Brief description of solution

The present solution consists of a search engine capable of yielding the most relevant news articles from a dataset of 50 000 news, according to a search keyword.

The engine is based on the fundamentals of NLP and consists of weighting the importance of each meaningful word in the dataset using the TF-IDF algorithm.

In order to compute the score of each word, the whole dataset text is first pre-processed, removing undesired characters such as special characters, wrongly-parsed characters (like '\xe0' and others), removing numbers and punctuation and converting the whole text into lowercase. After this pre-processing, the text is tokenized, i.e., all the text was split by words, and stemming is performed to remove common words (also known as stop-words).

Giving the bag of words of each document, it was organized into a Document-Term-Matrix which allowed to count the number of times each word occurs in each document. This way, the computation of the TF-IDT is straightforward, allowing to obtain the importance of each word of each document in the overall of the dataset. In other words, this way it's possible to establish a metric of a word's importance and better filter what are the documents in the entire dataset where that word is more important.

In order to allow a custom search, this proceeding is performed for the features title, publication, author and content, allowing to search by tags even with multiple tags, e.g. "title: trump publication: CNN".

For the use case in which the query contains more than one word, the software computes the probability of importance (score) for each word then computes the intersection, i.e., p(a and b) = p(a) x p(b).

Although this approach is naïve, it could be improved using a more accurate method such as the cosine similarity or a topic analysis. However, to keep the implementation simple, I decided to go naïve.

Another limitation of the software, which could easily be improved, is the loading time. I.e, for memory reasons, it was not possible to store TF-IDT values in an object, leading the software to compute that values every time it runs. Although in this case is not critical, once the dataset is small, for a larger data-set it could be more critical and it will need to be optimized.

Results:

Query:	Trump
Total Articles found:	19851
Output screenshot:	https://drive.google.com/file/d/1SEIXkBP8UGIo99hMxug6G8w4UvmLUTVZ/view?usp=sharing

Query:	turtle
Total Articles found:	63
Output screenshot:	https://drive.google.com/file/d/1SEIXkBP8UGIo99hMxug6G8w4UvmLUTVZ/view?usp=sharing

Query:	United States of America
Total Articles found:	4047
Output screenshot:	https://drive.google.com/file/d/1SEIXkBP8UGIo99hMxug6G8w4UvmLUTVZ/view?usp=sharing

Query:	title:chicago
Total Articles found:	187
Output screenshot:	https://drive.google.com/file/d/1SEIXkBP8UGIo99hMxug6G8w4UvmLUTVZ/view?usp=sharing

Query:	title:brazil publication:breitbart author:frances
Total Articles found:	26
Output screenshot:	https://drive.google.com/file/d/1SEIXkBP8UGIo99hMxug6G8w4UvmLUTVZ/view?usp=sharing

Query:	title:dead penalty AND content:boston
Total Articles found:	0
Output screenshot:	https://drive.google.com/file/d/1SEIXkBP8UGIo99hMxug6G8w4UvmLUTVZ/view?usp=sharing