SCIENCE MAP EMBEDDING BY DIMENSIONALITY REDUCTION OF SCIENCE JOURNALS

M.Sc. DATA SCIENCE DESSERTATION PROJECT

Arjunkumar Wandiwash Krishnakumar

Student id-2457663

Supervisor Name- Jens Christian Claussen

School of Computer Science

2022-2023

**ABSTRACT**

The aim of this project was to create a science map embedding that would showcase various papers from a variety of journals as individual points on a Euclidean space. After creating this Euclidean space with points of various papers as points we then give provision for user provided paper which would also be mapped on the space to give visualisation on the relative distance between the different papers. The reason for doing this would be to analyse whether the papers from each journals would group together on the map as in whether they all have similar targeted topics or subjects which would lead them to have clusters forming on the space. Secondly, to allow user to visualise where their paper falls in this space to make a decision on the journal where they should publish their paper. A number of natural language processing techniques are used and the best technique is picked out of them for converting the papers into numbers indicating the likelihood of the different papers falling under different subjects. These numerical are then processed using principal component analysis for dimensionality reduction where the reduced dimensions are used to plot the science map. The data is taken from open source science journal websites which allow provision for bulk downloads. A web crawler was created which navigates to this website and downloads the required science journals automatically and stores them in a designated location. The downloaded files are then processed where the necessary text from them such as the abstract and introduction are extracted and the special characters, text spacers and other symbols are removed in preparation for the text analysis. The cleaned text is subjected to a number of natural processing techniques where the best one is picked and used on all the papers from different journals.

Table of Contents

[1. Background research 7](#_Toc145887670)

[2. Dataset 7](#_Toc145887671)

[3. Data Collection 8](#_Toc145887672)

[4. Exploratory Data Analysis 9](#_Toc145887673)

[5. Preprocessing 10](#_Toc145887674)

[6. Modelling 12](#_Toc145887675)

[Model 1 14](#_Toc145887676)

[Model 2 17](#_Toc145887677)

[Model 3 20](#_Toc145887678)

[Model 4 21](#_Toc145887679)

[7. Results 30](#_Toc145887680)

[8. Conclusions 32](#_Toc145887681)

[9. Appendix 32](#_Toc145887682)

[10. Acknowledgements 33](#_Toc145887683)

[11. References 33](#_Toc145887684)

TABLE OF FIGURES

[Figure 1 Data Collection stage- selenium 9](#_Toc145875436)

[Figure 2 Exploratory Data Analysis- Science journal name 10](#_Toc145875437)

[Figure 3 pypdf2 to read the papers from science journal 11](#_Toc145875438)

[Figure 4 The read text from science journal-scientific african 11](#_Toc145875439)

[Figure 5 Text before Preprocessing 11](#_Toc145875440)

[Figure 6 Text after processing for modelling 12](#_Toc145875441)

[Figure 7 Text lemmatization for modelling 12](#_Toc145875442)

[Figure 8 Model building 14](#_Toc145875443)

[Figure 9 Model 1 15](#_Toc145875444)

[Figure 10 Zero shot classification 15](#_Toc145875445)

[Figure 11 results from model 1 16](#_Toc145875446)

[Figure 12 Results from Model 1 with confidence value 17](#_Toc145875447)

[Figure 13 Model 2 19](#_Toc145875448)

[Figure 14 Model 2 results 20](#_Toc145875449)

[Figure 15 Model 3 20](#_Toc145875450)

[Figure 16 Model 3 results 21](#_Toc145875451)

[Figure 17 Model 4 22](#_Toc145875452)

[Figure 18 Bulk run results score list 24](#_Toc145875453)

[Figure 19 Principal component analysis results 25](#_Toc145875454)

[Figure 20 Pre declared subjects 25](#_Toc145875455)

[Figure 21 Subject scores for 1 paper 26](#_Toc145875456)

[Figure 22 Subject scores compared among three papers 26](#_Toc145875457)

[Figure 23 Scatter plot showcasing science map 27](#_Toc145875458)

[Figure 24 Science map with k-means clustering 28](#_Toc145875459)

[Figure 25 Scatter plot of second journal 29](#_Toc145875460)

[Figure 26 K-means of second journal 29](#_Toc145875461)

[Figure 27 Science map results 30](#_Toc145875462)

[Figure 28 Final science map results combined with user point 31](#_Toc145875463)

**INTRODUCTION**

This project idea was a combination of the motives from Dr. Jens Christian Claussen and my intent to work on a project involving natural language processing. He gave the motivation to pursue the project on science maps and their utility in analysing the landscape for papers being published and their perceived success before they are even published. This sounded very interesting to me especially because of the freedom in this project that allowed me to use any data science techniques to achieve my goals. The ability to implement natural language processing into my final dissertation was important to me because of my interest in the subject. The experiment pipeline for this project is as follows. First, create a web scrapper/ web crawler where we navigate to the website from where we can download the open source science journals and use them for our project. The selection of the website with open access for science journals is important as the data source such that we can keep expanding our dataset of papers as needed and as required down the line. Once the dataset is acquired and downloaded to our local machine, we will move to the next step of the process of preparing the data for further processing and exploratory data analysis. The data is analysed for any extenuating issues and outliers and if any are found they are processed. Once the data is explored and pre-processed , we move on to the next step which is to apply data science models and techniques on them to predict their target topics and also to get a numerical confidence in those said topics for visualisation.

While model building we tried a variety of models before picking the best one which had the highest results reproducibility and time efficiency. There were a total of four models which were built each with their own advantages and disadvantages. The first model utilised the transformers library and the hugging face interface for the purpose. Transformers is a powerful neural network architecture introduced by google and Hugging face is an Artificial intelligence and machine learning platform within the transformers architecture with the aim allowing programmers to contribute towards models and have access to pre-trained models and architectures. These models could be applied to a variety of data types. Hugging face transformers also provide a variety of datasets and models which were trained in those said datasets such that they can be readily used for modelling. The first model is built different using different variations of this technique and the output is evaluated.

The second model utilises the Latent Dirichlet Allocation in a way to implement topic model and it works by generative probabilistic modelling where each provided text is assumed to be made up of different topics consisting of different proportions. The approach was decided upon after careful research on the possible methods that could be used to derive the topics of a provided text without actually giving any pre-determined topics by doing a bag of words representation approach. In model two we pick out the words which are repeated most number of times and use them in the LDA algorithm to use them in a sequence in an attempt to predict the topics. This method actually provides 5 topics which are likely using the words present in the text itself. But this ends up not being a very accurate portrayal of the topic subject and thus becomes the limitation for the model.

Model 3 is a direct inspiration from the Model 1 where we use the transformers neural networks architecture. The aim of model 3 was to be an improvement to model 1 by using a smaller linguistic pre-trained model instead of the larger but more bias less approach in the model 1. This approach showed promise at the theoretical stages and was a perceived improvement. But in actual implementation using a smaller trained model offered no noticeable benefits and the results were arguably more inaccurate. This setback on the approach gave way for the next model.

Model 4 was an incremental improvement aimed at the downside of model 2. In model 4 we tried to use the bag of words representation and also the LDA algorithm but the approach was targeted at the results produced by model 2. The downside of model 2 was that it gave results were the predicted topics was but the words from the provided text itself which wasn’t the widely regarded subject names most times. Thus in model 4, we tried to run the LDA algorithm and predict the possible topic from the given results from model 2. This incremental approach showed initial promise and was tested. But the results were similar to model 2 as to predict an well known topic name, we would have to pre-declare the topic names thus removing the upsides of this model completely and at this point it would have been just a lengthier implementation of the model 2 without better results. Thus after 4 models we picked model 1 to be the best model for the implementation of the project for topic prediction and thus went ahead with the bulk run of it on all the pdf files.

Once the bulk run was completed on all the papers of an edition of the Journal Scientific African, we are presented with the results in the form of a dictionary. The key of which are sequence, topics and scores where sequence is the text on which the model was ran, Topics are the subjects they can fall under which were initialised and scores are the scores for each of those topics to be about the sequence given. The scores all summate to 1 and the score written first is the highest values one thus being the most likely to be correct accordingly. After we get our results we store them locally and read them into a list of lists where each list is the scores of individual paper appended into a superset of lists.

We take the list of score list and process them for visualisation. Since the scores are composed of multiple labels and there are multiple scores, it is of higher dimension and thus required dimensionality reduction before enabling us to plot it. Principal component analysis is the basis of multivariate dimensional analysis, PCA is a statistical procedure usually applied on large dataset which produced multi-dimensional results each run. This techniques allows for higher dimension data to be portrayed in lower dimensions while preserving as much information in them as possible. Once we have the score list which is of lower dimensions which in our case is given by two data points per paper entry instead of original 8 corresponding to number of labels declared, We move on to the next step which is plotting and visualizing it.

We visual the data in the form of bar plots and scatterplot where the scatterplot acts as the science map and the bar plot is present to showcase the distribution of scores for the individual subjects. The plots provide us with clearer image of the results and shows different regions in the map where there are points being grouped together and also regions where the points are scattered. Thus, this required us to utilise the help of clustering techniques.

K-means clustering techniques was utilised to cluster the data points. K means clustering is a method od vector quantisation where the original n observations are clustered into k clusters. K means clustering also reduces the distances between clusters that is it reduces the squared Euclidean distances. This clustering techniques is then applied to our data of reduced dimensions and we visualise the results again. This results in the creation of the science map with different clusters showcasing papers with reduced squared Euclidean distance thus portraying their similarity in terms of the topics they are discussing about.

After the completion of the science map for the first journal we repeat the process for another journal to increase the data points on the map and afterwards we move on to the final step in the project which is to allow for the user to upload their own paper and visual its position on the science map. We do this with the help of python library tkinter and use it to prompt a window at the user to upload or show the path of their paper of choice. Once the paper is uploaded, it undergoes the entire pipeline where it is pre-processed, lemmatized and prepared for the model application. The model is ran on the text and the results of higher dimensions is applied with PCA to prepare for visualisation on the science map. Finally, the user point is plotted along with existing science map to showcase the final completed map and we can see exactly how the paper lies on the grand space among different papers between popular journal. This gives us the idea on how the paper might perform depending on its proximity to the papers from popular journals.

# Background research

The background research on this topic is very inspiring, the main concept from a paper co-published by my project supervisor and mentor professor Claussen,J.C in his Model for dynamical evolution of science in space. Where it is summarized as The precise position of new papers depends on previous topics of the respective authors and is chosen randomly in a surrounding neighborhood including novelty and interdisciplinarity. Depending on parameters, the spatial structure resembles a simple Gaussian distribution, or spatial clusters of side-topics are observed[1]. This project is a variation of the above idea with the usage of modern data science, natural language processing and machine learning techniques to create a 2 dimensional Euclidean science map. According to a traded metaphor “a university is just a group of buildings gathered around a library” [1,2], and ∗ Corresponding author.; j.claussen@jacobs-university.de this is akin of a seed of any model of science: Scientists provide knowledge to the library, interact through it, and grow into the surrounding.

Quoting from another fundamental paper behind this project , “About 40 years ago, Derek J. deSolla Price2 suggested studying science using the scientific methods of science. Since then, research in bibliometrics and scientometrics has developed techniques to analyze publication data sets”[3]. Most of the early work focused on identifying networks or clusters of authors, papers, or references[4,5,6].

Overlay journal model was discussed in this following map giving an interesting perspective. Quoting from the paper, “In the most common form, authors first upload their preprint to an OA repository and the link to the manuscript is sent to the journal editors. After peer-review and formal acceptance, which is managed by the journal editors, the authors upload the final article version to the OA repository. The overlay journal links to this and provides journal issue and volume information (hence ‘overlay’)”[7]

For the final main paper that inspired this topic and question forming, we look at the paper [8]. Quoting from here “It is apparent science journalism has yet to receive a clear, highly supported theoretical articulation in the literature that links theory to practice.”

# Dataset

For the problem, I needed the science papers from a variety of journals to remove any bias. If the science papers are just from one publisher or just from one region, then it may be heavily tilted towards one specific market as its target. Hence the Dataset had to be diverse from a variety of journals to meet the requirements. Furthermore, they also should allow for the individual papers from each edition of the journal to be downloaded via automation because doing this process manually would be a huge time dump and would remove any future scaling to be out of the conversation. Hence after extensive research I chose the website Science Direct which allows provision for reading various open source science journals and also allows to download them. Since there was no need for any sort of payment and the download process was straightforward, I created a web crawler using selenium which would navigate to the website and to the required journals which were selected ahead of time and start the download process. As such, for the first dataset I took an edition of the scientific African journal for a few reasons. Firstly, The Scientific African is a well established journal which tackles a variety of topics and has many papers in each edition of its publishing. Secondly, Because the papers are published in a particular journal, they largely follow a particular format which removes the effort which would be necessary when trying to remove their abstract or introductions. Because they are published in the same journal, we can extract the exact part we are interested in which is the introduction and abstract from the paper which gives the summary of the paper’s area of subjects. Since I created the web crawler with future upscaling in mind, the code is setup in ways to allow for it to be extended to download as many journals and papers as required in full automation. Having a large dataset is essential to remove bias and also to have a large science map where clustering techniques are applied to it to form clusters which would indicate that different journals have their published papers clustered together depending on their targeted topics.

# Data Collection

Data collection is the process of collecting and analysing data and information from a variety of sources with the goal of finding solutions to research problems and projects. When collecting data we need to take a variety of conditions into considerations to keep the research and project integrity. The Data has to be ethically sourced, anonymous or made anonymous by removing all the personal information if present that could identity any real person and should also be of high enough quality and standard such that the research findings could be proven true when replicated elsewhere. With such high policies taken into consideration, the data was decided to be open source science journals which would pose no issues while used for analysis. Data collection isn’t a new subject, it has been done for all of the research history, with such astronomically high amount of data available to use today due to internet access, it is of more importance than ever to carefully select the source of data to maintain the research standards and project’s smooth completion. Hence I selected ScienceDirect which provides open access science journals and also allows for bulk downloads of their monthly publications.

With the Data selection decided next comes the question of the methodology of the data collection to be used. The easiest method would be to manually select the journals to be downloaded and to download the papers we want individually and use them where necessary. This method would work if we are only using a couple of papers for our project or at most just a few more. But the scope of this entire project relies on using the text from multiple papers from a variety of journals and as such this method of manual data collection is extremely time limiting and not rewarding for the time invested. Hence the method of data collection had to be automated and allow for upscaling to download hundreds if not more science papers as needed.

Automated data collection can be achieved by creating a web crawler to fetch the data. A web crawler is a spider, search engine bot which navigates towards the target website, web location or forum of choice to search indexes and do downloads of the pre coded files and documents. The correct technical term for the type of bot created for our purpose here in this project is called a web scraper which is subtype of the web crawler because the bot is doing very targeted file access and downloads instead of searching through a catalogue of files and finding the required file. Our bot accesses the ScienceDirect website and navigates towards very particular versions and editions of the science journals which were pre selected for our purpose. There are multiple upsides for using this approach such as faster file acquisition because we are not investing time for searching for the file and secondly, we put much less stress on the servers.

Web scraping can be achieved through a variety of methods and one of the most popular techniques is using the selenium. Selenium is a python library which is useful for automation in a variety of tasks which are repeated a large number of times. One such purpose is to create web crawlers to access the necessary information and perform required tasks in selected websites. We use this library by first installing the package selenium into our python integrated development environment of choice which is jupyter notebook in our case. After installing the package, we import it and the necessary components from it for our purpose. The required options are declared and chrome driver is used to emulate the actions of a person launching a search engine and downloading the files. Chrome driver is a web driver which is an open source tool used for automated testing and tasks for web applications across many browsers. There are several download options available for chrome drivers and we choose the version compatible with our local machine’s version on the browser. Afterwards, we use the python get method to access the website of selection. Once navigated, we find the information we are interested in using a method to locate them which in our case is find element by its CSS selector. The science direct website has provision to select the individual papers we are interested in one by one or to select all the papers belonging to that version of the journal in one button. Once selected we navigate to the option to download all the selection and point out the location where we want the downloads to be saved. After completing these steps , we wait for the process to be completed at the end of which we will have a zip file with all the papers of our interest in the location of our preference.

A computer screen shot of text

Description automatically generated

Figure Data Collection stage- selenium

# Exploratory Data Analysis

Exploratory Data Analysis is used by Data Scientists to analyse and investigate the data before performing their data science models on them with the goal of finding the characteristics of the data. By doing EDA we hope to summarise the main points of interest of the data, discover patterns, anomalies, check for hypothesis etc. EDA is primarily for getting a better understanding of the dataset and their variables and the relations between them. It helps to look at the data before performing the processes on them and to maintain the research coherence. It is necessary because it affirms us if we are indeed asking the correct questions and would our data assist in answering the said questions. Once EDA is completed and we have more sophisticated data in our hand, it gives a much more clearer image on the tasks that need to be performed on the processed data to get the desired project output or the solutions to questions of the research.

In our case, since the data is papers published on science journals, we are mainly interested in the text part of the introduction and abstract which gives the gist of the research. Thus, through EDA we want to find any papers with content in these sections that could be extremely challenging to process such as images or symbols of high significance towards understanding their goals. We aim to find these papers and analyse if they can be salvaged and still be useful towards our goal with just the remainder of their texts. We do this by loading these papers into dataframes and looking at them to investigate our questions of importance. By doing so we can check all the primary concerns as well as doing other rudimentary checks including null checks and missing data checks.

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

Figure Exploratory Data Analysis- Science journal name

After EDA we have more clearer image of the data quality and coherence which enable us to proceed towards the next step in our process which is to preprocess the data. EDA informs us of the need for preprocessing as there are symbols and characters in the data that might lead to difficulty in running the different natural language processing models on them.

# Preprocessing

Since the data was collected in the form of pdfs of all the science papers issued in a month or a particular issue of the journal, they all follow somewhat similar formats making the need for high preprocessing unnecessary. Still there was need for removal of the special characters, symbols and other lexical resources that are not pivotal for the goal of the project. We use the method pypdf2 to read all the pdfs from the local dataset folder into our project. The pypdf2 is a great library which is quite useful when operating with pdfs as it has many resources built in it to help with the loading and processing of these files. Once we read all the content of the pdfs into our code, we then scan the text for particular key words such that we only take the content of interest to us for further usage. In the first run of the preprocessing stage, we take only a single pdf to perform all our processing to ensure the robustness and proof of concept for the method. This is also time saving to perform on a single paper to test the different processing methods and check the results. The paper is loaded into an object using the open method and we use the object to read the content using the pdfFilereader method. We find the number of pages present in the paper and extract the text by looping through all the pages using the python function called extractTextmethod. A computer screen shot of a program code

Description automatically generated

Figure pypdf2 to read the papers from science journal

The extracted text is displayed and from it we decide the keywords to target the text content of our interest. The text is processed using the encode, decode method and character conversion into lowercase.

A close-up of a text

Description automatically generated

Figure The read text from science journal-scientific african

A screenshot of a computer

Description automatically generated

Figure Text before Preprocessing

As we can see from the above image, the text before processing has many characters and symbols in addition to the text of interest. After the processing the text gets converted into a usable format without any clutter thus being ready for further processes. This text cleanup process is performed via the help of a number of processes which are explained further here. Lemmatizer is used to bring together different forms of a word such that they can be analysed together as a single item. Lemmatization brings out the context to the words such that it links the words with similar meanings to one word. This process is carried out with the library Natural Language toolkit(NLTK) which is a python toolkit used to work with human language data. Using this library we make changes to words such as changing their case to lowercase, splitting them where needed and also lemmatizing them appropriately. The result of it is showcased in the diagram.

A close-up of a screen

Description automatically generated

Figure Text after processing for modelling

The code used to implement this process is showcased here in this image where we import the nltk library, declare the relevant variables and do the process of preprocessing as needed where necessary.

A computer screen shot of a computer code

Description automatically generated

Figure Text lemmatization for modelling

# Modelling

Modelling in data science is creating a process that works on the data with the objective of assessing the rules guiding the data and finding the different relations between them. A data model un-complicates the data into useful information which can be used in meaningful decision making at higher level in the organisation or to guide the opinion of the data scientist in the solution of their research problems. With the huge influx of data in today’s society and more being created with each passing moment, it is imperative that the data model takes into account all the possible issues that could be present in data and suggests solutions that could be performed in downstream as actionable solutions. In simpler terms, Data modelling is the software modelling of the data such that it is converted into simpler information portraying information which aids in decision making or provide the solution to the project question. The type of data model we need to build depends on the type of data present and the type of output we are expecting from the model. Our data is text based and particularly it is human language text which contains high level academic lexical resource. Hence, we are looking towards language processing techniques which would be Natural Language Processing.

Natural language processing is the process used by computer programs to interpret the human language as it is written and spoken thus referred to as the natural language. NLP uses artificial intelligence to take natural world language as an input to process it and make meaning out of it in the way computers can use it as Data for processes they are working on. The NLP programs first break down the sentence using different techniques before applying the techniques. This break down can be done via a number of methods such as tokenization where the words are broken into smaller units to work with. Stop word removal where common words are removed such that the unique words from the texts are used to get the most information. Lemmatization where words are reduced to their base forms to process. Once the preprocessing is completed the NLP techniques are applied on them. They are of two main types which are Rules based system and Machine learning based system. Rules based approach uses carefully designed linguistic rules to approach the natural language. Machine learning systems uses statistical methods to perform tasks they learn from the training model on the testing models. Natural language processing uses a mixture of deep learning, machine learning and neural network techniques in combination to achieve their goals.

Broad applications of Natural language processing includes text classification, text extraction, machine translation and natural language generation. Text classification involves classifying texts and putting a label on them which is exactly the application we are looking for in our project. Text extraction is summarizing the texts and extracting the key information from them and saving the time and effort for people to manually read through them and summarize is themselves. Machine translation is a very innovative technology where texts in different languages can be translated into a language of choice based on the algorithm. Natural language generation is used to analyse and generate natural language.

For our project we used a in total four models and compared their results to choose the best one out of them based on factors such as reliability of the output, time taken to run the model and the usability of the numerical generated from the model in visualisation. We chose the model utilizing the hugging face transformers for a number of reasons. All the models are explained in further sections but the brief overview of the hugging face transformers is given here.

Transformers is a simple but powerful neural network architecture introduced by google as part of their google brain in 2017. It works on an attention mechanism instead of the more traditional sequential computation found in recurrent networks. Training the transformers which is a deep learning neural network from the scratch is not a simple task as it requires finding the required data for the problem in a correct form which can be extremely time consuming and also the hardware resources such as GPUs needed to train this can be very costly. This was also an issue we faced while performing the project which is the lack of labelled data on the science journals about their exact topic and the numerical indication on the percentage of likelihood of them falling in that topic. This is where the re-use of the deep learning neural networks come in handy. The re-use of models which were already trained using large amount of labelled data is useful because it allows us to retrain the head of the model such that they can be repurposed into our use case.

Hugging face is an Artificial intelligence and machine learning platform with the aim allowing programmers across the globe to democratically contribute towards models and have access to pre-trained models and transformer architectures. These models could be applied to a variety of data types such as text, images and voices. Hugging face transformers also provide a variety of datasets and models which were trained in those said datasets such that they can be readily used for modelling without the need for the user to build extensive labelled dataset and train them locally. Transformers enable their usage intuitively via the methods called pipeline. The pipelines provide APIs for interfacing a variety of tasks and encapsulate a variety of natural language processing techniques. We are trying to use a classification pipeline for our purpose in this project and picking the correct classification model is pivotal. Classification models are a type of supervised machine learning method where the model tried to predict the correct label for a topic or data on the testing dataset based on its training in the training dataset.

Zero shot classification is the classification technique which we picked in the hugging face transformers model where the classifier attaches a numerical score to the text provided and the score indicates the likelihood of the text falling into the predefined list of labels. This method is very impactful in the sense that it doesn’t require the user to create a large quantity of labelled meaningful related data and train the model on them beforehand to attain any results but rather it uses the hugging face transformer’s features of crowdsourcing the data labelling and model training to its advantage and uses them to categorise the user’s text into different labels and put a numerical confidence to it to indicate the likelihood of the provided topic to fall in them and by how much.

A computer screen shot of a program

Description automatically generated

Figure Model building

We tried and testing a variety of transformers models before deciding the one we want to use for the entire dataset. Since each run of the model takes sometime , it is not viable to bulk run every time while testing.

### Model 1

AutomodelforSequenceClassification is a hugging face transformers library which offers a quicker way to use pre-trained language models on a text classification job and can be used to output multi-labelled classifications. Because of its utility we picked it for the first iteration of the first model for testing. The second import is of the Autotokenizer. Tokenizer are used to prepare the inputs for a model. Any library comes with its own sets of tokenizers and these come in two flavours. A full implementation on python and another type is a fast implementation where we use Tokenizers from a Rust library. The fast implementation allows for a much quicker turn around times.

A screenshot of a computer program

Description automatically generated

Figure Model 1

From the image, we see we used a model called Facebook bart large mnli which is pre-trained Natural language inference model doing zero shot text classification. This method of using pre-trained natural language inference works by posing the text to be classified to the model as an NLI premise and asks it to construct a hypothesis for each of the candidate labels. These models require a series of candidate labels to work to classify the text into them and they do so by creating a sequence for each of them and evaluating those sequence into whether this belongs to the said labels. For example if we had a sequence talking about stock market and their was a candidate label called economy then the model will create a statement using the said sequence about stock market and evaluate it in likelihood of whether the sequence falls under the stock market label. This method works really well when utilised with large pre-trained models such as bart because it removes as much bias as possible which would be a difficult ask when training the language model from start without the investment of extensive language data and GPU requirements.

A white background with black text

Description automatically generated

Figure Zero shot classification

This model of Facebook bart large mnli can be used for our scenario by a zero shot classification pipeline or by the use of sequence classification via manual pytorch. Both the techniques were used and checked for time efficiency. In technique 1 we use the tokenizer to prepare the inputs before being pushed for prediction on the candidate labels. The results of the prediction is a list of each of the labels and corresponding confidence on the certainty of the said labels being falling into them with a numerical confidence attached to it. The results are outputted in the format of a dictionary where the first key is called ‘sequence’ and the value for it is the text used to predict the classification and the second key is the ‘labels’ with the list of labels we declared beforehand. The final key of the dictionary is the ‘scores’ where the confidence number for each of the respective label is given in order of the highest to lowest confidence number. All the confidence numbers summates to one.

A screenshot of a computer screen

Description automatically generated

Figure results from model 1

This figure showcases the final model one which gave the most promising results out of all the models which would be showcased in the following sections. The final predicted topic is the ‘Economy’ out of the given subjects of ‘Arts’,’Microbiology’,’Chemistry’ and ‘Economy’. The model was ran on smaller text for quick turnaround and time efficiency.

Advantages of Model 1-

The upsides of model 1 is in the reliability of its results. The results are derived from large pre-trained linguistic model and they are accurate and reproducible given the same circumstances. Since the results are also in numerical with the confidence digits, its comes useful in the further steps of the project for the principal component analysis and visualisation.

A screenshot of a computer

Description automatically generated

Figure Results from Model 1 with confidence value

#### Disadvantages of Model 1-

Some downsides to model 1 lies in its necessity to pre-declare the topics and the results to be in those topics which could be restrictive but this can be handles quite easily by simply having a large selection of topics and the topics to be carefully selected on the scientific domain. Secondly, it takes an increased amount of time comparatively to the other models to run as it is running quite the large model on the limited resource from a laptop GPU. This could also be resolved quite easily by having the program run on enterprise systems and GPUs which would be the case in situations of large scale adaptations.

### Model 2

For Model 2 the approach attempted was a bit different. In model 2, instead of declaring the labels the topic could possibly fall under, we decided to predict the topic from the provided text itself by using the most repeated words from it. The idea was to look at the text and start counting the words in them and arrange them in descending order of the words based on the number of times they were repeated on the provided text and then applying the Latent Dirichlet Allocation (LDA) on them. Latent Dirichlet Allocation is a way to implement topic model and it works by generative probabilistic modelling where each provided test is assumed to be made up of different topics consisting of different proportions. The approach was decided upon after careful research on the possible methods that could be used to derive the topics of a provided text without actually giving any pre-determined topics at all. The idea was that by not having any pre-determined topic, we could predict any topic and not have the restriction on the possible outcomes. Another possible goal for this was that because we aimed to derive the results from the text itself we could potentially have the model run much faster because we are not using any largescale linguistic pre-trained models.

LDA Algorithm-

* This algorithm works by for each document, it randomly allocates each word to a different pre assigned topic which in our case is also generated automatically by counting the number of times it is repeated and picking the highest counts as a pre-assigned topic.
* For each word w in the document, compute:
  + P(topic t| document d): This is the proportion of the words in the given document d to the pre assigned topic t
  + P(word w| topic t): This is the proportion of the assigned topics t to the words that come from the w
* Reassign topic T’ to word w with probability p(t’|d)\*p(w|t’) considering all other words and their topic assignments
* This final step is repeated multiple times till we reach the state where the topic assignment stops changing with each run and we reach an equilibrium. The proportion of topics for each document is then determined from these topic assignments.

From the image we see that first we tokenize the text and preprocess them. This process takes care of the words and different punctuations and makes the text ready for further steps. From the process token we create a dictionary out of it such that we can create a bag of words representation for it. This step is needed because we need the bag of words representation for the next step of LDA algorithm to function. Once the LDA algorithm is completed we get topic assignments to the text based on its results. We then display these topic assignments to analyse them on choosing the most likely contender.

A screenshot of a computer program

Description automatically generated

Figure Model 2

The results is a list of lists where each list is a topic where each topic has a series of words and the topic is predicted to be the combination of the said words together. So in the output we have five topics and these five topics are made up of combination of the words in their respective lists and the text is most likely to fall under the topic with the highest probability from the LDA algorithm and the topic is the combination of those words. In the result image we get the topic 3 to be predicted to be the most probable.

#### Downsides of Model 2-

The downside of this model lies in the strength of the model itself. The advantage of the model initially was that we could predict the topic of the text without defining any pre-determined labels and such that we could run this in any piece of text and predict the topic. But in reality we see that is hardly the case because that would only work if the text consistently has its topic of content being repeated in its sentences many times and then we get those words and predict them as part of our LDA representation. But that isn’t what we observe and thus this model’s output ends up being a combination of words which are repeated multiple times and all the 5 topics predicted from the model is a repetition of each other and we don’t derive any meaning out of it. The topics predicted when combined as the series of words representing them don’t depict any meaning and cannot satisfy our results criteria. Thus this model isn’t superior to our existing model.

#### Upsides of Model 2-

The advantages of this model lies in its incredibly fast runtimes. The model is almost instantaneous in its runtimes and the most likely reason for this behaviour is because we are not importing any large pre-trained language models and are simply performing operations in counting the words repetition and predicting their usage using the LDA algorithm. Secondly, We are not creating any pre-determined labels and this would in theory could have allowed us the possibility of predicted any topics for any science journals. But after extensive testing we find that is not the case.

A screenshot of a computer code

Description automatically generated

Figure Model 2 results

### Model 3

Model 3 was aimed to be a direct improvement from the Model 1 where we use the transformers neural networks architecture. The aim of model 3 was to use a smaller linguistic pre-trained model instead of the larger but more bias less approach in the model 1. In model 3, we used the Bidirectional encoder Representations from transformers(BERT) model which is from a family of language models introduced by google[9]. This model was selected over the original because of its smaller size and thus requiring lower power to run. The goal was to have a model smaller than model 1 for much quicker run times. The bert model is pre-trained model which was self trained, meaning no human intervention was present in its labelling in any way using the masked language modelling. Masked language modelling means that the model takes a sentence, masks some part of it like 10-15%, then runs the statement through the model for it to predict the completed statement. This process allows the model to learn in a bidirectional representation of the statement. To utilize this model properly we use the bert tokenizer and bert sequence classification.

A computer screen shot of a computer code

Description automatically generated

Figure Model 3

A close-up of a text

Description automatically generated

Figure Model 3 results

This approach showed promise at the theoretical stages and was a perceived improvement. But in actual implementation using a smaller trained model offered no noticeable benefits and the results were arguably more inaccurate. The results were more biased and in some cases were even wrong. The time savings that was expected wasn’t seen either. Thus this model proved not superior to the Model 1 in anyway. This setback on the approach gave way for the next model.

#### Advantages of Model 3-

The initial premise of the model 3 by using the bidirectional language model BERT was that because the bert model was smaller and was self-trained, it might prove to be faster and be just as accurate for our needs. For our purposes the results of model 1 are satisfactory but looking for improvements is always better and time efficiency is always worth exploring especially for large datasets such as ours.

#### Disadvantages of Model 3-

The theorised positives of model 3 were not realised. The time efficiency was marginal and would not summate to any noticeable difference even when ran for all the papers of the different journals. The time savings were predicted to be just few seconds per paper and few minutes after the bulk run on the entire model. But the trade off for this slight time savings was results which were highly inconsistent and inaccurate at many times. For example one paper which was about predominantly on economics and microbiology, we saw the results predicted to be physics which couldn’t be argued as slight inaccuracies.

### Model 4

The model 4 was theorized from the shortcomings of model 2. In model 2 we aimed to categorize the text without actually providing any labels or topics beforehand for the text. We tried to get the topic names from the actual text itself by doing a bag of words representations and counting the number of times words are repeated and thus picking the most repeated words. This did not perform up to our standards because the actual topic name wasn’t repeated multiple times in the text that was being analysed.

Model 4 had a clear goal, that is to take the results we get from the model 2 and try to categorize that said results in to topics that are widely recognised. Meaning, The bag of words representations from the model 2’s results would be fed into a zero shot classification linguistic model and we aim to categorize that results into topics we recognize and thus aim to save runtimes. The theoretical reasoning behind this was that because the bag of words representations are just five words at most, we would be getting very fast turnarounds in the classification neural network model run times and thus this model should have been superior. In actual practice , the model did perform its intended tasks but the combined run times for running two models combined in a sample text wasn’t any superior to the run times of direct zero shot classifications.

#### Advantages of Model 4-

The aimed advantages for model 4 were to overcome the disadvantages of model 2. We would aim to do a classification on the text without giving user labels by directly perceiving the labels from the text being analysed directly. This would have in theory allowed for any labels to be extracted from the text input and would not have had constraint on the results being produced and their variability. The second advantage aimed at was such that because we take the results from model 2 which is just but a few words, the run time while running our model on the results we would get much quicker turn around times.

#### Disadvantages of Model 4-

Because we are running two models instead of a single model in model 4, the aimed time efficiency was not realised. The two models being ran together in model 4 were the model 2 being run on the sample text and the results of model 2 being fed to a zero shot classification neural network model similar to model 1. Because the classifier model was only ran at 5 words results from the output, the results were expected to be produced faster which it did. But the combined time the entire model took from start to finish was very similar to the times the results were produced from running the model 1 on the text directly and thus its purpose was diminished. Thus after four iterations, model 1 was selected for the final runs.

A screenshot of a computer code

Description automatically generated

Figure Model 4

After the completion of the modelling stage, we move on to the data processing stage before visualising it for our final science map. The data output from modelling for all the papers on the journals is of higher dimension and thus we needed to do dimensionality reduction before visualization. The technique selected for dimensionality reduction was principal component analysis that was taught during course.

Principal component analysis-

Principal component analysis is applied on large dataset with multiple dimensions and we use it to project the data on reduced dimensions while preserving most of the information. The large set of variables are reduced to smaller set while preserving as much as possible from the larger dataset.

It is a fundamental tool in modern day data analysis – on a broad spectrum of fields ranging from neuroscience to computer graphics - because it is a straightforward , non-parametric method for extracting important information from large data sets with multiple data dimensions. With minimal effort PCA provides a roadmap for how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified structures that often underlie it.[10]

PCA is defined as an orthogonal linear transformation techniques that transforms the data to a new coordinate system such that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on[11,12]

First component- In order to maximize variance, the first weight vector **w**(1) thus has to satisfy



Equivalently, writing this in matrix form gives



Since **w**(1) has been defined to be a unit vector, it equivalently also satisfies

We identify the quantity to be maximized as a Rayleigh quotient. The biggest eigenvalue of the matrix, which happens when w is the matching eigenvector, corresponds to the maximum value of the quotient in a positive semidefinite matrix like XTX.

The first principal component of a data vector x(i) can therefore be expressed as a score t1(i) = x(i) w(1) in the transformed co-ordinates, or as the corresponding vector in the original variables, x(i) w(1) w(1), after w(1) has been discovered.



The full principal components decomposition of **X** can therefore be given as



where W is a weights matrix with p by p columns that correspond to XTX's eigenvectors. The whitening or sphering transformation are other names for the transposition of W. In PCA or factor analysis, loadings are defined as columns of W multiplied by the square root of the associated eigenvalues, or eigenvectors scaled up by the variances.[13,14]

We use this technique by importing the PCA module from the sklearn library.

A screenshot of a computer code

Description automatically generated

Figure Bulk run results score list

A screenshot of a computer

Description automatically generated

Figure Principal component analysis results

A screenshot of a computer

Description automatically generated

Figure Pre declared subjects

By doing Principal component analysis on the results from bulk run we get 2 variables per paper instead of the original 8 . This is optimal for visualisation as we have preserved much of the information while reducing the complexity of the results for easier representation. We plot the results from the modelling stage for a closer inspection in this stage as well for a look at the conclusions. In the figure we see that microbiology is winning by a close margin here compared to economy.

A graph with blue squares

Description automatically generated

Figure Subject scores for 1 paper

When doing these representations for 3 papers we see that some papers have high variances in their predicted subjects and some don’t have such high variances. Doing dimensionality reductions is important because it will showcase these results in lower dimensions.

A screenshot of a graph

Description automatically generated

Figure Subject scores compared among three papers

After all these steps we finally come to our science map results which is showcased in the form of a scatterplot. We see the points are scattered across the Euclidean spectrum and the results are to be interpreted from this plot which is the next stage and the final stage of the project.

A screen shot of a graph

Description automatically generated

Figure Scatter plot showcasing science map

We require the use of clustering techniques to cluster the scattered points in the Euclidean space because as it is represented currently we cannot make out the outliers to regular points in the science papers and thus the science map isn’t being utilised to its maximum potential. For this problem, we use the k-means clustering technique as a solution.

One of the most straightforward unsupervised learning algorithms to handle the well-known clustering problem is K-Means. The process uses a predetermined number of clusters (let's assume k clusters) fixed a priori to categorize a given data set. To define k centroids, one for each cluster, is the main notion. These centroids should be positioned deftly because different locations yield various effects. The preferable option is to situate them as far apart from one another as you can. Next, each point from a particular data collection is taken and connected to the closest centroid. The first step concludes once there are no vacant points, and an early group is eliminated. At this point, k new centroids that serve as the centers of the clusters produced by the previous phase must be recalculated. The identical data points must be bound again to the subsequent k new centroids that are closest to them. There has been created a loop. This loop may cause it to observe that the k centroids gradually shift positions until no more modifications are made. To put it another way, centroids are no longer in motion. Last but not least, the objective of this procedure is to minimize an objective function, in this case a squared error function. The objective function Where S is a K-cluster partition of the entity set represented by vectors yi (i∈I) in the M-dimensional feature space, consisting of non-empty non-overlapping clusters Sk, each with a centroid ck (k=1,2,…K).

The algorithm has the following steps:

A black and white math equation

Description automatically generated

1. Place some initial k points in the space represented by the objects that are being clustered. These points represent initial group centroids.

2. Assign each object to the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the k centroids.

4. Repeat Step 2 and 3 until the centroids no longer move.[15]

There are different methods to selecting the number of clusters and elbow method is one of the more commonly used methods.[16]

A graph with colored dots and red and blue dots

Description automatically generated with medium confidence

Figure Science map with k-means clustering

From the figure we see that the clustering on our science map has given us quite the clear picture of the grouping of different science papers on how they are distributed on the scientific topics domain. We can see the results and interpret their closeness to one another as them being falling quite close to each other in their target subjects and their distance being the difference the topics they are aiming for. Now that the results are visualised, we can extend the same for multiple journals to increase our data points and increase the results of our science map. The same pipeline is extended to a second journal and the science map for it is plotted as well.

A screen shot of a graph

Description automatically generated

Figure Scatter plot of second journal

The second paper chosen was of smaller in nature and had only a dozen or so in papers being published per edition. The science map of both the papers are then combined to plot a combined science map and the science map combined is shown.

A graph with colored dots and numbers

Description automatically generated

Figure K-means of second journal

A screen shot of a graph

Description automatically generated

Figure Science map results

Once the combined science map is plotted, we get to the final objective of this project which is to provide a facility for the user to map their own paper in the science map and see where their papers position in the space.

For the user to input their own paper into our combined science map we make use of the python library tkinter. This library opens a window to the user when ran and the user can navigate this window to the folder where they have kept their paper. After successfully pointing to the file path, the program displays the folder’s path for confirmation. On confirmation, the program starts the entire pipeline on the selected file. The required text is fetched, the text pre-processing is completed. The processed text is undergone the same modelling stage the rest of the journals were subjected to. The results of the text were processed using the principal component analysis technique for dimensionality reduction and the reduced dimensions were mapped to the combined science map with different icons for better visual clarity for the user on making out were their papers fall under the other science papers in the domain.

# Results

The results are depicted in the figure shown. The different journals are spread across the Euclidean space where some are close together and some are far from the clusters. The results are interpreted such that the from the left the labels are ['Economy', 'Microbiology', 'Chemistry', 'Arts', 'Biology', 'Physics', 'Astronomy', 'Agriculture']. The points at the left most ends are falling more towards the subjects at the start of the topics list and the topics at the rightmost are falling for the latter parts of the topics list. The difference in between the points of the same cluster is indicative of their difference in subtopics or it could be said as the difference in between their content while falling under the same subject. The red cross marks are the indicators for cluster centres. The individual clusters are each colour coded. The user data points are shown by purple points and this user point is falling at the border between cluster 2 and cluster 3 but it lies more to cluster 3.

The results showcases that the zero shot classification model 1 utilising the linguistic model of hugging face transformers was proven to be superior to the other models using the LDA method and the other incremental approaches. The chosen model provided results which were the confidence numerals for each of the journal to fall in each of the mentioned subjects. The results were of higher dimensions and these needed dimensionality reductions for it to be plotted. Principal component analysis was used to reduce the dimensions and the results were plotted on a scatter plot to create our science map.

Clustering techniques such as k-means clustering was used to cluster the data points for representations. The Tkinter library was used to allow user to plot their paper among the science map to visualize the relative position of them.

The results showcase the combined science map with papers from different science journals being plotted in a common Euclidean space and their closeness or farness being the difference between their targeted topics. The model utilises many techniques in coherence and is unique in its execution and solution. The project is implemented with efficiency in mind which is showcased by utilizing 4 models and picking the best performing one in both its reliability and time efficiency.

A screen shot of a graph

Description automatically generated

Figure Final science map results combined with user point

# Discussions

The original goals that were set for this project while submitting the project proposal were all successfully completed which is something I am proud about. We can see that all the goals from the figure in project proposal were achieved. There are certainly space for improvements in the number of data points which could have been included or the type of clustering techniques which we could have used but the currently used methods prove adequate. Better modelling techniques for improved efficiency are always a welcome addition.

A close-up of a text

Description automatically generated

Figure Project goals from project proposal

# Conclusions

This project explores a wide array of technologies which are currently the crème of the crop for data science and Machine learning technologies. The idea of a common science map by utilizing the linguistic data modelling to predict the target topic of the text with the goal of plotting the said papers as numeral on a shared Euclidean space to predict their closeness to other papers in the same topic domain was an interesting learning experience. I believe that the science map area is still very much unexplored with lots of room to be studied and developed going forward. The usage of statistical models for data dimensionality reductions, utilising of web crawlers for data collection, data processing using lemmatization techniques, data modelling using neural networks and visualisation techniques all being used in cohesion was a tough but rewarding challenge. The project achieved its intended goals from the proposal stage in a highly satisfactory manner.

# Appendix

The source code for this project can be found at this git lab link-

<https://git.cs.bham.ac.uk/projects-2022-23/axw263>

There is a jupyter notebook file in this location which is the source code and there are two zip files which are the original datasets which are fetched with the help of the web scrapper. The source code should be run on the jupyter notebook and it needs necessary libraries to be pre-installed in the user’s local machine beforehand. If using anaconda navigator, these libraries can be installed via the use of conda install statements. A chrome web driver is necessary for the usage of web scrapper successfully whose version should match the user’s local chrome browser version. Once the proper setup is complete the code should be ready to be executed and should give no errors as it was thoroughly debugged and was executed successfully.

# Acknowledgements

This project was made possible with the guidance of professor Jens Cristian Claussen and professor Vincent Rahli. The project idea was suggested by Professor J.C.Claussen and the changes were made for its execution and technical aspect by my suggestions. This project incorporates a wide range of data science and Machine Learning technologies and utilizes statistical knowledge as well. It has been a great learning experience which provided ample hardships to make it challenging.

# References

[1]Joseph, J.M. and Claussen, J.C., 2014. A model for dynamical evolution of science in space. *arXiv preprint arXiv:1407.8422*.

[2] Shelby Foote, cited after <http://izquotes.com/quote/297238>

[3] Boyack, K.W., Klavans, R. and Börner, K., 2005. Mapping the backbone of science. *Scientometrics*, *64*, pp.351-374.

[4] Garfield, E., 1955. Citation indexes for science: A new dimension in documentation through association of ideas. *Science*, *122*(3159), pp.108-111.

[5] Garfield, E., Sher, I.H. and Torpie, R.J., 1964. The use of citation data in writing the history of science.

[6] Garfield, E. and Merton, R.K., 1979. *Citation indexing: Its theory and application in science, technology, and humanities* (Vol. 8). New York: Wiley.

[7] Rousi, A.M. and Laakso, M., 2022. Overlay journals: A study of the current landscape. *Journal of Librarianship and Information Science*, p.09610006221125208.

[8] Secko, D.M., Amend, E. and Friday, T., 2013. Four models of science journalism: A synthesis and practical assessment. *Journalism Practice*, *7*(1), pp.62-80.

[9] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[10] Shlens, J., 2014. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

[11] Jolliffe, I.T., 2002. *Principal component analysis for special types of data* (pp. 338-372). Springer New York.

[12] Jolliffe, I. T. (2002). [*Principal Component Analysis*](http://link.springer.com/10.1007/b98835). Springer Series in Statistics. New York: Springer-Verlag. [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1007/b98835](https://doi.org/10.1007%2Fb98835). [ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)) [978-0-387-95442-4](https://en.wikipedia.org/wiki/Special:BookSources/978-0-387-95442-4).

[13] Geiger, B.C. and Kubin, G., 2013, January. Signal enhancement as minimization of relevant information loss. In *SCC 2013; 9th International ITG Conference on Systems, Communication and Coding* (pp. 1-6). VDE.

[14]  [*"Engineering Statistics Handbook Section 6.5.5.2"*](http://www.itl.nist.gov/div898/handbook/pmc/section5/pmc552.htm)*. Retrieved 19 January 2015.*

[15] Kodinariya, T.M. and Makwana, P.R., 2013. Review on determining number of Cluster in K-Means Clustering. *International Journal*, *1*(6), pp.90-95.

[16] Andrew, N., 2012. Clustering with the k-means algorithm. *Machine Learning*.