Topic Modelling on BBC News Articles

Almabetter Capstone Project

Team: AETHER

Ankur Bhattacharjee

Bhabakrishna Talukdar

Mayank Tiwari

Md Suhel Ansari

Pratheek T M

Dated: 26/03/2022

Outline:

News is vital for our everyday life as it keeps us aware of changes in the world, and we make choices based on how we perceive the world to be and in response take actions. News has grown to take on more forms than ever. There are a variety of news those we come across and seem important to be informed about such as, politics, health and lifestyle, sports, international, business and so on.

BBC News is one of the most prominent news broadcasting company. It's an operational business division of the British Broadcasting Corporation responsible for the gathering and broadcasting of news and current affairs in the UK and around the world.

We were given a few news articles separated by their topics such as, business, politics, entertainment, sports and technology. Our job is to use unsupervised clustering algorithms to segregate and find the hidden topics in these news articles.

Problem Statement:

- We build an unsupervised model to cluster the news articles using algorithms such as, LDA(Latent Dirichlet Allocation) and LSA(Latent Semantic Analysis).
- We are provided with folders specifically consisting news articles of different types in text format. We combine them to get a proper dataset and finally we convert it to csv format.
- Performing some EDA on the dataset to understand the distribution of different types of news articles.
- Preparing our dataset for Unsupervised Machine Learning models by cleaning the articles like removing stopwords etc.
- Clustering using LDA and LSA.

Introduction:

Topic modelling is an approach to discover abstract topics present in a corpus of text documents. The hidden topics or themes from a collection of documents can be recognized using Natural Language Processing(NLP) and Clustering techniques. The dataset consists of a corpus of news articles from BBC News which are segregated into five topics, business, entertainment, politics, sports and technology. The objective is to build an unsupervised machine learning model using clustering algorithms to identify hidden topics from news articles.

The whole process of Topic Modelling on BBC News articles begins with data preprocessing where all the text documents are aggregated from different topics to create a single dataset and duplicate articles are removed, data exploration is done through both collection of News articles and respective topics, data cleaning such as removal of stop-words, short-length words, special characters, numbers, additional white-spaces, newlines and other unnecessary elements from texts, Lemmatization to reduce words to their root form, Vectorization is performed using TF-IDF vectorizer and finally modelling using clustering algorithms such as Latent Dirichlet Allocation(LDA) and Latent Semantic Analysis(LSA).

Data Description:

The Data folder consists of 2225 documents from the BBC news website corresponding to stories in five topics from 2004-2005.

- Topics: Business, Entertainment, Politics, Sport, Tech.
- Features of dataset after combining the above folders are:

    1. Unnamed - serial number of news articles
    2. News - The news article
    3. Type - Topic of news article

Steps Involved:

Step 1 - Data Exploration:

- Checking the initial information of the dataset, such as number of rows and columns, data-types of columns and checking for null-values.
- Checking the description of features, to have an idea about their distributions.
- Dropping the column 'Unnamed:0' since this column was just serial numbers and all were unique.
- Removing 98 duplicate news articles from the combined dataset.
- Creating columns 'length' and 'word-count' of news articles.

Step 2 - Exploratory Data Analysis:

- Finding the percentage of types of news in the dataset and plotting it.
- Plotting the distribution of lengths of different types of news.
- Plotting the individual histograms for the distribution each news type.
- Plotting top 20 most frequent words in the corpus.
- Plotting Top 20 most frequent words in the corpus after removal of stopwords.
- Plotting top 20 bigram/trigram words to understand most frequent/hot topics.

Step 3 – Data Pre-processing:

- Removal of special characters, new-line characters, numbers, extra white-spaces.
- Conversion of all words to lowercase.
- Performing Lemmatization to replace all words with their root words.
- Removing short-length words(words-length less than 3)
- Removal of stopwords and some additional non-contextual words
- The data has almost been reduced to 50% after Data Pre-processing.

Step 4 - Model Implementation:

- Using TFIDF vectorizer performing LDA and plotting pyLDAvis visualization
- Using t-sne algorithm performing LSA and plotting the scatter-plot of resultant clusters.
- Training a set of classification models on the dataset and evaluating their results.
- Selection of the best model based on Evaluation Metrics.

Data Exploration

Handling Duplicate Values

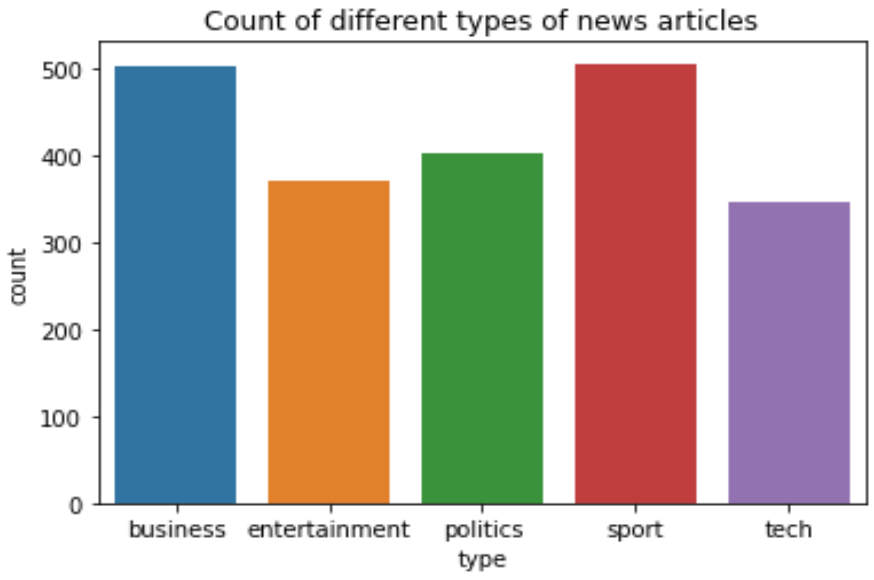
	news	type
count	2225	2225
unique	2127	5
top	b'MPs issued with Blackberry threat\n\nMPs wil...	sport
freq	2	511

Observation:

- The dataset has no null values but it has 98 duplicate news articles.
- The new shape of the dataset is 2127 rows with 3 columns after removal of duplicates

Exploratory Data Analysis

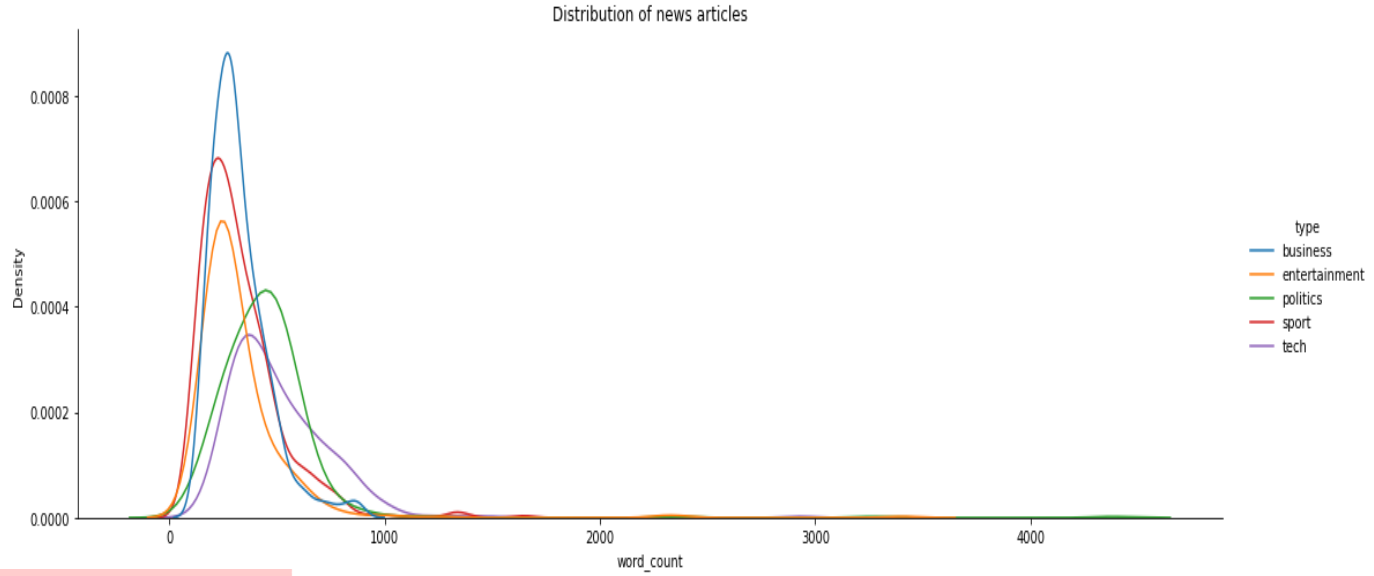
Count of different types of news articles



Observation:

- **Topics business and sports have the more number of articles in the dataset.**

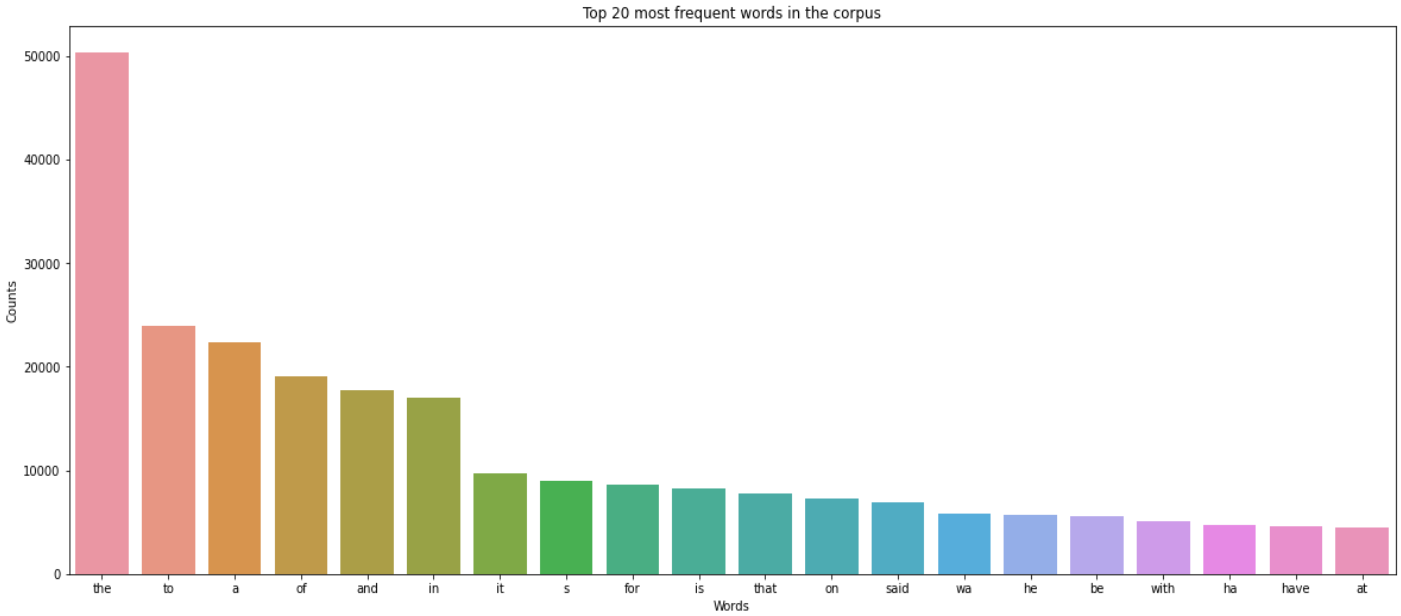
Distribution of word-counts of different topics.



Observation:

- **Business has more articles of lesser word-counts.**
- **Politics and Entertainment articles are bigger than any other topic.**
- **The curve shows most of the articles are of length 500 words approx.**

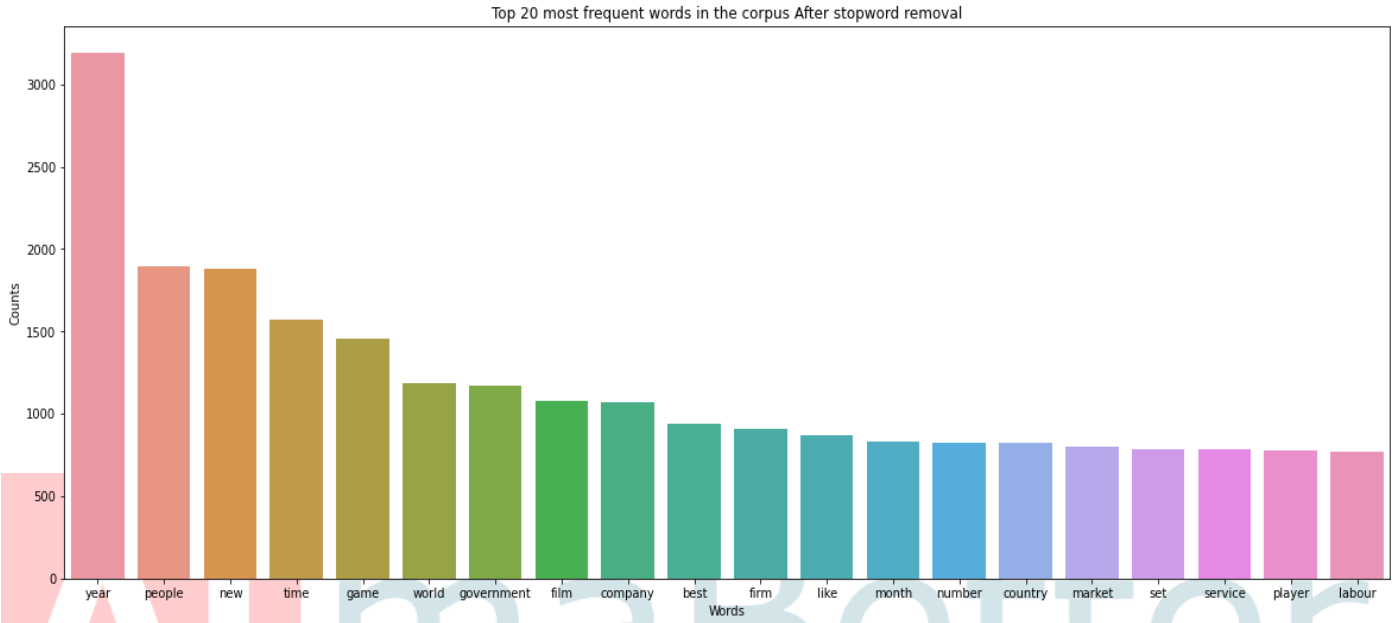
Top 20 most frequent words in the corpus after performing Lemmatization



Observations:

- **The graph shows that stopwords are most frequent in the whole dataset. So these stopwords need to be removed.**
- **Also the short length (< 3) words are most frequent.**

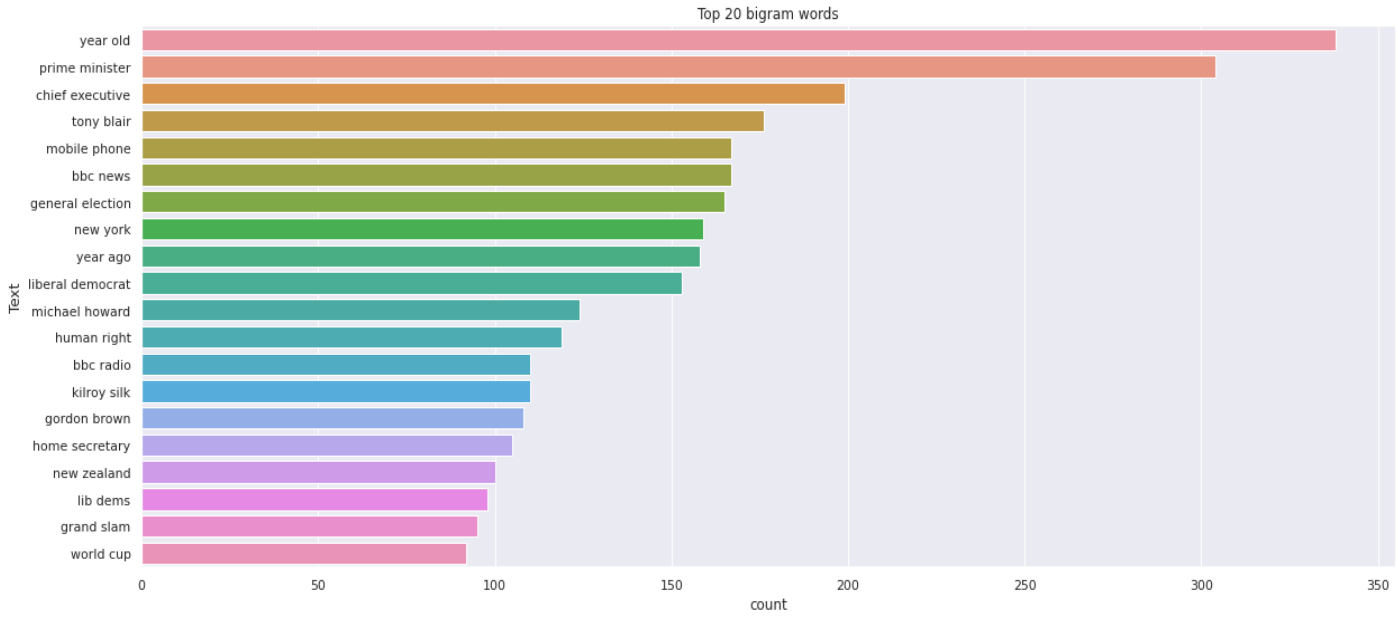
Top 20 most frequent words in the corpus after removing stopwords.



Observations:

- **We can see that after removal of stopwords frequencies of contextual words are more apparent in our Count-plot.**

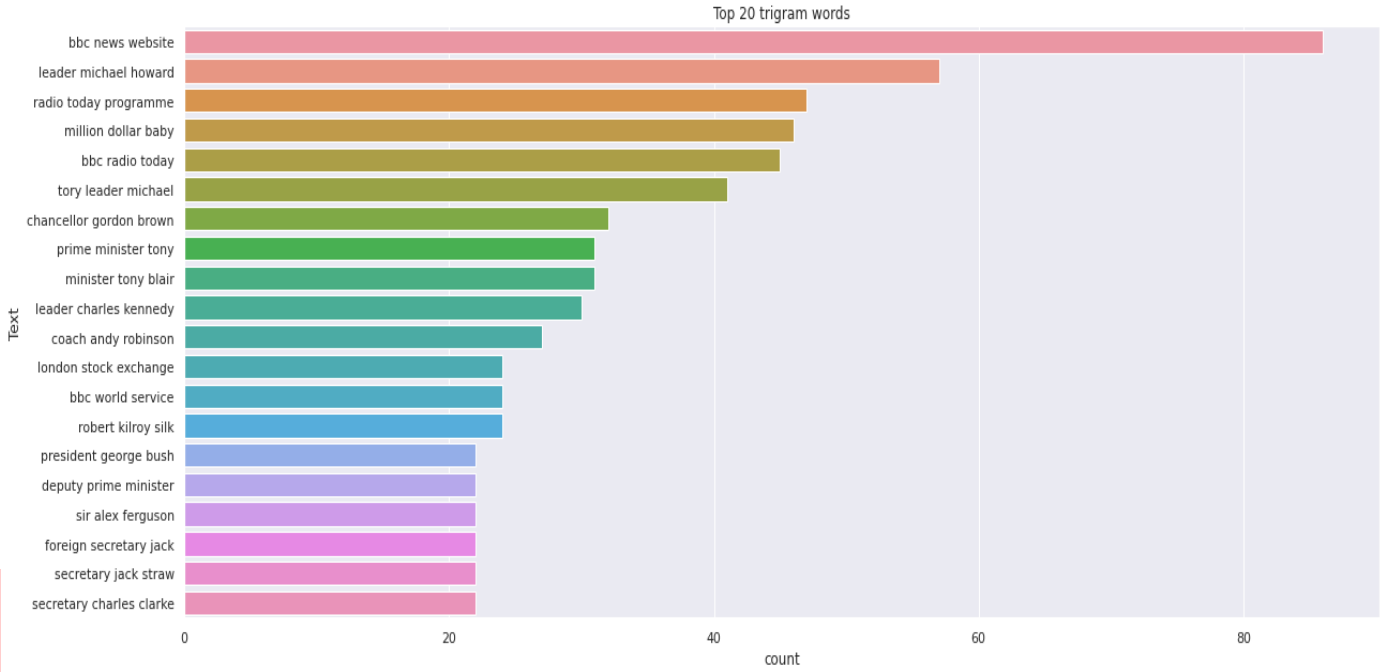
Plotting top 20 bigram words.



Observations:

- **These are top 20 most frequent bigrams (pairs of words) in articles across different topics.**

Plotting top 20 trigram words.



Observations:

- **These are top 20 most frequent trigrams (triplets of words) in articles across different topics.**

Cleaning of Articles

Removal of special characters etc, stop-words:

- We have removed the special characters like, !, @, #,$,%,^,&,*,(,),_,+,?,/ etc, new-line characters, numbers and extra white-spaces in the articles as they were of no use for our analysis.
- **Stop-words:** Stop words are a set of commonly used words in a language. Examples of stop words in English are “a”, “the”, “is”, “are” and etc. Stop words are commonly used in Text Mining and Natural Language Processing (NLP) to eliminate words that are so commonly used that they carry very little useful information.

Lemmatization:

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meanings to one word.

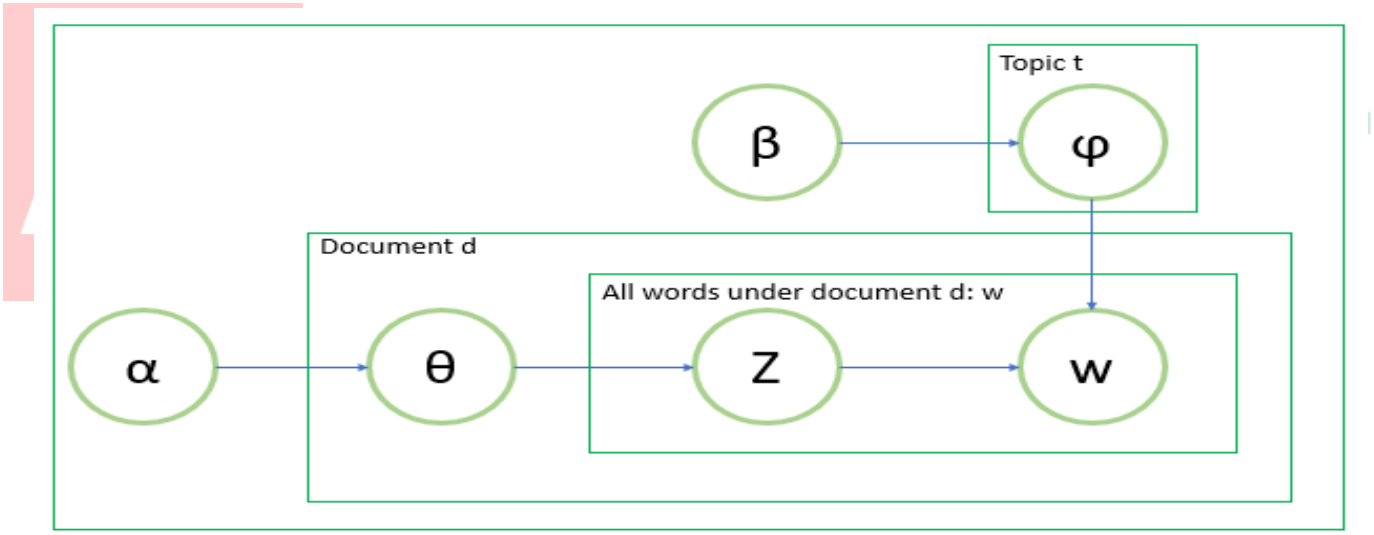
Model Implementation

Latent Dirichlet Allocation.

LDA is introduced by David Blei, Andrew Ng and Michael O. Jordan in 2003. It is unsupervised learning and topic model is the typical example. The assumption is that each document mix with various topics and every topic mix with various words.

Finance	Weather	Arts	Sport
Money	Sunny	Music	World Cup
Stock	Cloud	Piano	Soccer
Trade	Humanity	Calligraphy	Tennis
Market	Temperature	Photography	Sailing

However, “a”, “with” and “can” do not contribute on topic modelling problem. Those words exist among documents and will have roughly same probability between categories. Therefore, stop-words removal is a critical step to achieve a better result.



α: probability on the per-document topic distribution
β: probability on the per-topic word distribution
θ_m : the distribution for document d
φ_k : the word distribution for topic t
Z_{mn} : the topic for the n^{th} word in document d
W_{mn}: the specific word

For particular document d, we get the topic distribution which is θ. From this distribution (θ), topic t will be chosen and selecting corresponding word from φ.

Grid Search CV:

In GridSearchCV approach, machine learning model is evaluated for a range of hyperparameter values. It searches for best set of hyperparameters from a grid of hyperparameters values.

Perplexity:

In general, perplexity is a measurement of how well a probability model predicts a sample. In the context of Natural Language Processing, perplexity is one way to evaluate language models.

Document Term Matrix:

A document-term matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms.

Vectorization:

Word Embeddings or Word vectorization is **a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers which used to find word predictions, word similarities/semantics**. The process of converting words into numbers are called Vectorization

TF-IDF Vectorizer:

TF-IDF stands for Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (data-set).

Terminologies:

- **Term Frequency:** In document d, the frequency represents the number of instances of a given word t. Therefore, we can see that it becomes more relevant when a word appears in the text, which is rational. Since the ordering of terms is not significant, we can use a vector to describe the text in the bag of term models. For each specific term in the paper, there is an entry with the value being the term frequency.

The weight of a term that occurs in a document is simply proportional to the term frequency.

$$tf(t,d) = count\ of\ t\ in\ d\ /\ number\ of\ words\ in\ d$$

- **Document Frequency:** This tests the meaning of the text, which is very similar to TF, in the whole corpus collection. The only difference is that in document d, TF is the frequency counter for a term t, while df is the number of occurrences in the document set N of the term t. In other words, the number of papers in which the word is present is DF.

$$df(t) = occurrence\ of\ t\ in\ documents$$

- **Inverse Document Frequency:** Mainly, it tests how relevant the word is. The key aim of the search is to locate the appropriate records that fit the demand. Since tf considers all terms equally significant, it is therefore not only possible to use the term frequencies to measure the weight of

the term in the paper. First, find the document frequency of a term t by counting the number of documents containing the term:

$$df(t) = N(t)$$

where,

df(t) = Document frequency of a term t

N(t) = Number of documents containing the term t

Term frequency is the number of instances of a term in a single document only; although the frequency of the document is the number of separate documents in which the term appears, it depends on the entire corpus. Now let's look at the definition of the frequency of the inverse paper. The IDF of the word is the number of documents in the corpus separated by the frequency of the text.

$$idf(t) = N/\,df(t) = N/N(t)$$

The more common word is supposed to be considered less significant, but the element (most definite integers) seems too harsh. We then take the logarithm (with base 2) of the inverse frequency of the paper. So *if* of the term *t* becomes: $idf(t) = log(N/\,df(t))$

- **Computation:** Tf-idf is one of the best metrics to determine how significant a term is to a text in a series or a corpus. tf-idf is a weighting system that assigns a weight to each word in a document based on its term frequency (tf) and the reciprocal document frequency (tf) (idf). The words with higher scores of weight are deemed to be more significant.

Usually, the tf-idf weight consists of two terms-

1. **Normalized Term Frequency (tf)**
2. **Inverse Document Frequency (idf)**

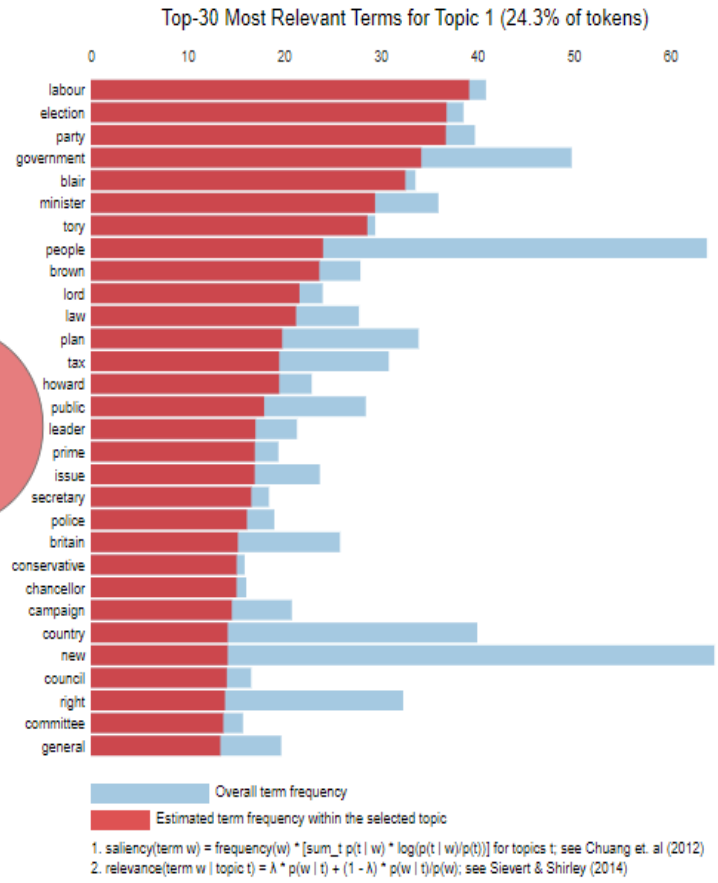
$$tf - idf(t,d) = tf(t,d) * idf(t)$$

In python tf-idf values can be computed using *TfidfVectorizer()* method in *sklearn* module.

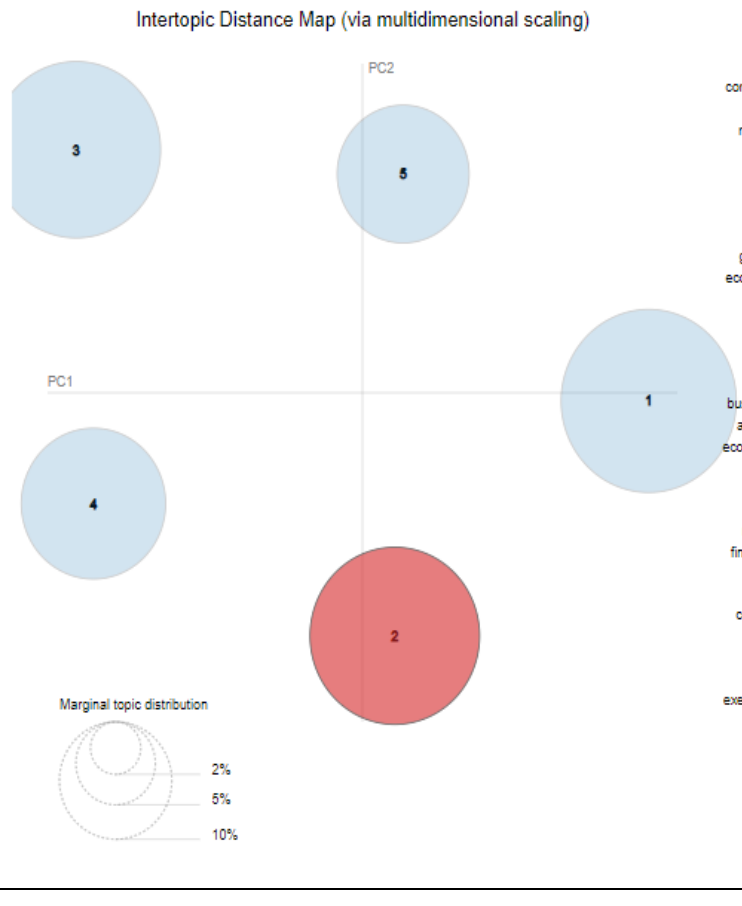
Implementation.

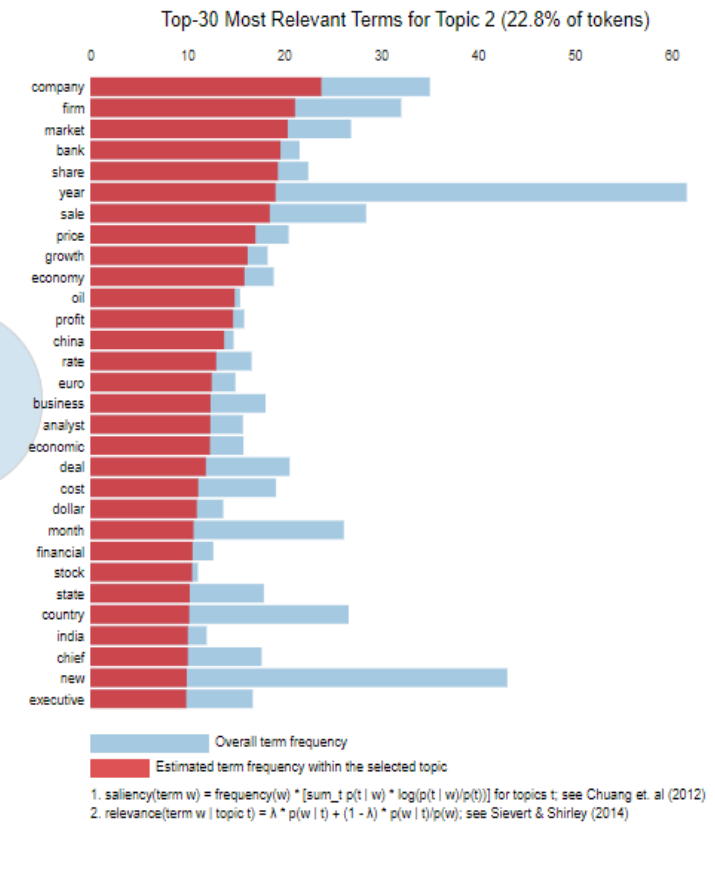
- We got the perplexity score to be 1431.693.
- We found from hyper-parametric tuning the number of clusters n = 5 is best being classified.

Topic 1: Politics

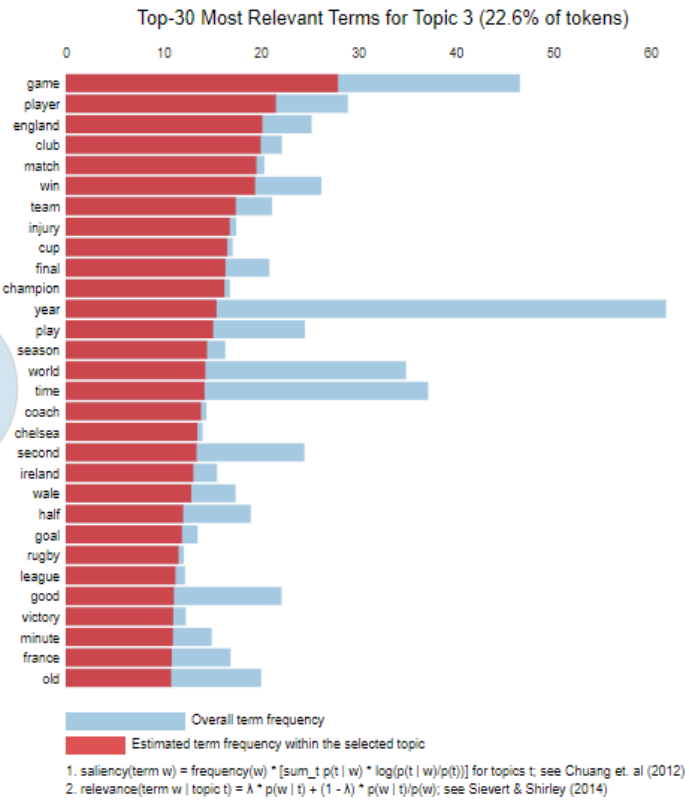


Topic 2: Business

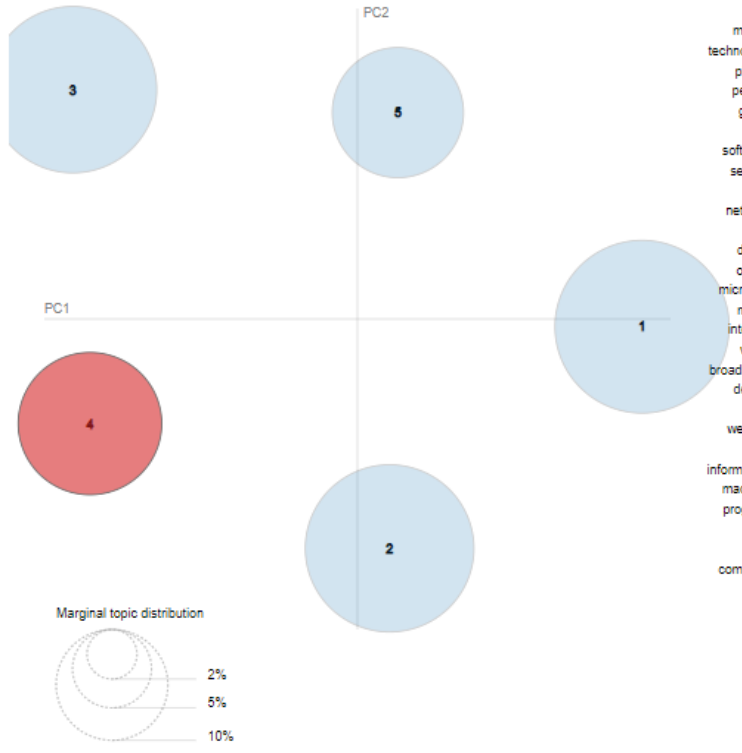


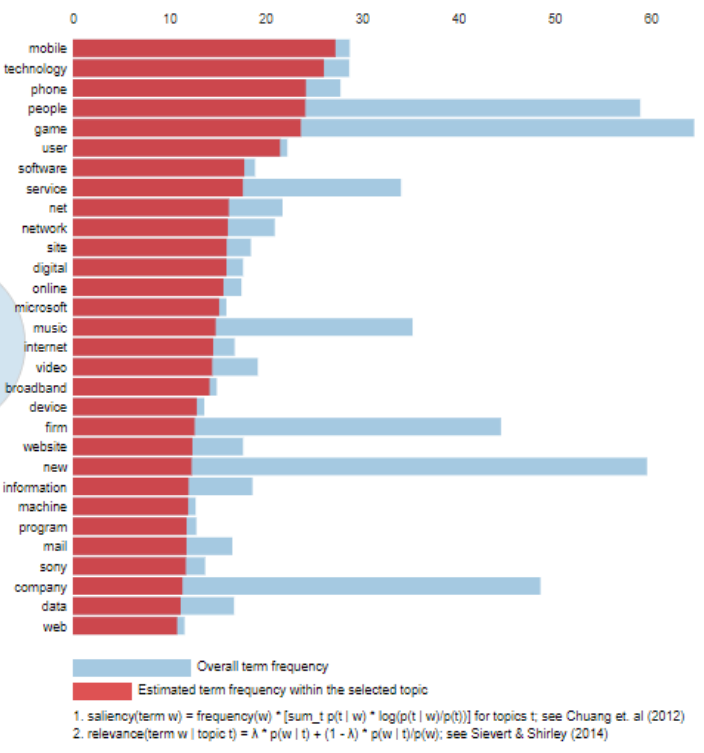


Topic 3: Sports

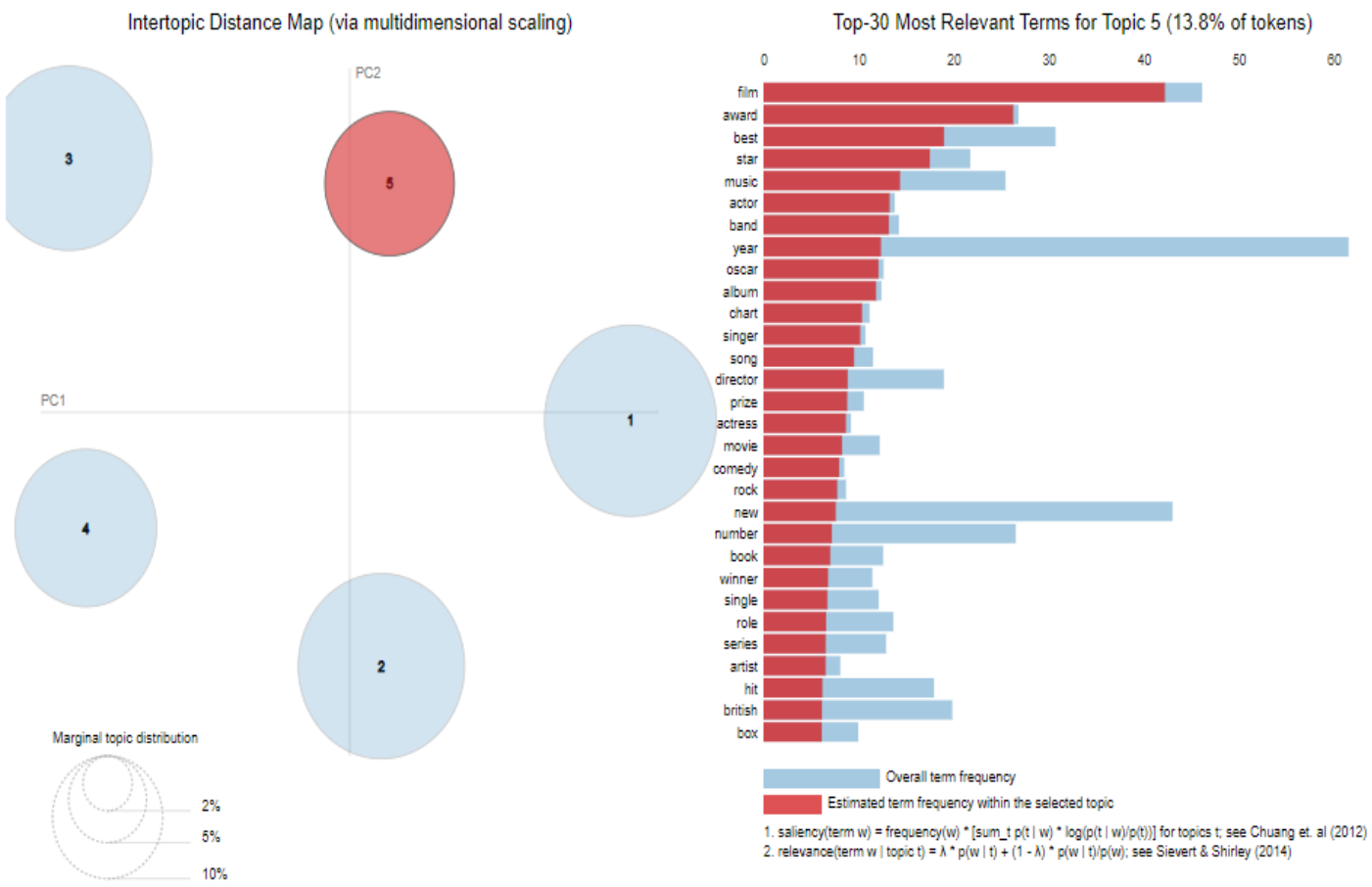


Topic 4: Tech





Topic 5: Entertainment



Observations:

- **LDA performs well & shows 5 different clusters present in the Corpus.**

Latent Semantic Analysis.

Singular Value Decomposition:

The Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices. It has some interesting algebraic properties and conveys important geometrical and theoretical insights about linear transformations. It also has some important applications in data science.

Mathematics behind SVD

The SVD of m x n matrix A is given by the formula:

$$A = UWV^T$$

where:

- U: *mxn* matrix of the orthonormal eigenvectors of AA^T.
- V^T: transpose of a *nxn* matrix containing the orthonormal eigenvectors of $A^{(T)A}$

- W: a *nxn* diagonal matrix of the singular values which are the square roots of the eigenvalues of $A^T A$.

LSA

LSA for natural language processing task was introduced by Jerome Bellegarda in 2005. The objective of LSA is reducing dimension for classification. The idea is that words will occurs in similar pieces of text if they have similar meaning. We usually use Latent Semantic Indexing (LSI) as an alternative name in NLP field.

First of all, we have m documents and n words as input. An m x n matrix can be constructed while column and row are document and word respectively. You can use count occurrence or TF-IDF score. However, TF-IDF is better than count occurrence in most of the time as high frequency do not account for better classification.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

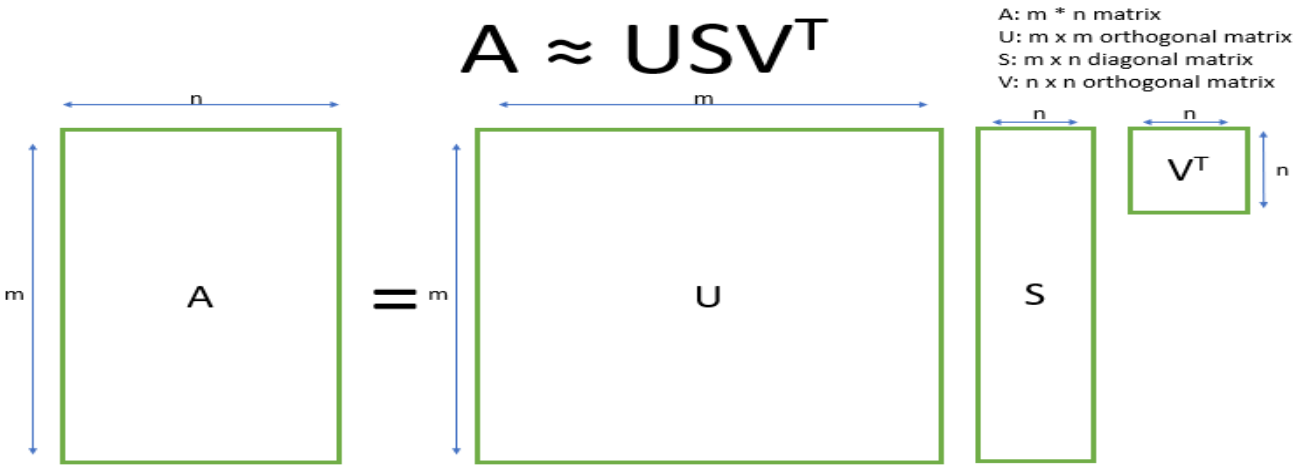
$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

The idea of TF-IDF is that high frequency may not able to provide much information gain. In another word, rare words contribute more weights to the model. Word importance will be increased if the number of occurrence within same document (i.e. training record). On the other hand, it will be decreased if it occurs in corpus (i.e. other training records).

The challenge is that the matrix is very sparse (or high dimension) and noisy (or include lots of low frequency word), so truncated SVD is adopted to reduce dimension.



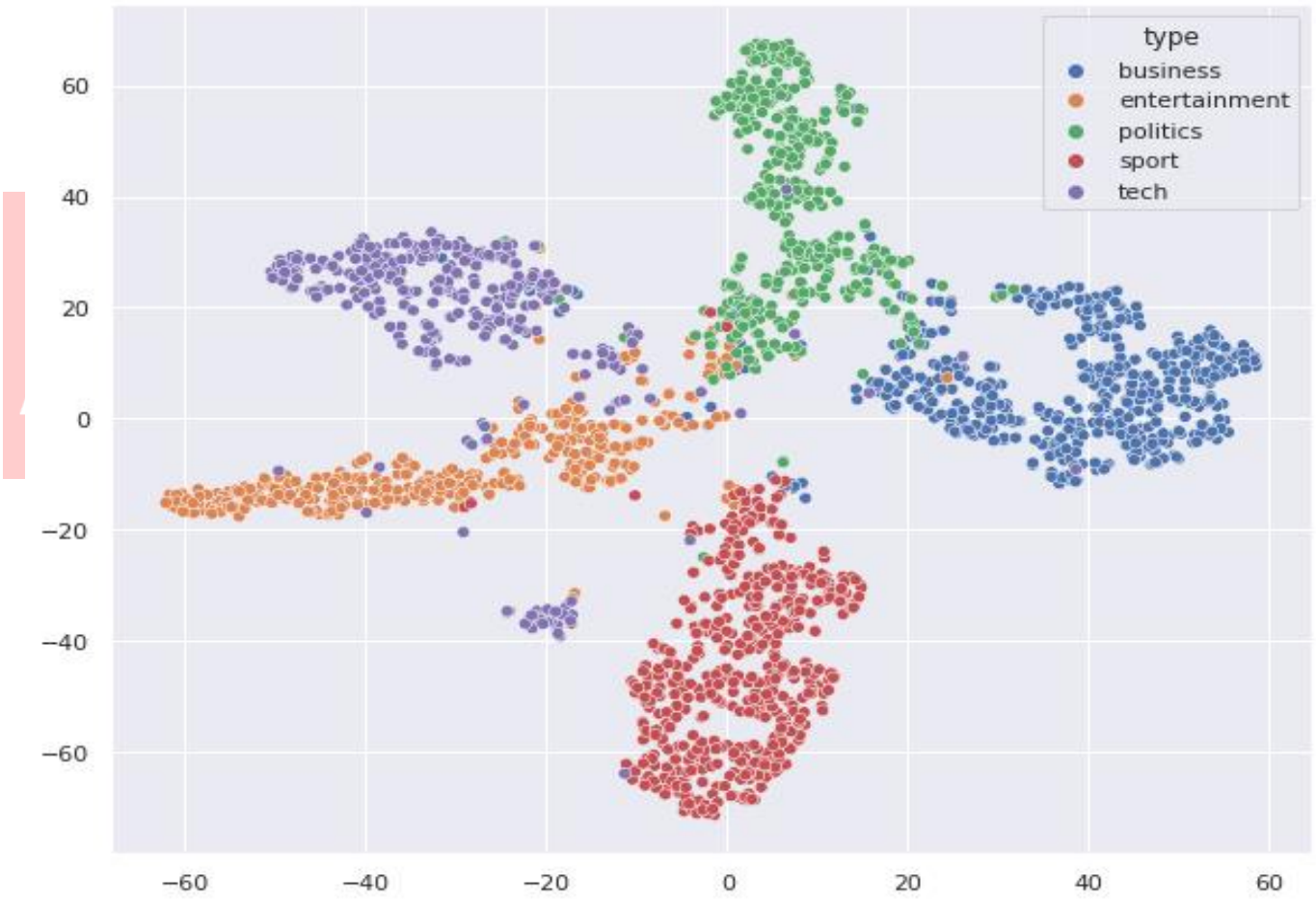
The idea of SVD is finding the most valuable information and using lower dimension t to represent same thing.

t-SNE: t-Distributed Stochastic Neighbour Embedding (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data. In simpler terms, t-SNE gives you a feel or intuition of how the data is arranged in a high-dimensional space. It was developed by Laurens van der Maatens and Geoffrey Hinton in 2008.

It is a nonlinear dimensionality reduction technique well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions.

Implementation

Scatterplot from LSA



Observations:

- **LSA also shows the 5 different clusters present in the scatterplot, but the clusters are not that clear as some elements of few clusters are actually scattered into other clusters.**

Conclusion:

To sum up this Topic Modelling process, after creating a segregate dataset from the corpus of news articles, we found that the dataset consisted of 2225 mews articles from 5 different topics but, as there were 98 duplicate news articles, the dataset is reduced to 2127 news articles. From initial explorations it can be seen that topics business and sports have higher proportion of news articles compared to other topics. Articles of topic business are shorter in terms of word-counts and those of topics politics and entertainment are longer. Most of the articles in the corpus have word-counts close to 500.

Exploring the dataset even further, the plot of top 20 frequent unigrams(words) highlighted the interference of stop-words and short-length words(words with lengths less than 3) on our analysis. So, plotting top 20 frequent unigrams, bigrams and trigrams, after removal of such words along with special characters, numbers, additional white-spaces and new lines verified that the dataset now is clean with 50% of the dataset being reduced and we can proceed to next step. Also Lemmatization reduced each word to their root form in order to group them for coming analyses.

Vectorization is performed using TF-IDF vectorizer. Finally, models with clustering algorithms Latent Dirichlet Allocation(LDA) and Latent Semantic Analysis(LSA) are built. LDA is able to cluster the news articles into 5 different topics which is visualised using pyLDA visualisation plot and individual wordclouds of 5 topics. LSA is also able to cluster the articles into 5 topics, but the scatterplot and the wordclouds of different topics show that clustering is not as good as LDA. Hence we conclude that LDA is the best algorithm for Topic Modelling on BBC News Articles.

References:

i. GeeksForGeeks
ii. Towardsdatascience