Abstract:

Probabilistic topic models are widely used to discover latent topics in document collections, while latent feature vector representations of words have been used to obtain high performance in many NLP tasks. In this document, we extend Latent Dirichlet Allocation model by incorporating latent feature vector representations of words trained on very large corpora to improve the word-topic mapping.

Introduction:

Natural language processing is the processing of languages used in the system that exists in the library of nltk where this is processed to cut, extract and transform to new data so that we get good insights into it. Topic modelling is recognizing the words from the topics present in the document or the corpus of data. This is useful because extracting the words from a document takes more time and is much more complex than extracting them from topics present in the document.

Procedures:

Here in the attached code file, I did the following sequence:

1- Importing packages like Pandas, NumPy, Sklearn and NLTK.
2- Downloading stop words.
3- Read in this dataset and have a look at it.
4- Doing some text processes for the Abstract feature like:

- Stop word removal
- Stemming: using Porter stemmer to convert words to their root.
- Strip punctuation symbols like: “’\.,-_ exc.
- Removing double spacing.

5- Then CounterVectorizer object will be used to transform text to vector form.
6- Building the LDA model which works very well for longer text documents.
7- Assigning the n_components parameter to 3 as it`s required to create 3 topics.
8- Creating a function, which returns a dataframe, to show you the topics we created. Remember that each topic is a list of words/tokens and weights. Here as mentioned before we have 3 topics, 10 words per topic.
9- Evaluating the mode by calculating the approximate log-likelihood as score which is -3823175.9069933514
10- Finally, importing pyLDAvis sklearn package which is firstly converting the vectorized Abstract feature to NumPy matrix. Visualizing the topic modelling performed through LDA into 3 clusters and their word`s weights in a bar graph.