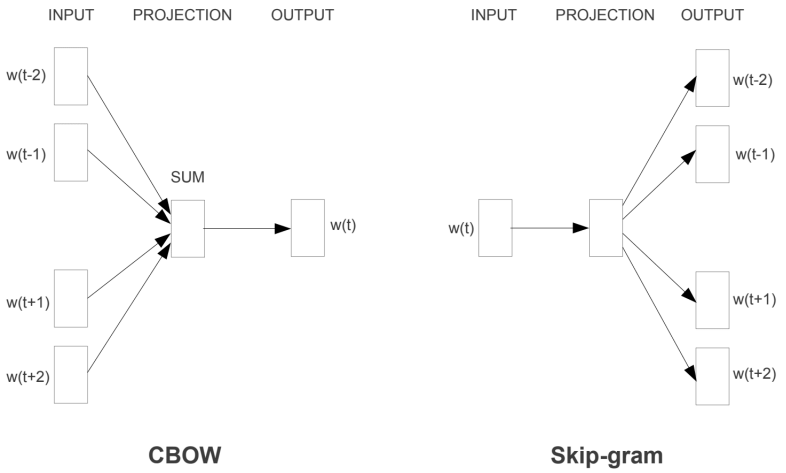

Figure 2.1: Illustration of CBOW & Skip-Gram [14]

Both datasets were tested in the natural language processing and clustering of the summary tables, summing the vectors for each word and averaging the result. However, neither model successfully captured the semantic meaning of pages, resulting in very poor clusters with cluster confidence costs (explained in Section 2.3.2) of 0.47 and 0.43 for the Google News and Twitter datasets respectively. This is likely due to uncommon descriptive words being used in the tables that are not present in either dataset. Any words that are not present in the dataset do not have associated vectors and are ignored, losing the semantic meaning of the table. Therefore, a different method was tested that used the main text of each page to train a new Word2Vec model. This was based on the idea that previously unknown words will be present in the main text such that their vectors can be generated based on the context of their surrounding words. The code to create a new model is shown below (Code 2.2).

```
class MyCorpus:
    def __iter__(self):

        with jsonl.open('data.jsonl','r') as f:
            for line in f:
                #removal of punctuation
                processed = strip_punctuation(line['text'])
                yield utils.simple_preprocess(processed)

def getModel():

    sentences = MyCorpus()
    model = gensim.models.Word2Vec(sentences=sentences,min_count=1,epochs=15,sg=1)
    model.save("word2vec.model")

    return model
```

Code 2.2: Code for training Word2Vec using the main text

By training a new model, Word2Vec successfully found the semantic similarities between tables, resulting in significantly improved clustering with confidence costs of 0.093 for both CBOW and Skip-Gram models.

Cluster confidence was also found to improve by limiting the number of rows used to create the vectors for each table. A likely explanation for this improvement is that the key information appears at the top of each table, whereas less meaningful information (e.g. equivalent myths from other cultures) is found lower in the table. By limiting the number of rows, only the key information is used to create vectors that capture only the meaningful information to cluster. Furthermore, cluster confidence was found to further improve by the removal of punctuation and lower-casing each word

used in the training and usage of the Word2Vec model. This ensures the words stored in the model are consistent with the words in the summary tables, leading to a model that is not skewed by alternative uses of grammar. For example, 'god' and 'God's' will have the same vector representation so will still be recognised as having identical meanings, resulting in improved clustering.

2.2.2 Sentence Embeddings

Sentence embeddings are an extension of the ideas of word embeddings, with the key difference being that whole sentences, rather than individual words, are vectorised creating similar vectors for sentences of similar meanings [16]. As discussed in the literature review, there are many methods to achieve this with the most modern being those that use deep learning transformer models that can process entire sentences at once, rather than one word at a time. This leads to a much quicker execution time and greater understanding of text [17].

The sentence embedding method used in this project is SBERT, a transformer based model provided by the 'Sentence Transformers' python library. SBERT was selected over other similar models due to its ability to analyse sentences very efficiently whilst retaining a high level of precision. This has been demonstrated in other projects, in which BERT took 65 hours to analyse 10000 sentences, whilst SBERT took only 5 seconds to achieve the same level of precision [18]. Additionally, 'Sentence Transformers' provides many pretrained SBERT models for a variety of use cases, together with an in-depth evaluation of each model's performance and speed and this allows for experimentation into the most suitable model for this project. The models tested in this project and their performance are as follows:

(a) 'sentence-t5-xxl': Selected as it is the best performing model provided by 'Sentence Transformers'. However, the size of the model is larger than the available RAM on the laptop used for this project, so the model failed to load.

(b) 'all-roberta-large-v1': Selected due to its high performance and smaller model size. This model resulted in very good cluster confidence costs. However, the model is very slow, as confirmed in the 'Sentence Transformers' documentation.

(c) 'all-MiniLM-L6-v2': Selected due to its efficiency (18x faster than 'all-roberta-large-v1') and equally high performance. This model also resulted in very good cluster confidence costs, with much faster calculation of the sentence vectors.

In addition, 'Hugging Face' provides a number of multilingual pretrained models, by which similar sentences in different languages result in vectors of similar values. These models were also tested with the idea that the presence of non-English (i.e. describing myths from other cultures) text would benefit from a model capable of distinguishing and understanding different languages. However, the multilingual models performed significantly worse than the monolingual models. This may be explained by the majority of text on the Wiki being in English, so models trained solely in English text are more effective. Therefore, the sentence embedding model selected for the analysis of the main text of each page is 'all-MiniLM-L6-v2' due to its high performance and speed, leading to well-defined clusters that clearly display the similarities between myths.

2.3 Dimensionality Reduction and Clustering

The word and sentence embeddings both produce high dimensional vectors of lengths 100 and 384 respectively, in which each dimension is a feature that partly represents the meaning of a myth. However, with a high number of features the data becomes sparse and clustering fails to find the

connections between each myth. To resolve this, dimensionality reduction is performed to reduce the number of features with minimal loss of information, leading to well-defined clusters as a result of decreased noise within the data [19]. Furthermore, reducing the number of dimensions lowers the computational cost involved in clustering, decreasing run-time and allowing for more experimentation in the parameter tuning (Section 2.3.3) to further increase cluster quality.

The dimensionality reduction method chosen in this project is UMAP (Uniform Manifold Approximation and Projection), due to its ability to balance between global and local structure to create an accurate approximation of the original data in a lower dimensional space. Additionally, when compared to other similar methods, such as t-SNE, UMAP is significantly quicker and efficiently scales to large datasets [20].

2.3.1 Clustering

With the dimensionality reduction performed, the myths can be clustered. As discussed in the literature review, a number of different clustering methods are available, with two of the most popular being density-based and hierarchical clustering. Density-based clustering methods detect areas of high concentration in the data and create clusters for these areas, allowing for varying and arbitrary-shaped clusters that are not skewed by outliers [21]. In contrast, hierarchical clustering creates a tree of nested clusters, starting with all points contained in their own clusters before iteratively combining the two closest clusters until one large cluster is created containing all data points. The tree generated shows how clusters are merged over time and allows for the user to stop the algorithm at a desired number and size of clusters [21].

HDBSCAN is an algorithm that combines the methods of density-based and hierarchical clustering, allowing for the advantages of both methods to be utilised. HDBSCAN therefore ignores the presence of outliers and can easily find clusters of varying shapes due to the benefits of a density-based model, whilst also allowing for the size and number of clusters to be chosen by the user from the hierarchy created [22]. These advantages are key to the success of this project, as myths from the Wiki are likely to contain outliers and form non-uniform clusters. The ability to choose the level of clustering allows for the ideal size and number of clusters to be created and their cultural proportions calculated to highlight the similarities of the different cultures. Therefore, HDBSCAN is the chosen clustering method for this project.

2.3.2 Cluster Evaluation

Once the clusters are created, they must be validated to ensure they are good quality, well-defined, and do not include outliers that could skew conclusions. Several methods to achieve this were used in this project, with their performance shown below:

Silhouette Coefficient

The first method tested was the silhouette coefficient that calculates the validity of a cluster by performing the following:

1. Calculate the point's average distance to all other points in the same cluster. Name this a_i
2. Calculate the point's minimum distance to a point within a different cluster and average this for all the other clusters. Name this b_i
3. The silhouette coefficient for the point is $s_i = (b_i - a_i)/max(a_i, b_i)$

The above calculates a silhouette coefficient for a single point that can be averaged for every point within a cluster to determine the quality of a whole cluster [23]. The ideal value of 1 represents a cluster with points that are close together, and points from other clusters being further away. In this project, the silhouette coefficient was implemented using sklearn's silhouette_score package. However, in practice the silhouette values generated from testing all models and parameters (Sections 2.2 and 2.3.3) were always low. This is likely due to the calculation of silhouette coefficient using distances to measure the separation of points within clusters, whereas HDBSCAN uses density to create clusters. To achieve a more accurate cluster evaluation score a model using density rather than distance should therefore be used [24].

Density-Based Cluster Validation

The second method tested was DBCV (Density-Based Cluster Validation), provided by HDBSCAN's relative_validity package. This method considers the density and shape of clusters to calculate cluster quality, so is more suited to the HDBSCAN clustering used in this project. The calculation performed uses the inverse of the density of each point with respect to the density of all other points within its cluster, with an ideal value of 1 [24]. This method performed much better than the silhouette coefficient, creating varying scores for different HDBSCAN parameters. However, it does not provide an objective measure of cluster quality, but instead a relative score for the use in the parameter tuning of HDBSCAN. Therefore, scores generated from different parameters of UMAP cannot be accurately compared.

Cluster Confidence Cost

The final method tested was the implementation of a score that can be compared between runs with different UMAP parameters. This is achieved by the use of the 'probabilities_' attribute provided by HDBSCAN, in which each point has a score showing the certainty of it being placed in the correct cluster. By calculating the proportion of points under a certainty threshold, a score can be generated to show the clustering quality.

```
def score_clusters(clusters, prob_threshold = 0.05):

    cluster_labels = clusters.labels_
    label_count = len(np.unique(cluster_labels))
    total_num = len(clusters.labels_)
    cost = (np.count_nonzero(clusters.probabilities_ < prob_threshold)/total_num)

    return label_count, cost
```

Code 2.3: Code to calculate cost score

The code snippet above calculates the percent of points with less than a five percent cluster certainty, with scores closer to 0 indicating better clustering. This method resolved the issues of the two previously described methods, allowing for the effective calculation of cluster quality between runs with different parameters of UMAP and HDBSCAN.

2.3.3 Parameter Tuning

By using the cluster confidence cost described in the previous section, efficient parameter tuning can be achieved due to the easy comparison of cluster quality between runs with different parameter values for UMAP and HDBSCAN. To generate meaningful and well-defined clusters, the parameters requiring tuning are as follows:

UMAP (a) n_components: The number of dimensions in the low dimensional representation

(b) n_neighbors: Balances local versus global structure in the data, low values focus on local structure and higher values focus on global structure

(c) min_dist: Provides the minimum distance that points are allowed to be in the low dimensional representation

HDBSCAN (a) min_cluster_size: Smallest number of points allowed to be considered a cluster

(b) min_samples: Provides a measure of how conservative the clustering is, with larger values being more conservative and more points labelled as noise

Additionally, for this project to be successful, the calculated cost and the number of clusters generated are equally important. This is because a small number of clusters leads to poor semantic separation of myths due to the combination of a wide variety of myths in the formation of larger clusters. To obtain clusters that are both meaningful and of high quality, the above parameters must therefore be tuned to create a large number of clusters, whilst retaining a low cluster confidence cost. This was initially achieved by changing one parameter at a time to maintain a low cost whilst increasing the number of clusters created. However, as there were five parameters to tune, this was an extremely time consuming process. Therefore, two automated methods were tested to find the ideal values for each parameter.

Random Search

The first method implemented was a random search to automatically evaluate hundreds of parameter combinations, storing the results in a dataframe sorted by the cost calculated in the cluster confidence method. By plotting the pairwise relationships between the parameters, the general pattern of how each parameter changes the number and quality of clusters can be derived.

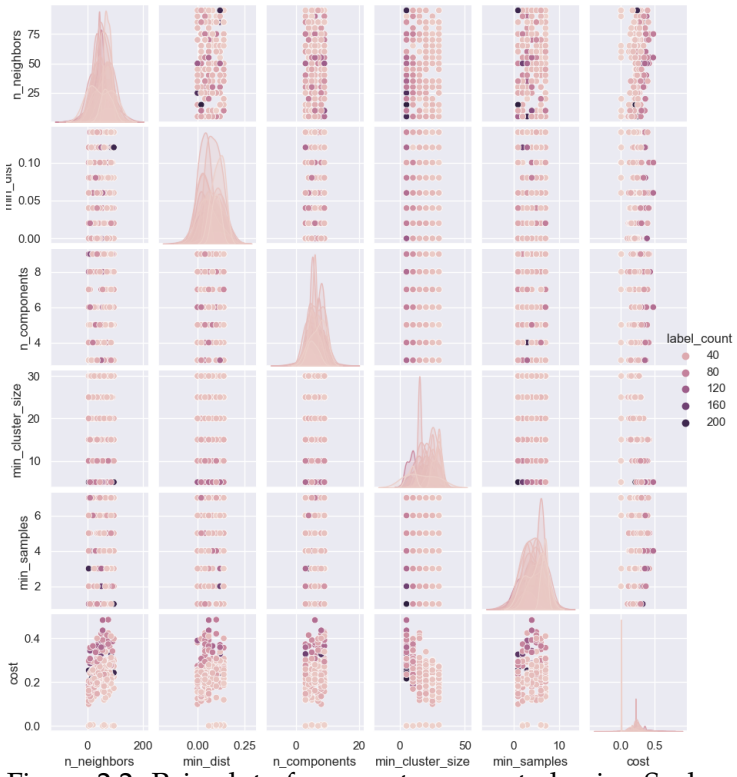

Figure 2.2: Pair-plot of parameters: created using Seaborn

Figure 2.2 above shows that by reducing the min_cluster_size, min_samples and n_components the number of clusters created is increased, leading to improved semantic separation of myths. The

random search can then be repeated, limiting the search space to smaller values of the three discussed parameters, leading to improved parameter values that result in both a large number of clusters and a low cost. This proved to be an effective but very slow method, due to randomly testing hundreds of parameter values each time it is run [25]. Another similar method was therefore tested that uses knowledge from each iteration to improve the parameters over time, rather than continually choosing random values.

Bayesian Search

Bayesian search is a popular parameter optimisation method used in a wide variety of problems. It works by keeping track of previous parameter evaluations that are used to form a probabilistic model, mapping parameters to an objective function [26]. In this project, the objective function is to minimise the confidence cost (Section 2.3.2), whilst also generating a sufficient number of clusters. Therefore, the objective function returns the cluster confidence cost with an added penalty if the number of clusters is outside a specified range.

```
def objective(params, embeddings, label_lower, label_upper):

    clusters = generateClusters(embeddings,
                                n_neighbors = params['n_neighbors'],
                                min_dist = params['min_dist'],
                                n_components = params['n_components'],
                                min_cluster_size = params['n_components'],
                                min_samples = params['min_samples'])

    label_count, cost = score_clusters(clusters, prob_threshold = 0.05)

    if (label_count < 35) | (label_count > 200):
        penalty = 0.10
    else:
        penalty = 0
    loss = cost + penalty

    return {'loss': loss, 'no_clusters': label_count, 'status': STATUS_OK}
```

Code 2.4: Code for the objective function used in Bayesian Optimisation

The objective function (Code 2.4) is then used in the Bayesian Optimisation of the parameters using the 'hyperopt' python library. This method proved more efficient in the parameter tuning of UMAP and HDBSCAN than the random search, due to its use of previous knowledge to select the next parameters to be tested. The penalty term added also resulted in the selected parameters yielding results of low cost and a high number of clusters, both of which are ideal for use in this project.

2.4 Cluster Labelling and Culture Proportions

Once the clusters are created, their automatic labelling can be achieved through the implementation of count vectors. Count vectors are a relatively simple concept in which the number of occurrences of a word within a cluster represents its importance. For example, if the word 'God' appears frequently within a cluster, it is considered important and used to label the cluster. Count vectors is implemented in this project by the use of the CountVectorizer package provided by sklearn. Additionally, the preprocessing of text to remove all words that are not nouns, adjectives or verbs is performed by the use of the 'spacy' python library. In this way, only words used to describe each myth are taken into consideration, reducing the presence of unimportant words and therefore results in better automatic labelling of each cluster.
Table 2.1 below shows two example clusters in which count vectors correctly identify clusters representing Arthurian legends and Underworlds that use a river to transport the dead to the afterlife.

Example Pages in a Cluster	Top 3 Words from Count Vectors
Camelot, Excalibur, King Arthur	Arthurian, Legend, Century
Charon, Erebos (Hades), Styx (river to Underworld)	River, Underworld, Dead

Table 2.1: Showing count vector labels

Additionally, TF-IDF (Term Frequency Inverse Document Frequency) was also implemented due to its ability to automatically ignore unimportant words by comparing the occurrences of a word within a cluster to its occurrences within a whole text corpus [27]. However, TF-IDF performed significantly worse, likely due to the discarding of words that are present in many clusters (e.g. 'God') but are still important to the labelling of individual clusters. Therefore, TF-IDF was not used in this project.

2.4.1 Cluster Proportions

Finally, to find similarities between cultures the proportion of cultural origins for each cluster must be determined. To achieve this, the categories provided on each Wiki page (described in Section 2.1) are used and the occurrences of each category in a cluster are counted. However, in this project, only categories that include reference to the cultural origin of a myth (e.g. Egyptian Mythology) are used to calculate the proportions, as only these are relevant to determine cultural similarities within a cluster.

In addition, the Wiki often separates myths into very specific cultural categories that are unhelpful in showing the cultural similarities within a cluster. To address this issue, multiple smaller categories were grouped to form broader categories based on their regional and cultural origin. For example, Berber, Yoruba and Ashanti mythologies are combined to form a broader category of African mythology, as shown in code snippet 2.5 below. The full list of mappings between cultures and their regions is provided in the appendix.

```
"African mythology": ["Berber mythology","Yoruba mythology","Ashanti mythology"],
"Middle Eastern mythology":["Islamic mythology","Jewish mythology","Arabian mythology","Levantine
    mythology","Magian mythology","Persian mythology","Semitic mythology", "Maltese folklore","
    Islamic demons"]
```

Code 2.5: Code to map smaller mythologies to larger regions

By combining the automatic labelling and calculation of cultural proportions, a detailed analysis of the cultural similarities can be achieved due to the ability to find cultural similarities within specific subsections of myths. For example, the cultural proportions for each cluster with labels representing Gods of the sea can be compared to find similarities between cultures at a higher level of detail than simply comparing the clusters of all Gods.

Initially, an alternative method to compare cultures within subsections of myths was considered that used the categories provided on each page to determine the types of myths present within a cluster. However, as the categories are determined by the writer of each page, the level of detail present within the categories varies significantly between pages. For example, the page of Ogma (Celtic God of speech) contains only the category of "Celtic Gods", whereas Aengus (Celtic God of youth and love) contains categories including "Celtic Gods", "Love and lust Gods" and "Gods of Poetry". Therefore, using the categories to label clusters relies heavily on the author of the page and would likely lead to poor classification of myths and, for this reason, this method was not used.

3 Results

Using the methods described above, each page from the Wiki was collected and clustered based on their semantic meaning. In this section, the analysis and evaluation of the generated clusters is presented, with the aim of answering the research questions proposed in Section 1.1.

3.0.1 Exploratory Analysis (RQ1)

To gain a better understanding of the Wiki, exploratory analysis was performed to find patterns and potential anomalies within the dataset. At the time of data collection there was a total of 3131 pages on the Wiki. However, after removal of gallery and disambiguation pages (explained in Section 2.1) a total of 2988 pages were collected. To find the distribution of myths between cultures, the percentage of myths from each cultural region within the dataset (explained in Section 2.4.1) were calculated. The results are displayed in the table and figure below. It is important to note that 245 myths on the Wiki do not have any specified cultural origin (e.g. the types of Magic), so do not contribute to the calculated proportions of each cluster.

Region	% of Data	Region	% of Data
Greek	17	Japanese	4
Norse	14	Egyptian	4
European	11	Middle Eastern	3
Indigenous American	6	Roman	3
North American	5	Asian	3
Celtic	5	Mesopotamian	2
Indian	4	Australian	2
South American	4	African	1
British	4	Chinese	1

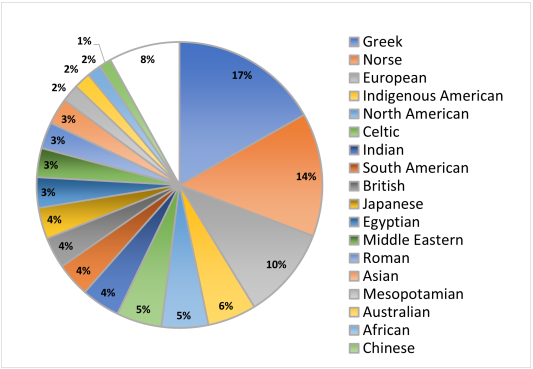

Table 3.1: Table and pie chart showing cultural distribution of the Wiki

Table 3.1 above shows that the distribution of myths varies significantly between cultural regions, with Greek, Norse and European cultures containing almost half of the total myths on the Wiki. However, other cultures such as Asian, Roman and African each contain only 2-3 percent of myths available on the Wiki. This imbalance shows a bias in the myths that are recorded on the Wiki and could lead to clusters that are skewed, with too much focus placed on cultures that have more myths on the Wiki[28]. This bias is reinforced in the level of detail for each myth, with pages describing myths from Greek and European cultures often containing more text with a higher level of detail when compared to myths from the smaller cultural regions. It is important to consider this bias when analysing the clusters generated, as the cultural proportions calculated will likely favour those cultures that have a much higher number of recorded myths on the Wiki.

The Wiki also contains pages that are labelled as stubs, that represent either very short or incomplete pages. These could potentially lead to noise within the data due to a lack of information for the natural language processing to analyse. However, they are included in the final clustering as there are currently 1210 stubs that account for over a third of the total myths on the Wiki. Therefore, the removal of stubs would lead to a considerable loss of information useful to this project. Many of these pages also appear to be incorrectly labelled as stubs (e.g. Ghoul and Vishnu) as they contain a large amount of information that the natural language processing can analyse. Furthermore, as HDBSCAN is not affected by outliers, the clustering is able to make use of the information in the incorrectly labelled stubs, whilst not being skewed by the stubs containing minimal information.

3.0.2 Clusters Formed (RQ2)

In this project, natural language processing and clustering was performed on both the summary tables and the main text present on the Wiki pages. This section provides analysis of the clusters formed from both sources of information with the aim to answer research question 2 (Can myths be separated into meaningful clusters?).

Clustering of Summary Tables

By using the methods previously described, the text contained within each table was successfully processed and separated into 55 clusters with a cluster confidence cost of 0.093. The low cost calculated shows that the clusters formed are of good quality and separate the myths with a high level of certainty (as explained in Section 2.3.2). However, for the clusters to be meaningful, they must not only be of good quality but also logically separate myths based on their semantic meanings. Table 3.2 below shows two of the generated clusters with examples of some pages they contain:

Pages in Cluster	Description of Page	Cultural Proportions
Amaterasu Ame-no-Ohabari Sesshō-seki Shani	Japanese Sun Goddess Legendary Japanese Sword Large Japanese Boulder Hindu God of Judgement	92% Japanese 8% Indian
Álfheimr Andvaranautr Skǫfnungr Hindarfjall	Norse Realm of Elves Magic Norse ring Sword of a Norse King Norse mountain	100% Norse

Table 3.2: Example clusters from the summary tables

It is clear from the above examples that clustering of the summary tables fails to separate myths into clusters of similar semantic meanings, with each cluster containing unrelated myths. Furthermore, the myths within each cluster tend to come from a single culture, so are not useful in this project as they do not show the similarities between cultures. This is consistent across all clusters, and the most likely explanation is because the majority of summary tables do not contain sufficient information for the word embeddings to effectively derive the meaning of each myth.

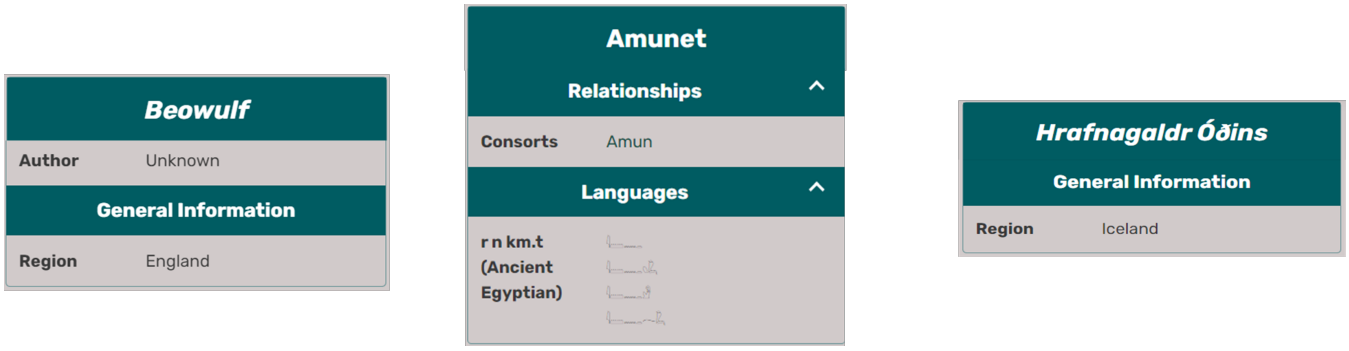

Figure 3.1: Example summary tables

Figure 3.1 above shows three example summary tables that provide very little useful information, with the only important information being the cultural origin, resulting in poor semantic separation of myths with clusters formed from singular cultural regions.

Clustering the Main Text

By using the sentence embedding and clustering methods (Sections 2.2.2 & 2.3.1) on the main text of each page, the myths were separated into 107 clusters with a confidence cost of 0.15, showing that the clusters formed are well-defined and of good quality. However, as before, it is important to the aims of this project to check that the clusters formed are meaningful and successfully separate myths based on their semantic meanings. Table 3.3 below shows some example clusters and their cultural proportions.

Cluster 3	**Proportions**
Heaven	Greek: 33%
Paradise	Asian: 17%
Ether (sky)	Mesopotamian: 17%
Seven Skies	Middle Eastern: 17%
Space	European: 17%
Cluster 16	**Proportions**
Vampire	European: 60%
Zombie	Celtic: 10%
Undead	Egyptian: 10%
Mummy	Asian: 10%
Alp	N. American: 10%

Cluster 21	**Proportions**
Elf	European: 31%
Fairy	British: 21%
Imp	Norse: 21%
Sprite	Celtic: 17%
Treefolk	Others: 14%
Cluster 37	**Proportions**
Blackbeard	N. American: 29%
Ghost ship	S. American: 21%
Black Caesar	European: 17%
El Caleuche	British: 17%
Flying Dutchman	Others: 17%

Table 3.3: Example clusters from the main text

The above examples clearly show that the clustering of the main text of each page was successful in creating meaningful clusters that separate myths based on their semantic meaning. Furthermore, each cluster combines myths from multiple cultures, allowing for the similarities of myths between different cultures to be derived. Therefore, the remainder of the results section focuses on the clusters generated from the main text due to their success in creating meaningful clusters to answer research questions 3 and 4.

3.0.3 Common Tropes and Similarities Between Cultures (RQ3 & RQ4)

By using the top five words from the count vectorisation, each cluster was successfully identified as belonging to a specific sub-genre of myths. These sub-genres vary significantly between clusters and cover a wide range of topics from zodiac constellations to malevolent creatures. However, a number of common tropes within the sub-genres were identified, with a large number of clusters representing myths of Gods, Creatures and Kings. Table 3.4 and the graph below show the 5 most common tropes together with the number and percent of clusters representing them.

Trope	**Number of Clusters**
Gods	34
Creatures	19
Kings	15
Hell and Demons	7
Spirits	6

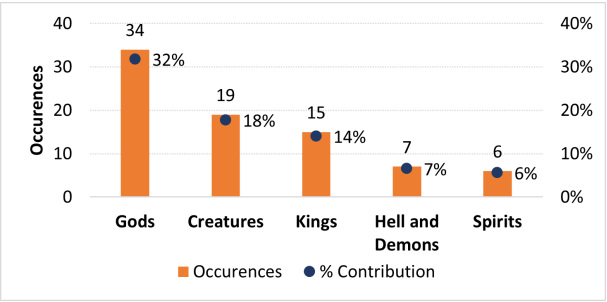

Table 3.4: Table and graph showing common tropes between clusters

The analysis of cultural similarities and differences therefore focuses on these tropes as they contain multiple clusters to examine.

Cultural Similarities Within Gods

Of the total clusters generated, approximately one third were categorised as representing myths centered around Gods, with every culture represented on the Wiki being present in at least one cluster. This section analyses these clusters as they provide key information on the similarities and differences between cultures. Figure 3.2 below shows 12 of the clusters representing Gods, displaying the specific category (e.g. Gods of Death) and the proportions of cultures that form each cluster. Furthermore, the clusters selected for analysis are those that contain Gods from multiple cultures, as other clusters created represent a singular culture and do not provide information that is useful to the aim of this project.

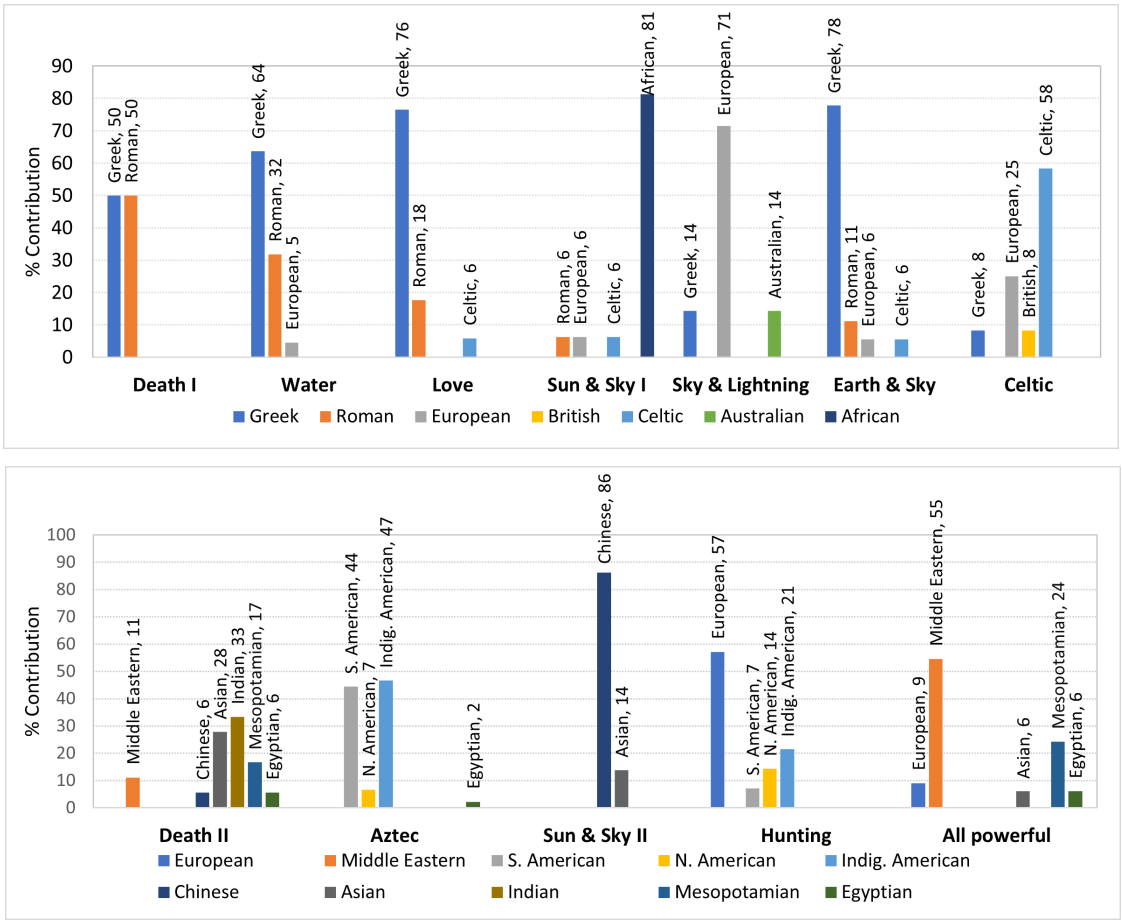

Figure 3.2: Example clusters about Gods

Figure 3.2 above shows a clear pattern between the geographical location and similarity of cultures, with Gods of cultures that are geographically close together having more similarities than those that are further apart. This is consistent between all clusters but is most clearly shown in the two clusters representing Gods of Death, with the first cluster (Death I) containing Gods from only Greek and Roman cultures, whilst the second (Death II) contains Gods from African and Asian cultures (e.g. Chinese, Mesopotamian). This observation is expected as cultures in closer geographical proximity are likely to have shared cultural influences, such as shared ancestry, languages and similar natural landscapes [29]. Furthermore, the increased contact and communication between cultures of similar locations leads to the merging of mythical beliefs between these cultures. This results in increased similarities between geographically close cultures. In addition, the figure above shows that American Gods are more closely related to European Gods than those of Asia or Africa. This may seem unusual at first, as America and Europe are not geographically close, but is likely explained by the early

migration of Europeans to America in the 1600s, leading to the spread and adoption of European myths into American culture [30].

It is important to note that similarities between cultures that do not co-exist within clusters can also be found, with common types of Gods (e.g. Gods of Death, Sun and Sky) identified across European, Asian and African cultures. A possible reason for this is that all cultures created Gods in order to explain shared life experiences (e.g. Death, the Sun and Sky), therefore resulting in worldwide similarities in the different cultural myths.

Cultural Similarities Within Creatures

Among the clusters generated, the trope of mythological creatures emerges as the second most common, representing 19 of the total 107 clusters. Figure 3.3 below shows 14 of these clusters, again selecting those that combine multiple clusters to display the similarities and differences between cultures.

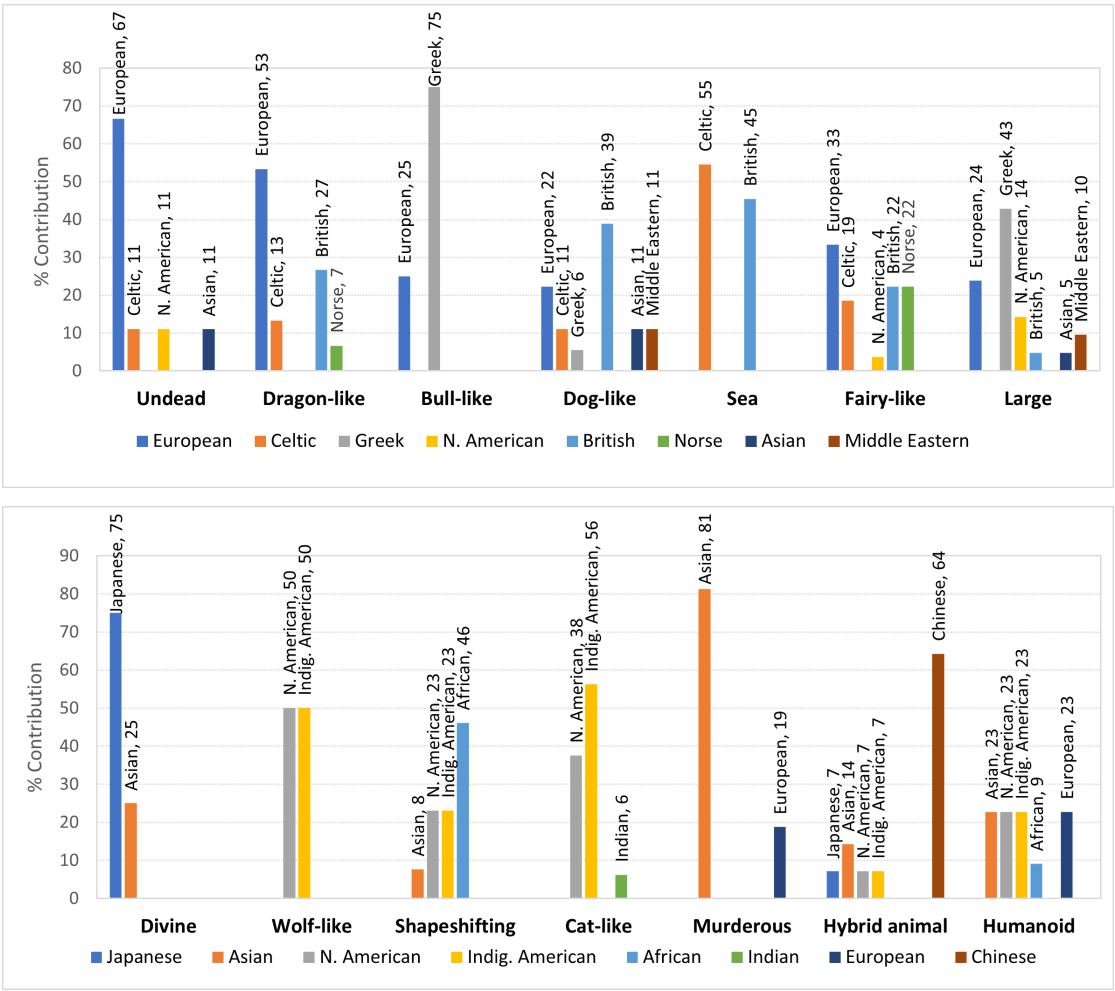

Figure 3.3: Example clusters about creatures

The clusters shown above demonstrate a similar pattern to those of the Gods previously described, with geographically closer cultures sharing more similarities than those that are located further apart. For example, the clusters for dragon and bull-like creatures are comprised solely from European (e.g. Celtic, British and Norse) cultures, whereas divine and hybrid animal creatures are comprised of

mainly Asian (e.g. Japanese and Chinese) cultures. However, unlike clusters of Gods, the geographical influence on mythological creatures does not appear to be as pronounced, with clusters for Undead and Large creatures combining myth from Asian, American and European cultures. However, further analysis into the clusters provides a possible explanation, as the types of mythological creatures in each culture tend to be based on the animals that exist within their respective environments. For example, the clusters representing wolf and bull-like mythological creatures are formed from American and European cultures respectively, both of which are native to their respective regions. In additon, humanoid creatures (i.e. clusters labelled fairy-like and humanoid) combine cultures from multiple world regions. This is likely due to the inherent and consistent humanoid characteristics across different cultures, leading to the development of similar mythological creatures across world regions.

It is also interesting that the cluster representing dragon-like creatures does not have any Asian influence, despite their prevalence in modern Asian culture. However, with further analysis into the clusters generated, it was found that the four Asian dragons present on the Wiki were instead clustered with Asian Gods. This clustering is justified as these Asian dragons (e.g. Shenlong, the Chinese God of wind and rain) are primarily representations of Asian Gods rather than mythical creatures.

Cultural Similarities Within Kings

The trope of Kings is the third most common, representing 7 of the 107 generated clusters. However, it was found that these clusters tend to focus on a single historic figure, combining all their familial relations (e.g. consorts, parents and children) to form clusters. For example, a cluster representing Irish Kings contains Cermait (High King of Ireland), Mac Cecht (Cermait's son) and Dagda (Cermait's brother). These clusters therefore tend to originate from a singular culture, providing no meaningful information into the similarities between cultures, and are therefore not analysed further within this project. However, these clusters could be useful within other areas of research that investigate the relations of royal families throughout history.

Cultural Similarities Within Demons and Spirits

Demons and spirits are the fourth and fifth most common tropes, representing 7 and 6 of the total clusters generated. Both of these tropes are analysed and compared as they represent similar aspects of mythology, with demons often described as being malevolent spirits. Figures 3.4 and 3.5 below show example clusters for demons and spirits respectively.

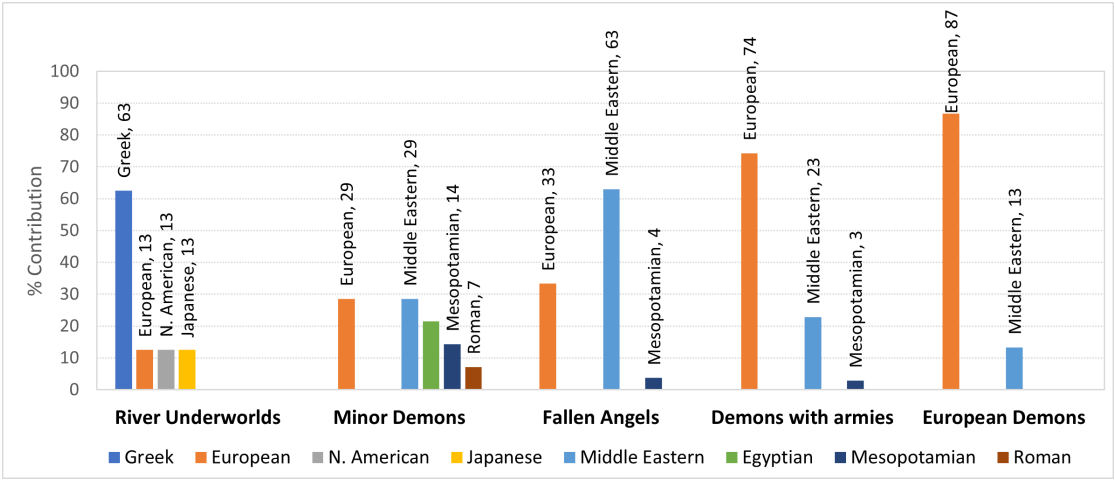

Figure 3.4: Example Clusters of Demons

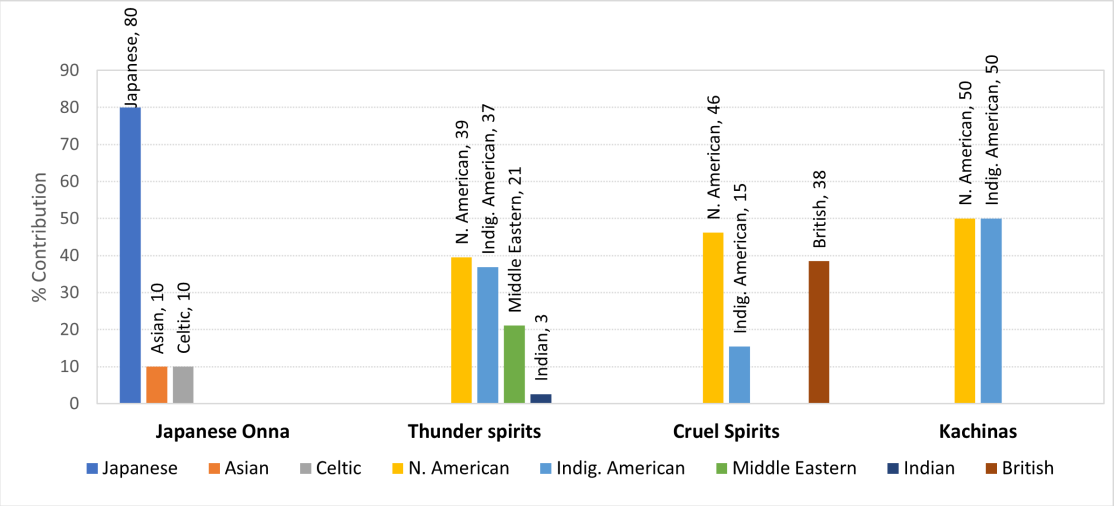

Figure 3.5: Example Clusters of Spirits

The clusters shown above again demonstrate a pattern between geographical proximity and cultural similarities, with cultures that are geographically closer having more similarities than those further apart. This is most clearly displayed in the differences of cultures within the two figures, with the clusters representing demons having a predominantly European origin, whereas the clusters representing spirits come from predominantly Asian and American cultures. However, the clusters representing spirits also contain myths of European (Celtic and British) origin. The most probable explanation for this is that the word Demon comes from European religious beliefs, whereas the term Spirit is a universal term used more widely across the world. Therefore, the difference in words used to describe these myths (i.e. Demon vs Spirit) may have resulted in potentially similar myths being placed into separate clusters.

4 Project Discussion

This project applied natural language processing and clustering methods on a dataset created by the automatic collection of pages from the Myths and Folklore Wiki, with the aim of finding similarities and differences between myths of different cultures. These relationships provide a greater understanding of cultural origins, thus leading to a more inclusive and accepting society. The five functional requirements (outlined in Section 1.1) were fulfilled, allowing for justified conclusions to be drawn for each of the four research questions. As the machine learning and clustering methods were successful in identifying the similarities and differences between cultures, the project was deemed to be a success.

Furthermore, the methods used successfully created well-defined and meaningful clusters that logically separated myths based on their semantic meaning. The common tropes within these clusters were identified and similar cultural patterns were found between each trope. As expected, geographically closer cultures showed greater similarities than those that are located further apart, suggesting accurate clustering between cultures.

4.0.1 Unexpected Results & Limitations

Whilst the project effectively identified similarities and differences between myths from different cultures, it encountered certain limitations that led to unexpected results. For example, despite Norse mythology having the second largest number of pages on the Wiki, the clusters showed minimal connections to other cultures with many clusters being solely Norse. This was an unexpected result as comparisons are often drawn between Norse, Greek and Roman myths, particularly those relating to Gods (e.g. Odin vs Zeus vs Jupiter). This may be attributed to the structure and content of the Norse pages on the Wiki, as these pages tend to describe myths based on their relations to other myths using many Norse words. For example, Odin is described as:

"*Odin is the King of Ásgarðr and is also the chief ruler (the Allfather) of the Æsir (the main pantheon of Norse gods) in Norse mythology. Óðinn is compared to Mercury by Tacitus. In wider Germanic mythology and paganism, Óðinn was known in Old English as Wōden, in Old Saxon as Wōdan, and in Old High German as Wuotan or Wōtan, all stemming from the reconstructed Proto-Germanic theonym *wōđanaz.*".

The natural language processing therefore struggled to capture the meaning of Norse myths and led to the generation of Norse clusters that focused on repeated unknown Norse words across pages (e.g. Æsir). In addition, the bias in the number of myths recorded for each culture (explained in Section 3.0.1) resulted in clusters that tended to be dominated by the larger cultures. It was therefore more difficult to derive similarities within the clusters, as the cultures with fewer myths within that cluster were more likely to have been overlooked.

A limitation in the automatic labelling of clusters was also found, as the larger clusters struggled to identify the key words used in the description of the myths within the cluster. A possible explanation for this limitation is that larger clusters encompass a greater number of myths, resulting in a much larger pool of words to analyse. Therefore, identifying the five most important words by occurrence becomes less accurate as a result of increased noise within the words to be selected.

4.0.2 Future Research

To address the limitations introduced by the imbalance in the number of myths recorded for each culture, consideration could be given to using re-sampling methods that have been designed to test cluster stability within imbalanced datasets. These methods work by taking random samples from the larger categories to equalise the proportions of data prior to clustering, and testing the cluster stability by comparison of multiple runs [31, 32]. With such further analysis, it would be interesting to see if the influence of the larger cultures within clusters would change with the use of these re-sampling clustering methods. This may result in the discovery of new links and similarities between cultures when compared to traditional clustering methods.

Additionally, to address the limitation of labelling larger clusters, topic modelling methods could be used. These methods work by identifying clusters of words within text to establish common themes and topics within documents [33]. Accurate determination of the key themes using topic modelling requires a large volume of text that is not present in most clusters. Therefore, this approach was not initially considered as a viable cluster labelling method for this project. However, in the larger clusters that have more volume of text, topic modelling may yield better results than count vectorisation and tf-idf, as there is more text available for analysis.

Within this project, the cultural regions used to find cultural similarities across the world combined many smaller cultures (e.g. European combines Baltic, German, Romanian, Swedish etc). Further

investigations could therefore be conducted to limit the analysis to a singular cultural region. This would provide deeper insights into the specific cultures of the larger regions, therefore showing similarities between individual cultures rather than world regions. For example, by analysing only the myths from Europe, the similarities between each European culture may be found rather than collectively being labelled as belonging to Europe.

The Myths and Folklore Wiki is part of a family of Wikis provided by Fandom. Fandom has websites dedicated to over 385,000 different topics, ranging from popular films, video games and artists to medicine, science and art. As all Wikis provided by Fandom have the same HTML document structure, the methods of data collection, natural language processing and clustering used in this project could therefore be used for any other Wiki provided by Fandom with minimal code editing. This opens up a wide variety of future applications for this project to find the similarities and differences between thousands of different topics. The cultural similarities identified in this project could therefore be compared to the findings from other Wikis to provide insights into how myths have influenced modern popular culture including arts, music, television and films of across cultures.

References

[1] Steven J. Heine and Matthew B. Ruby. "Cultural psychology". In: *WIREs Cognitive Science* 1.2 (2010), pp. 254–266.

[2] Vasyl Shynkaruk, Salata Galyna, and Tatiana Danilova. "MYTH AS A PHENOMENON OF CULTURE". In: *National Academy of Managerial Staff of Culture and Arts Herald* (Nov. 2018).

[3] Ali Al-Haidari. "Meaning, Origin and Functions of Myth: A Brief Survey". In: (May 2012).

[4] Fatih Mehmet Berk. "The Role of Mythology as a Cultural Identity and a Cultural Heritage: The Case of Phrygian Myhtology". In: *Procedia - Social and Behavioral Sciences* 225 (2016). Conservation of Architectural Heritage (CAH), pp. 67–73.

[5] Theo Meder et al. "Automatic Enrichment and Classification of Folktales in the Dutch Folktale Database". In: *The Journal of American Folklore* 129.511 (2016), pp. 78–96. (Visited on 05/05/2023).

[6] David Hirshleifer and Joshua B. Plotkin. "Moonshots, investment booms, and selection bias in the transmission of cultural traits". In: *Proceedings of the National Academy of Sciences* 118.26 (2021).

[7] William G. Doty. "Mythophiles' Dyscrasia: A Comprehensive Definition of Myth". In: *Journal of the American Academy of Religion* XLVIII.4 (Dec. 1980), pp. 531–562.

[8] Carmelo Cassisi et al. "Similarity measures and dimensionality reduction techniques for time series data mining". In: *Advances in data mining knowledge discovery and applications* (2012), pp. 71–96.

[9] Dionisios N. Sotiropoulos et al. "Machine Learning in Intangible Cultural Analytics: The Case of Greek Songs' Lyrics". In: *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. 2021, pp. 299–305.

[10] Jacob Werzinsky, Zhiyan Zhong, and Xuedan Zou. "Analyzing Folktales of Different Regions Using Topic Modeling and Clustering". In: *arXiv preprint arXiv:2206.04221* (2022).

[11] Cosima Rughinis. "Citizen science, gallaxies and tropes: Knowledge creation in impromptu crowd science movements". In: Sept. 2016, pp. 1–6. DOI: 10.1109/RoEduNet.2016.7753210.

[12] Leonard Richardson. *Beautiful soup documentation*. 2007.

[13] Milan Straka et al. "Sumeczech: Large czech news-based summarization dataset". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[14] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. "Exploiting similarities among languages for machine translation". In: *arXiv preprint arXiv:1309.4168* (2013).

[15] Felipe Almeida and Geraldo Xexéo. "Word embeddings: A survey". In: *arXiv preprint arXiv:1901.09069* (2019).

[16] Oren Barkan et al. "Scalable attentive sentence pair modeling via distilled sentence embedding". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 3235–3242.

[17] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[18] Lahbib Ajallouda, Kawtar Najmani, Ahmed Zellou, et al. "Doc2Vec, SBERT, InferSent, and USE Which embedding technique for noun phrases?" In: *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*. IEEE. 2022, pp. 1–5.

[19] Mebarka Allaoui, Mohammed Lamine Kherfi, and Abdelhakim Cheriet. "Considerably improving clustering algorithms using UMAP dimensionality reduction technique: a comparative study". In: *Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4–6, 2020, Proceedings 9*. Springer. 2020, pp. 317–325.

[20] Djuradj Milošević et al. "The application of Uniform Manifold Approximation and Projection (UMAP) for unconstrained ordination and classification of biological indicators in aquatic ecology". In: *Science of The Total Environment* 815 (2022), p. 152365.

[21] Dongkuan Xu and Yingjie Tian. "A comprehensive survey of clustering algorithms". In: *Annals of Data Science* 2 (2015), pp. 165–193.

[22] Lianhui Wang et al. "Ship AIS trajectory clustering: An HDBSCAN-based approach". In: *Journal of Marine Science and Engineering* 9.6 (2021), p. 566.

[23] M Steinbach, V Kumar, and P Tan. "Cluster analysis: basic concepts and algorithms". In: *Introduction to data mining, 1st edn. Pearson Addison Wesley* (2005).

[24] Davoud Moulavi et al. "Density-based clustering validation". In: *Proceedings of the 2014 SIAM international conference on data mining*. SIAM. 2014, pp. 839–847.

[25] Pratheep Kumar, Geetha G Nair, et al. "An efficient classification framework for breast cancer using hyper parameter tuned Random Decision Forest Classifier and Bayesian Optimization". In: *Biomedical Signal Processing and Control* 68 (2021), p. 102682.

[26] Tinu Theckel Joy et al. "Hyperparameter tuning for big data using Bayesian optimisation". In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE. 2016, pp. 2574–2579.

[27] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[28] Wei-Chao Lin et al. "Clustering-based undersampling in class-imbalanced data". In: *Information Sciences* 409-410 (2017), pp. 17–26. ISSN: 0020-0255.

[29] Sonja A Sackmann and Margaret E Phillips. "Contextual influences on culture research: Shifting assumptions for new workplace realities". In: *International Journal of Cross Cultural Management* 4.3 (2004), pp. 370–390.

[30] Alden T Vaughan. *New England Frontier: Puritans and Indians, 1620-1675*. University of Oklahoma Press, 1995.

[31] Hans-Joachim Mucha and Hans-Georg Bartel. "Resampling techniques in cluster analysis: is subsampling better than bootstrapping?" In: *Data science, learning by latent structures, and knowledge discovery*. Springer. 2015, pp. 113–122.

[32] Irina M Gana Dresen et al. "New resampling method for evaluating stability of clusters". In: *BMC bioinformatics* 9 (2008), pp. 1–10.

[33] Pooja Kherwa and Poonam Bansal. "Topic modeling: a comprehensive review". In: *EAI Endorsed transactions on scalable information systems* 7.24 (2019).

Appendix

```
for item in text:
        add = True
        headers = []
        headers.append(item.parent.find_previous_sibling(['h2','h3']))
        headers.append(item.find_previous_sibling(['h2','h3']))

        if (item.has_attr('class')):
            add = False

        elif (item.find(class_="portable-infobox pi-background pi-border-color pi-theme-wikia pi-layout-default") != None):
            add = False


        else:
            for parent in item.parents:
                if (parent.has_attr('class')):
                    if(parent['class'][0] in ["portable-infobox", "references"]):
                        add = False


        if (len(headers) != 0):
            for header in headers:
                if header != None:
                    if (any(notCol for notCol in notCollect if (notCol in header.get_text().lower())):
                        add = False



        if add == True:
            if (item.get_text() != "\n"):
                mainText.append(item.get_text().strip('\n'))
```

Code 4.1: Code snippet for selecting important text

```
mythologyByArea = {  "African mythology": ["Berber mythology","Yoruba mythology","Ashanti mythology"],
                     "Australian mythology":["Australian Aboriginal mythology","Māori mythology",'Aboriginal gods'],
                     "Asian mythology":["Buddhist mythology","Elamite mythology","Indonesian mythology","Korean mythology",
                                 "Philippine mythology","Tibetan mythology","Philippine creatures",'Buddhist demons','Buddhist folklore',
                                 'Philippine demons'],
                     "European mythology":["Baltic mythology","Basque mythology",'German mythology',
                                 "Christian mythology","Ars Goetia Demons","Dacian mythology","Etruscan mythology","Finnish mythology",
                                 "French mythology","Galician mythology","Germanic mythology","Portuguese mythology",'Etruscan goddesses',
                                 "Romanian mythology","Slavic mythology","Spanish mythology","Christian folklore",
                                 "Danish folklore","French folklore","German folklore","Icelandic folklore","Norwegian folklore",
                                 "Scandinavian folklore","Swedish folklore","Nymphs",'European dragons','Medieval European creatures',
                                 'French creatures','Germanic weapons','Armenian mythology','Christian demons','Germanic heroic legends',
                                 'Dutch mythology'],
                     "Middle Eastern mythology":["Islamic mythology","Jewish mythology","Arabian mythology","Levantine mythology",
                                      "Magian mythology","Persian mythology","Semitic mythology", "Maltese folklore","Islamic demons"],
                     "Chinese mythology":['Chinese creatures'],
                     "Japanese mythology":["Japanese folklore","Japanese weapons","Japanese items",'Kitsune'],
                     "Indian mythology":["Hindu mythology","Women in Hindu mythology",'Hindu goddesses'],
                     "Celtic mythology":["Galician mythology","Manx folklore","Irish mythology","Irish gods","Irish kings","Irish folklore",
                                 'Celtic goddesses','Celtic gods'],
                     "Egyptian mythology":['Egyptian creatures'],
                     "British mythology":["English mythology","Scottish mythology","Welsh mythology","British folklore","English folklore",
                                 "Manx folklore","Scottish folklore",'English creatures','Arthurian legends'],
                     "Mesopotamian mythology":["Mesopotamian Mythology",'Ishtar'],
                     "Indigenous American mythology":["Aztec mythology","Native American mythology","Native American folklore"],
                     "North American mythology":["Cajun mythology","Caribbean mythology","Hawaiian mythology","Hopi mythology",
                                      "Inuit mythology","Maya mythology","Mexican mythology","Māori mythology",
                                      "Polynesian mythology","Oceanian mythology","Zuni mythology","American folklore"],
                     "South American mythology":["Aztec mythology","Brazilian mythology","American folklore","Brazilian folklore",
                                      "Latin American folklore","South American folklore"],
                     "Classical mythology":[],
                     "Greek mythology":['Greek creatures',"Greek people", "Greek figures","Greek heroes","Greek goddesses","Greek gods",
                                 "Greek items",'Greek deities'
                                 "Greek monarchs","Demigods from Greek myths","Peoples in Greek mythology","Achaeans","Daemons",
                                 "Heracles",'Iliad characters', 'Trojans','Greek locations','Greek events','Twelve Olympians',
                                 'Mortal demigods from Greek myths','Mortals from Greek myths','Greek poems','Greek giants',
                                 'Greek monarchs'],
                     "Roman mythology":['Roman creatures'],
                     "Norse mythology":['Æsir','Norse figures','Norse weapons','Norse locations', 'Norse realms','Norse items',
                                 'Old Norse studies scholars','Runes']
}
```

Figure 4.1: Mappings from cultures to regions