Pubmed Topic Modeling

Topic Modeling is a type of statistical model for discovering the abstract "topics" that occur in a collection of articles or documents. The "topics" produced by topic modelling techniques are clusters of similar words. A topic model captures this intuition in a mathematical framework, which allows examining a set of documents and discovering, based on the statistics of the words in each, what the topics might be and what each document's balance of topics is.

The topic modelling in the notebook attached is done via the **Latent Dirichlet Allocation algorithm (LDA)** which is similar to another very popular dimensionality reduction technique called Principal Component Analysis (PCA). Latent Dirichlet Allocation (LDA) is a popular topic modelling technique to extract topics from a given corpus. The term latent conveys something that exists but is not yet developed. In other words, latent means hidden or concealed.

In the attached notebook I will be making topic modelling for Pubmed Articles with their titles, the dataset has 5000 articles with various medical topics: Cancer, Covid, Treatments and Research. The output is a data frame with the top 3 topics in each article and the contribution of that topic.

The pipeline starts with loading the dataset as **pandas** and doing some exploratory data analysis to know the most repeated words and have an idea of what is needed for preprocessing the data.

No null values were detected in the titles or articles, so no null handling was needed in the dataset.

Preprocessing:

- In the preprocessing **Stopwords** are words that are commonly used and were removed. Using the popular **NLTK** package in python, imported the stopwords into the English language and saved them. Then used them for modelling purposes. The stopwords were extended with a list from the SciKit learn library.
- After stopwords removal comes **distracting characters removal** using **regex**.
- All **punctuations** need to be **removed**, so a function was written for that and the **Gensim package** is used for that.
- **Gensim's Phrases** method was also used for **Bigrams**, which are two words frequently occurring together in the document and **Trigrams** which are 3 words frequently occurring.
- **Spacy** was used for **lemmatization** which is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. Nouns, adjectives, verbs and adverbs are the only parts of the sentences that are allowed.

Modelling:

Modelling starts with **LDA**, its approach to topic modelling is it considers each document as a collection of topics in a certain proportion. And each topic is a collection of keywords, again, in a certain proportion.
Once you provide the algorithm with the number of topics, all it does it to rearrange the topics distribution within the documents and keywords distribution within the topics to obtain a good composition of the topic-keywords distribution.

Evaluation:

After the LDA model is built, there has to be a metric to evaluate how good the model is. The evaluation of the model is done based on 2 metrics.

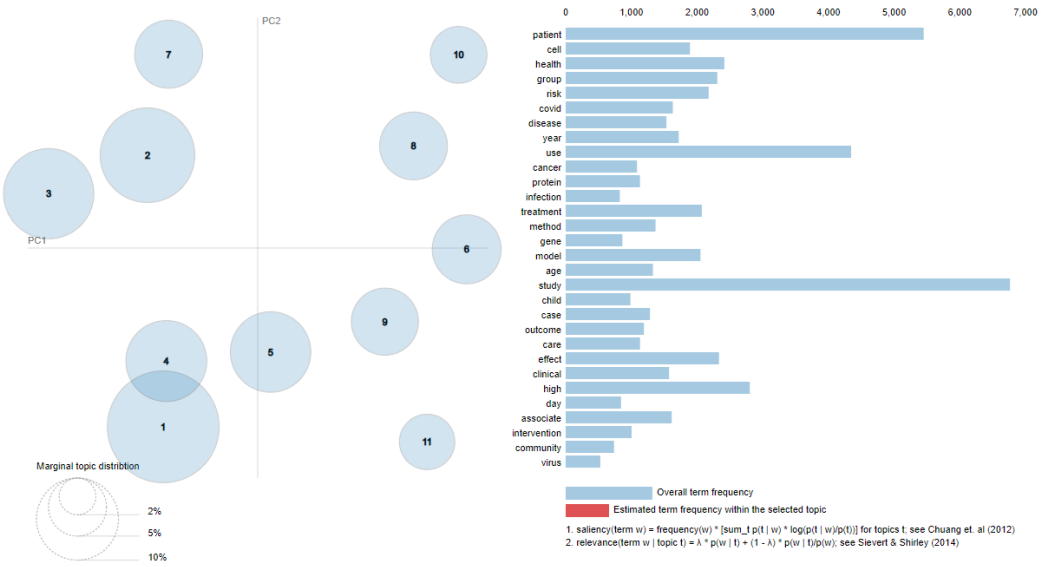
1) **Perplexity** - This is a statistical measure of how well a probability model predicts a sample. As applied to LDA, for a given value of k, you estimate the LDA model. Then given the theoretical word distributions represented by the topics, compare that to the actual topic mixtures, or distribution of words in your documents. - Lower the better.

The model Perplexity: -8.641374900166003

2) **Coherence** Score - Is defined as the average/median of the pairwise word-similarity scores of the words in the topic - Higher the better.

The Coherence Score: 0.4598378391939521

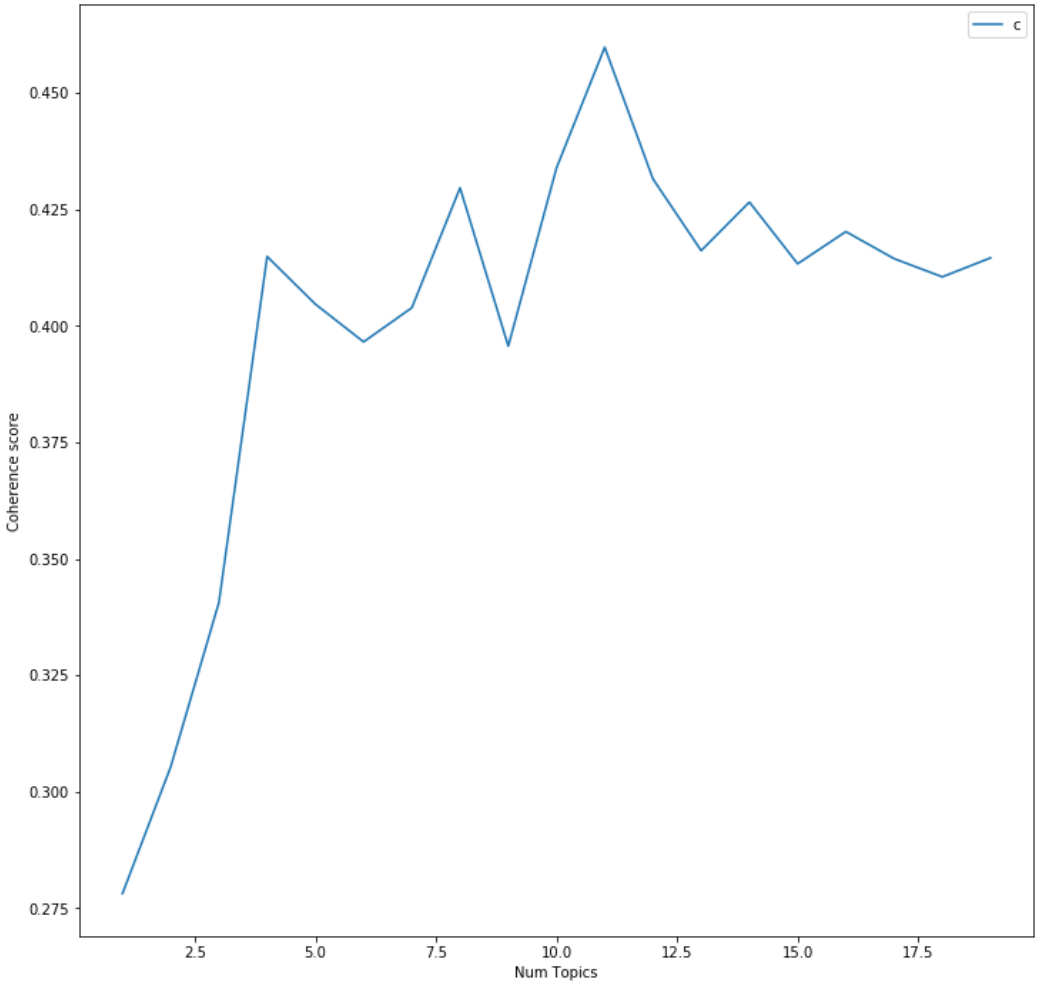
The LDA revealed the visualization shown below generated with the **pyLDAvis**:



Interpreting the topic visualization shows how topics are shown on the left while words are on the right. Here are the main things that should be considered:

Larger topics are more frequent in the corpus Topics closer together are more similar, and topics further apart are less similar. When a topic is selected, the most representative words for the selected topic can be seen. This measure can be a combination of how frequent or how discriminant the word is. Hovering over a word will adjust the topic sizes according to how representative the word is for the topic.

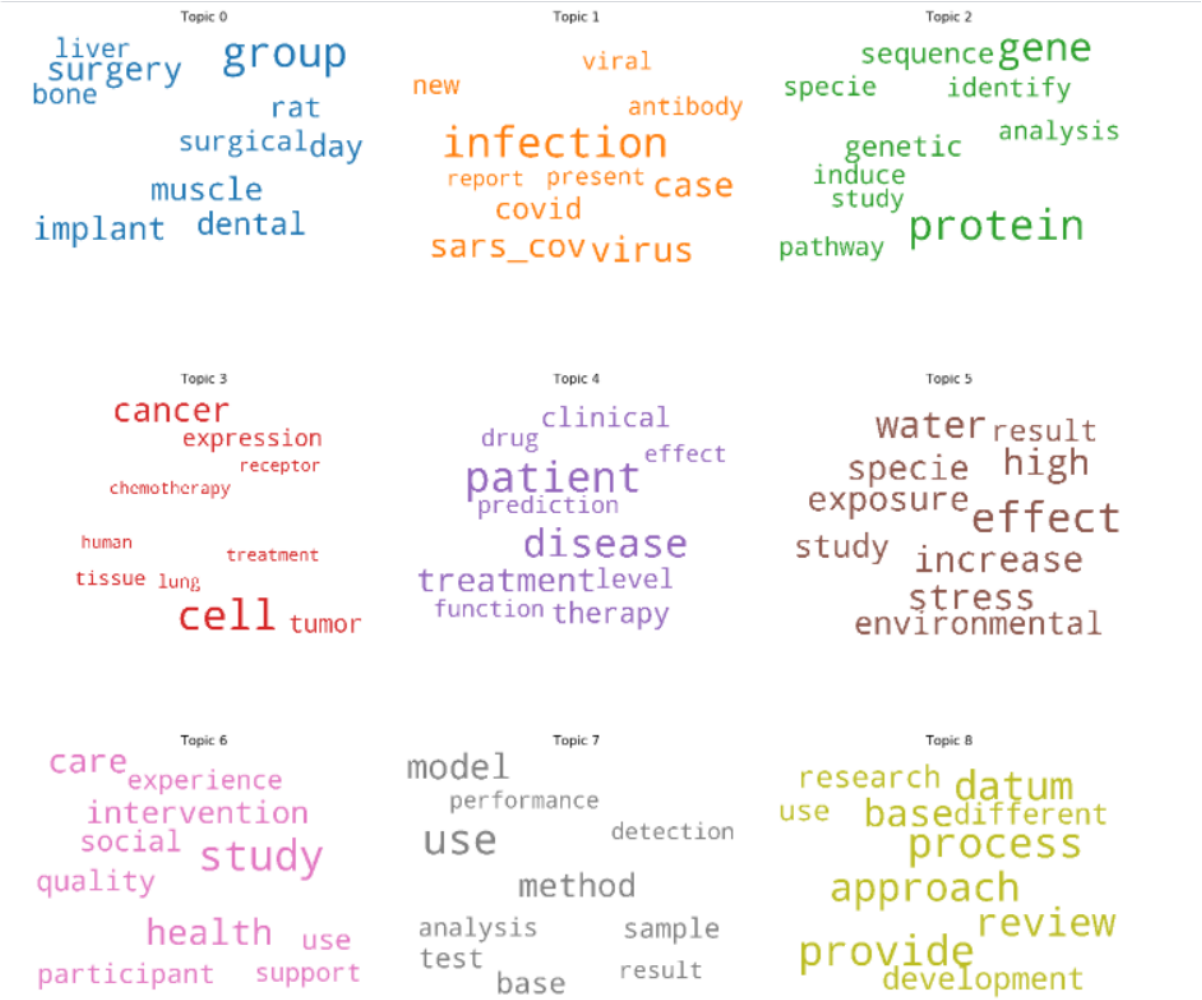
The coherence value for 11 topics was selected based on the function to get the coherence values for the count of topics < 20 which yielded the graph below:



From this function, the Number of **Topics is = 11** has a Coherence Value of 0.4598 is the **highest coherence** reached, which we used in the LDA.

The graph oscillations indicate that further data preprocessing is needed, which I plan to work on in the future.

The topics 0 to 8 had the top 10 words shown in the word cloud below:



The word cloud shows so many synonyms of medical terms such as sars_cov and covid so in the further processing grouping of the domain-specific keywords would be ideal, building tailored word vectors and using **BlueBERT** are 2 approaches that would make perfect sense with the topic modelling and will improve the quality of the results. The **Hierarchical Dirichlet Process** also sounds like a reasonable approach for this dataset since we do not have a fixed number of models. Further hyperparameter tuning of the LDA will improve the results too.