

**ECM3401**

Individual Literature Review and Project (A, TRM1+2 2022/3)



1080831



Marker, please refer to marking guidelines at <http://ex.ac.uk/dyslexia-marking-guidelines>

**235897**

**Coursework:** Literature Review

**Submission Deadline:** Wed 23rd Nov 2022 12:00

**Personal tutor:** Dr Diego Marmsoler

**Marker name:** N/A

700013809

**Word count:** 0

By submitting coursework you declare that you understand and consent to the University policies regarding plagiarism and mitigation (these can be seen online at [www.exeter.ac.uk/plagiarism](http://www.exeter.ac.uk/plagiarism), and [www.exeter.ac.uk/mitigation](http://www.exeter.ac.uk/mitigation) respectively), and that you have read your school's rules for submission of written coursework, for example rules on maximum and minimum number of words. Indicative/first marks are provisional only.



# Machine Learning Approaches for the Analysis of Mythology and Folklore


Michael Hills

## Abstract

This paper outlines the study of mythology and folklore using techniques from machine learning and cultural analytics. The dataset used for analysis is the Mythology and Folklore Wiki, an open and easy access website for the collection and study of mythological and folkloric traditions. The methods utilised in the field of cultural analytics are covered and a critical analysis of their potential effectiveness on the Wiki's data are discussed. The evaluation strategies for each of the project requirements and a plan for the completion of the project are presented.

The Mythology and Folklore Wiki can be found here: [https://mythus.fandom.com/wiki/Main\\_Page](https://mythus.fandom.com/wiki/Main_Page)

I certify that all material in this dissertation which is not my own work has been identified.

Signature: 

---

# 1. Introduction

Throughout time, culture has greatly impacted the human way of life, with behavioural patterns, values and principles all being influenced by an individual's upbringing and cultural beliefs. This has led to growing research into the many different cultural backgrounds of human civilisation, with studies investigating the different aspects of each culture, such as its history, arts, stories, and social customs. Such studies provide an understanding of how cultures develop and the effects they have on people and society [1][2]. Traditionally, research has been undertaken by closely analysing written text by hand. This is a time-consuming process, and the focus is often only on specialised subsets of materials, presenting a challenge to find the broader links and themes between different cultures around the world [3]. The issue has been compounded over time by the increased interest in digitising cultural datasets resulting in a larger number of narratives available online for research [4]. It is therefore impractical to closely analyse even a small percentage of the available material [3].

Cultural analytics aims to improve analysis of digitised cultural datasets by using modern computational and visualisation methods to analyse large datasets and gain insight quickly and automatically with minimal human interaction. This allows for deeper analysis across broader areas of research and also removes any subjective bias from prior knowledge or preconceived connections between cultures [5]. The use of cultural analytics therefore allows for new unseen and harder to find connections between cultures to be uncovered [4].

Mythology and folklore are often viewed as the building blocks of culture and show how old beliefs and practices help to form a cultural identity over time. This opens up an interesting question, to see how different myths spread throughout the world and how they have been adjusted and reinterpreted to form different cultures [6]. This project aims to answer this question through the means of collecting data from the *Myths and Folklore Wiki* and using computational tools to find similarities between cultural beliefs around the world to help understand its spread and effects on modern cultural identity. This is a small field in its own right, often referred to as computational folkloristics [3]. Although there are projects that aim to solve similar problems, they often focus on a single genre of mythology, such as the analysis of Dutch folklore or the case of Greek songs [4][1]. In addition, other published projects looking at a wider range of mythology often have shallow analysis that fail to offer conclusions with meaningful information [7]. These gaps in knowledge open up an exciting research opportunity to improve on the research that has already been undertaken.

# 2. Methods

The foundation of computational folkloristics makes use of modern technological developments allowing for the collection and analysis of large amounts of data with relative ease. This section will cover the key computational tools used in folkloristics that collect and navigate large datasets to help develop meaningful conclusions.

## 2.1. Collection of Data

**Paimon**

**Paimon** or **Paymon** is one of the Kings of Hell, more obedient to Lucifer than other kings are, and has two hundred legions of demons under his rule. Paimon is depicted as a man with an effeminate face, wearing a precious crown, and riding a dromedary. Before him often goes a host of demons with the shape of men, playing trumpets, cymbals, and any other sort of musical instruments.

[Contents](#) [\[show\]](#)

**Overview**

He has a great voice and roars as soon as he comes, speaking in this manner for a while, until the conjurer compels him and then he answers clearly the questions he is asked. When the conjurer invokes this demon he must look towards the northwest, for there is where he has his house, and when Paimon appears he must be allowed to ask what he wishes and be answered, in order to obtain the same from him.

Paimon teaches all arts, philosophy and sciences, and secret things; he can reveal all mysteries of the Earth, wind and water, what the mind is, and everything the conjurer wants to know, gives good familiars, dignities and confirms them, binds men to the conjurer's will.

Paimon	
<b>Ranking</b>	King
<b>Commander</b>	200 Legions
General Information	
<b>Species</b>	Demon
<b>Form</b>	Human
<b>Domain</b>	Art, Philosophy, and Science Secret Knowledge Familiars Restoring Dignity Binding Others to One's Will

Fig. 1. Example page from Myths and Folklore Wiki

The Myths and Folklore Wiki is an open, easy access website for the cataloguing and study of mythological and folkloric tales and traditions. The site currently contains 2968 pages and the aim of this project is to extract information from each page and use machine learning techniques to find connections between different cultures and their mythologies.

Each page in the Wiki is structured slightly differently. However, the majority have the same broad characteristics with a summary style table containing the key features of each myth and a short paragraph providing an overview (Figure 1). The use of software packages such as Beautiful Soup allow for easy traversal of the HTML elements on each page and the common features can be selected and key information extracted [8]. This information is then stored in a format easily readable by a computer, such as a JSON file, for later analysis.

## 2.2. Word Embeddings

A word embedding is a learned representation of a word as a vector made up of hundreds of dimensions in which words with similar semantic meanings create vectors of similar values [9]. An illustration is shown in Figure 2 where the words “lecture” and “talk” have similar vector values, since the meanings of both words are closely related.

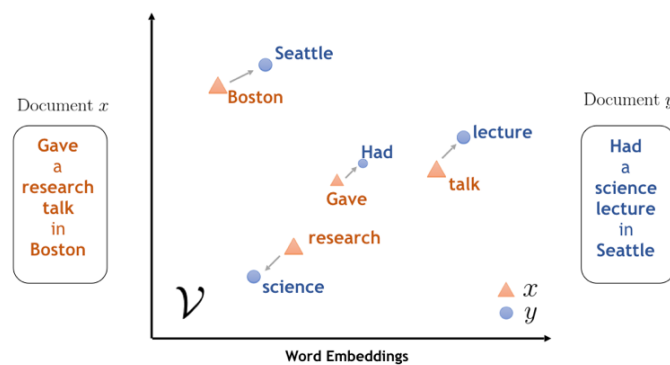


Fig. 2. IBM: Illustration of word embeddings

There are many models to create word embeddings, each being trained and developed with different methods. However, these different models can be broadly split into two categories: count-based embeddings and prediction-based embeddings [10].

### 2.2.1. Count-based Embeddings

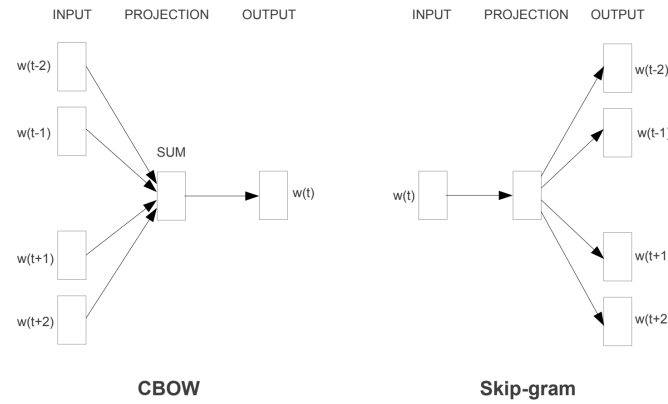
Count-based embeddings focus on the concept that similar words will be used in similar contexts, so will often be written alongside a shared subset of words [10]. For example, “cat” and “dog” will likely be used alongside the words “pet” or “animal” so will have similar vector representations. The vectors are calculated by taking the whole text corpus and creating a co-occurrence matrix that counts every time words are used together [11]. For example, if the corpus consists of the two sentences; “I have a pet cat” and “I love my pet dog”, the following matrix is obtained, with the columns representing the vector for the corresponding word.

	-	Have	Love	Pet	Cat	Dog
D1	1	0	1	1	0	
D2	0	1	1	0	1	

Count-based embeddings have been successful in using a simple algorithm to maintain semantic relationships between words. However, due to the need for large amounts of memory to store the co-occurrence matrix and the development of newer and better performing techniques, prediction-based embeddings are now more commonly used [12].

### 2.2.2. Prediction-based Embeddings

Prediction-based embeddings utilise a neural network to predict, rather than count, words that are commonly used together. There are multiple ways to achieve this, with two of the most common being CBOW and Skip-Gram.



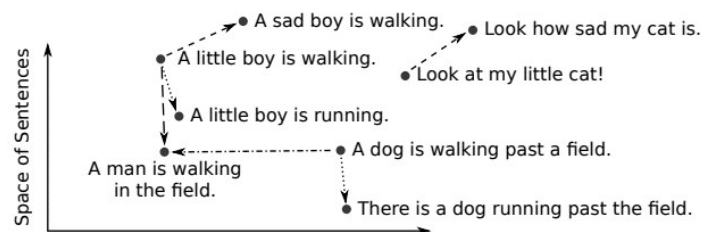
**Fig. 3.** CBOW & Skip-Gram

Both methods train the neural network to learn the probabilities of words appearing in the same context. These probabilities are used to create the word embedding vectors, with words of similar meanings having similar probabilities and hence similar vectors. However, the key difference between these two methods is the part of the text that is being predicted. CBOW uses the surrounding words as input to predict a central word based on the context. In contrast, Skip-Gram uses one central word as input to predict the surrounding words, as shown in the illustration in Figure 3 [10].

Both CBOW and Skip-Gram have been successful in the use of word2vec, and in this project could be used on the summary table of each myth to provide a way to calculate the similarities of their key factors. However, it should be noted that in the original paper by Mikolov, it is shown that Skip-Gram works well with smaller datasets and better represent rarer words, whereas CBOW tends to train faster and gives better results with more common words [13]. Skip-Gram may therefore be better suited to this project as the words used in the summary tables tend to be rarer and found only in myths and folktales.

### 2.3. Sentence Embeddings and Transformers

In order to analyse the larger forms of text on the Wiki, such as the overview sections on each page, sentence embeddings can be used. This is an extension on the ideas of word embeddings, with the key difference being whole sentences are vectorized rather than individual words. The sentence vectors are created based on the semantic meaning of the whole sentence, so sentences of similar context will create similar vectors, as illustrated in Figure 4 [14].



**Fig. 4.** Illustration of Sentence Embeddings

There are many different methods to achieve this, with the simplest merely averaging the word embeddings in each sentence [15]. However, in recent years with the advancements in deep learning and the introduction of the transformer model in 2017, semantic understanding of text has significantly improved, as shown by the success of models such as BERT[16]. This is due to their ability to process

the entire sentence at once rather than just one word at a time, allowing for quicker training time due to increased parallelism and a greater understanding of the text [16].

Sentence embeddings are therefore a key tool for machines to understand natural language, and will be extremely helpful in evaluating which myths have similar semantic descriptions through analysis of their corresponding sentence vectors. Links between myths of different cultures can therefore be easily explored with further computational analysis.

## 2.4. Clustering Algorithms

A clustering algorithm is an unsupervised machine learning algorithm that automatically groups data points that share similar properties [17]. In this project, the properties used for clustering will be the vectors created through the word and sentence embeddings of each Wiki page. This results in clusters that outline the semantic similarities found between myths and allows questions to be answered regarding the correlation between mythologies of different cultures. There are many different types of clustering algorithms, each with their own advantages, disadvantages and use cases.

### 2.4.1. Types of Clustering Algorithms

Centroid clustering is a technique that makes use of points at the centre of each cluster called a centroid, with the most popular example being K-means clustering. Data points are assigned to the cluster with the closest centroid, and these centroids are moved incrementally to remain in the centre of its respective cluster [17]. This is a relatively simple method and is computationally inexpensive. However, it struggles with clusters that are not spherical as it uses a simple distance measurement to assign clusters. In addition, centroid clustering struggles to cluster data containing outliers as this will result in the centroids being moved away from the centre of each cluster [17]. Centroid clustering is therefore unlikely to work well when clustering vectors from the Myths and Folklore Wiki as the clusters are unlikely to be spherical and there will be outliers in myths that are unique to a single culture.

Distribution-based clustering uses a distribution model, such as Gaussian distribution. This means that the further from a cluster's centre, the lower the probability a point is part of as cluster. As this method uses probability to determine clusters, more realistic clustering is achieved as the output is not a binary "yes" or "no". However, since the distribution of the Wiki dataset is not known, the clustering will likely be sub-optimal [18].

Density-based clustering detects areas where data points are highly concentrated and creates clusters for these areas, with these clusters being separated by areas of low concentration which are labelled as noise. Therefore, this method allows for varying and arbitrary-shaped clusters as there is no central point required to determine each cluster [17]. In addition, the algorithm is also not skewed by outliers as they are not assigned to a cluster. However, density-based clustering tends to perform poorly with data of varying densities due to difficulty in finding the edges of a cluster [18]. This method could therefore be useful in this project as it will be able to find oddly shaped clusters. If, however, the vectors created from embeddings are of varying density, the algorithm may struggle, and a different method will need to be considered.

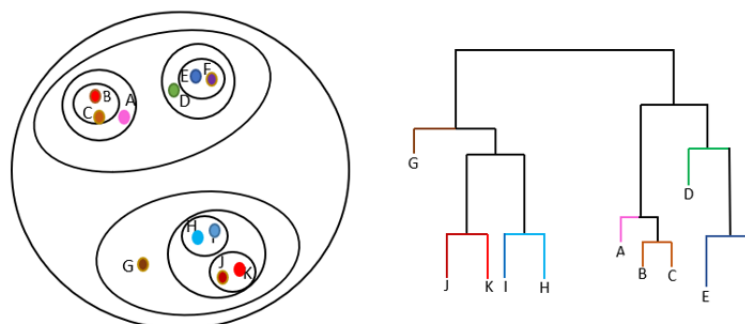


Fig. 5. Illustration of Hierarchical Clustering

Hierarchical clustering uses a tree of nested clusters, that starts with each data point contained in its own cluster. The two most neighbouring clusters are then merged and this is repeated until there is one cluster containing all points. The tree generated (example in Figure 5) shows how the clusters have been merged over time, and the user can decide a position to cut the tree for the desired size and number of clusters [18]. This is a key advantage over other clustering techniques, and will likely be helpful for this project as the types and sizes of clusters is unknown, so being able to choose could give better results. However, hierarchical clustering is computationally expensive and results tend to include outliers.

In addition, algorithms exist that combine multiple methods of clustering. One such algorithm is HDBSCAN, that uses density based clustering in a hierarchical manner, allowing for the benefits of both methods to be combined. Therefore, HDBSCAN easily finds clusters of varying shapes due to the benefits of density based clustering, and also improves on clustering of data with varying densities due to the hierarchy created [19]. An algorithm that combines multiple types of methods may therefore provide the best results when clustering the data from the Wiki.

## 2.5. TF-IDF

TF-IDF, or Term Frequency-Inverse Document Frequency, is a method to determine how relevant a word is in a document. This is achieved by calculating how many times a word appears in a document compared to the whole text corpus. For example, if a word appears multiple times in one document, but very rarely in the whole corpus, the word will be deemed important [20]. This is commonly used to interpret clusters by finding important words that formed the clusters, such as its use in BERTopic [21] [22]. The TF-IDF methodology may also be applied to this project to help evaluate clusters. For example, if the key words used in a cluster include "deity", "God", "underworld" and "hell", it is likely the cluster represents the rulers of hell for each culture.

## 3. Aims and Objectives

To successfully complete a project that delivers meaningful conclusions from the data on the Myths and Folklore Wiki, there are many requirements that must first be fulfilled.

### 3.1. Collect all Data

Before any computational analysis can be undertaken, the data must be collected using methods described in Section 2.1. The website currently contains 2968 pages, so the collection is complete when the dataset contains approximately 2968 pages. This number, however, cannot be exact as there are some pages that consist only of a gallery containing more images of a myth that could not fit onto its main page. Other pages also exist that only consist of links to other pages, such as the "Talents of witches" page, containing links to Alchemy, Spellcasting, Summoning etc. Neither provide useful information to cluster so there is no need to collect data from these pages.

In addition, not every page features the summary table and overview mentioned in Section 2.1. Therefore, some experimentation will be needed when it comes to collecting sufficient data from each page. To resolve this, one potential option is to implement a user decision. For example, if a page does not have an overview section, the user is given a list of the other sections to choose from to use in its place (e.g. Etymology, Myths and Legends etc). This ensures sufficient data is collected to allow for good quality clusters to be generated.

### 3.2. Create Embeddings

Once the data has been collected, the word and sentence embeddings can be created. There are many different methods and models which can be used, with word2vec and SBERT being two of the most common word and sentence embeddings respectively [22]. However, there are many other pre-trained embedding models available and to find the most suitable for this project will require experimentation. However, it is hard to evaluate the success of an embedding until later computation is performed. In the case of this project, the tendency of the vectors to form clusters, discussed later in Section 3.3.1, helps show if the vectors created are useful or if another model should be used to obtain better results.



### 3.3. Create and Evaluate Clusters

Clusters will then be created using the vectors generated in the word and sentence embeddings. However, it is vital that these clusters are well defined and separate the data correctly to be able to draw any meaningful information. To test this, there are two main methods that evaluate the success of clustering. These are tendency to cluster and cluster quality.

#### 3.3.1. Tendency to Cluster

Before creating clusters, it is important to check the data has a tendency to cluster, meaning the data naturally fits into groups. If there is no clustering tendency, the clustering algorithm will fail to create meaningful results, regardless of the algorithm used. To measure the clustering tendency, statistical methods such as the Hopkins test to assess the spacial randomness of the data points can be used[17]. This test compares the distances between the data points to be clustered and a new random distribution of points of the same variation. The comparison results in an H value, and values close to and lower than 0.5 have little to no clustering tendency whilst values closer to 1 have a high clustering tendency[23]. This is illustrated in Figure 6, in which an H value of 0.47 shows data with no viable clusters, whereas larger H values show increasing distinction between clusters as H increases.

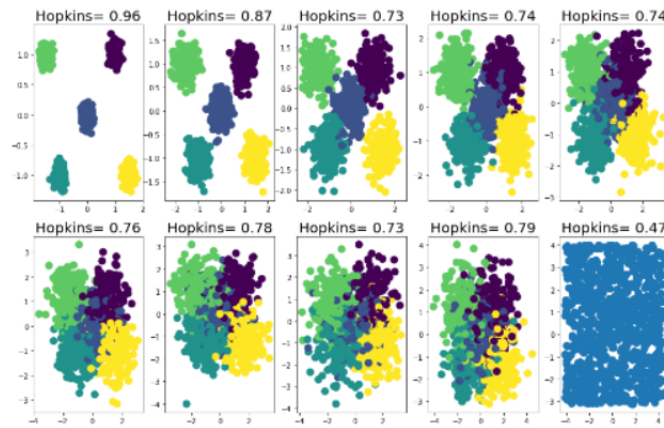


Fig. 6. Illustration of Hopkins Values

In this project, if the Hopkins value created before clustering shows no tendency, the sentence and word embeddings will need to be computed again with a different model, as embeddings have failed to separate the different meanings of each page.

#### 3.3.2. Cluster Quality

Once the clusters are created, they must be validated to ensure they are of good quality, are well defined, and do not include outliers that could skew conclusions. There are several methods to achieve this, with calculation of the silhouette coefficient being one of the most popular. The following steps are used to compute the silhouette coefficient for an individual point:

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  $s_i = (b_i - a_i) / \max(a_i, b_i)$  [17]

The silhouette coefficient for each point calculated is between -1 and 1, and can be averaged for every point within a cluster to determine the quality of a whole cluster. Negative values indicate bad quality clustering as this means the average distance between points within a cluster is larger than the distance between clusters. An ideal value is 1, meaning minimal distance between points within a cluster [17].

The silhouette coefficient, therefore, makes it easier to determine if the individual clusters are well defined and of good quality. Calculating the silhouette coefficient over different numbers of clusters helps determine the optimum number of clusters to form, or where to cut a hierarchical clustering for the ideal number of clusters. This methodology is used in a paper published by Springer [24], that outperformed three other algorithms in finding the optimal number of clusters. This allows for more meaningful clusters to be created that leads to better conclusions from the data.

### 3.4. Extract Meaningful Information

To extract useful data from the clusters, there needs to be a way to identify the features that each cluster represents, and this will vary depending on the question to be answered. For example, to see which culture's mythologies are most closely related, there is a tag on each page displaying the culture of origin. This could therefore be used to label each data point and the proportion of points from each culture that coexist within clusters will show the extent to which cultures are closely related.

In addition, using TF-IDF described in Section 2.5, questions can be answered relating to how the clusters have formed based on similar myths, such as finding the most common types of myth for each culture or finding which cultures contain similar narratives, e.g. a story of a great flood. Data visualisation tools, such as Matplotlib and Seaborn, can also be used to create charts that are easily interpreted to answer such questions. If the resulting graphs do not give clear results, experimentation with the previous embeddings and clustering will be needed to test alternative methods.

## 4. Project Management

The project consists of multiple tasks and deliverables that are outlined below:

1. Collect data from all pages on the Myths and Folklore Wiki and store as a JSON file containing around 2968 entries.
2. Create word embeddings from the summary tables.
3. Create sentence embeddings from the main larger text sections.
4. Cluster the embeddings using methods described in Section 2.4
5. Evaluate cluster quality using methods described in Sections 3.3.1 and 3.3.2. If the quality tests show poor clustering, try different methods for the embeddings and clustering, described in Sections 2.2-2.4.
6. Label clusters using TF-IDF.
7. Visualise clusters and extract information.
8. Prepare a first draft of the report.
9. Complete the final version of the report.
10. Record a 20 minute presentation.

Section 4.3 shows a timeline of these deliverables with their approximate due dates.

### 4.1. Ethical and Legal Risks

As the data collected has been created by third parties, it is important to check the rights of use. In this case, the Wiki uses the GNU Free Documentation License that states the following [25]:

"Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed."

"Collections of documents - You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various

documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.”

The Wiki also states that the use of bots is allowed provided these bots are not making edits. However, there is a risk of overwhelming the Wiki’s server by flooding it with internet traffic. Therefore, this project could include a pause between each page being read to mitigate this happening.

This project does not therefore breach any terms of the license or the Wiki, as it only requires collection of data and no edits to the Wiki.

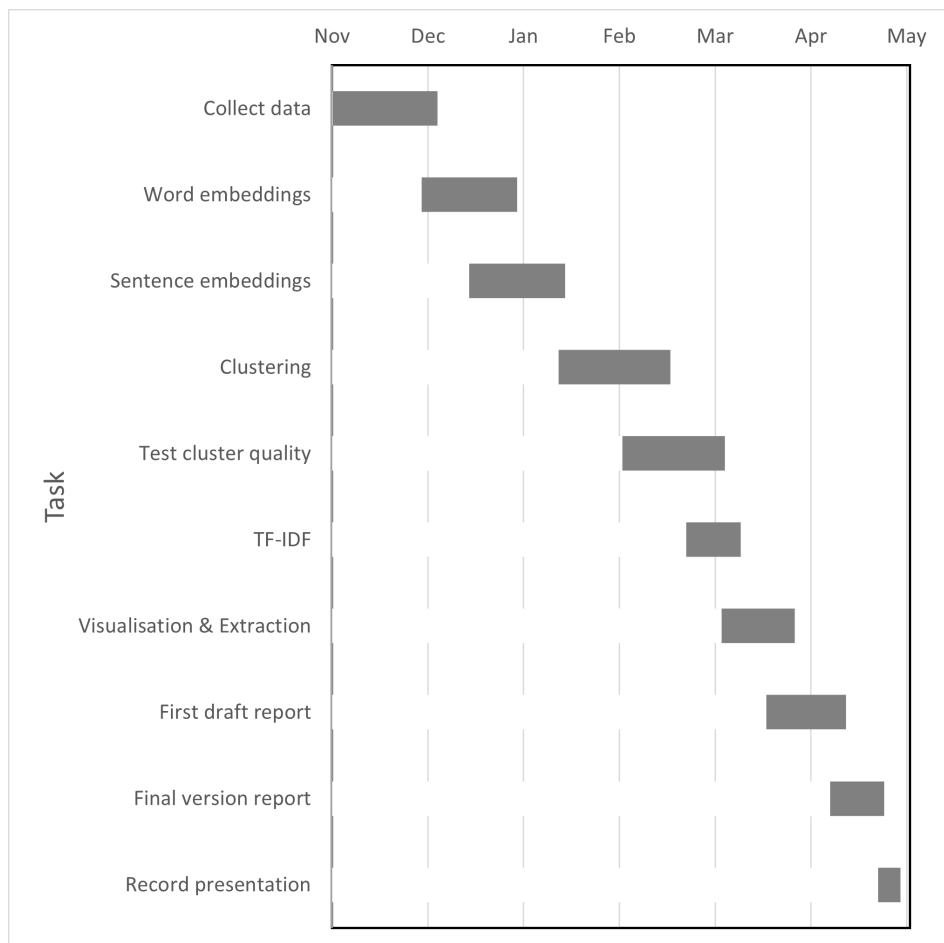
## 4.2. Project Risks

By design, this project is a highly computationally expensive task, so efforts to use efficient algorithms must be taken. More specifically, efficient approaches must be taken during the embeddings and clustering, whilst maintaining accuracy in the results. This has been shown in similar projects in which BERT took 65 hours to analyse 10,000 sentences, whilst SBERT only took 5 seconds to achieve the same level of precision [26]. In addition, this project will benefit from the availability of a powerful PC (from project supervisor) that can be used for parts of the project that are computationally expensive.

Furthermore, since the Wiki consists of pages that are created and edited by any user, there is no guarantee that the information on each page is accurate. The Wiki tries to counteract this by requiring users to cite their information sources. However, without any checking of these sources, incorrect information can still appear that could affect the end results of the project.

## 4.3. Schedule of Deliverables

The chart below shows the deliverables and their approximate due dates:



**Fig. 7.** Project Schedule

- 
- [1] D. N. Sotiropoulos, G. A. Tsihrintzis, M. Virvou, and E.-A. Tsichrintzi, "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.
  - [2] S. J. Heine and M. B. Ruby, "Cultural Psychology," *WIREs Cognitive Science*, vol. 1, no. 2, pp. 254–266, 2010.
  - [3] J. Abello, P. Broadwell, and T. R. Tangherlini, "Computational Folkloristics," *Commun. ACM*, vol. 55, no. 7, p. 60–70, July 2012.
  - [4] T. Meder, F. Karsdorp, D. Nguyen, M. Theune, D. Trieschnigg, and I. E. C. Muiser, "Automatic Enrichment and Classification of Folktales in the Dutch Folktale Database," *The Journal of American Folklore*, vol. 129, no. 511, pp. 78–96, 2016.
  - [5] L. Manovich, *Cultural Analytics*. The MIT Press, 2020.
  - [6] Z. Zhang and J. Yang, "Data mining of myths, legends and folk tales in the context of artificial intelligence," in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, 2020, pp. 841–844.
  - [7] J. Werzinsky, Z. Zhong, and X. Zou, "Analyzing folktales of different regions using topic modeling and clustering," *arXiv preprint arXiv:2206.04221*, 2022.
  - [8] C. Zheng, G. He, and Z. Peng, "A study of web information extraction technology based on Beautiful Soup," *J. Comput.*, vol. 10, no. 6, pp. 381–387, 2015.
  - [9] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 302–308.
  - [10] F. Almeida and G. Xexéo, "Word embeddings: A survey," *arXiv preprint arXiv:1901.09069*, 2019.
  - [11] D. Liang, J. Alotaibi, L. Charlin, and D. M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 59–66.
  - [12] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 238–247.
  - [13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
  - [14] O. Barkan, N. Razin, I. Malkiel, O. Katz, A. Caciularu, and N. Koenigstein, "Scalable attentive sentence pair modeling via distilled sentence embedding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3235–3242.
  - [15] X. Zhu, T. Li, and G. de Melo, "Exploring semantic properties of sentence embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 632–637.
  - [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
  - [17] M. Steinbach, V. Kumar, and P. Tan, "Cluster analysis: basic concepts and algorithms," *Introduction to data mining, 1st edn. Pearson Addison Wesley*, 2005.
  - [18] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.
  - [19] L. Wang, P. Chen, L. Chen, and J. Mou, "Ship ais trajectory clustering: An hdbscan-based approach," *Journal of Marine Science and Engineering*, vol. 9, no. 6, p. 566, 2021.
  - [20] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, no. 1. New Jersey, USA, 2003, pp. 29–48.
  - [21] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based tf-idf procedure," *arXiv preprint arXiv:2203.05794*, 2022.

- [22] E. Gündoğan and M. Kaya, "Deep learning based conference program organization system from determining articles in session to scheduling," *Information Processing & Management*, vol. 59, no. 6, p. 103107, 2022.
- [23] A. Banerjee and R. Dave, "Validating clusters using the hopkins statistic," in *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, vol. 1, 2004, pp. 149–153 vol.1.
- [24] D.-T. Dinh, T. Fujinami, and V.-N. Huynh, "Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient," in *International Symposium on Knowledge and Systems Sciences*. Springer, 2019, pp. 1–17.
- [25] F. S. Foundation, "GNU Free Documentation License," 2008. [Online]. Available: <https://www.gnu.org/licenses/fdl-1.3.html>
- [26] L. Ajallouda, K. Najmani, A. 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.

