Team members:

1. **Kaustubh Pandey** **S20160010041**
2. **Wasim Ishaq Khan** **S20160010107**
3. **Prashant Kumar Mahanta** **S20160010066**
4. **Deepak Kumar** **S20160010022**

News Search Engine

Abstract: - In this project we aim to build a search engine that can search from news feeds(articles). We plan to make a search engine that would be able to handle queries related to dates (when news got posted) and content of the news. In our case we are handling only English based news content.

Key concepts of IR:

1. Handling structured data: The documents would be divided into fields date, title and body. BM25F algorithm will be used to rank the documents.
2. Rank based retrieval: The search engine would retrieve results based on relevance. The most relevant result would appear first.

Impact of this project:

People (especially in big cities) generally don't want to read newspapers as it is time consuming and want to get only the news which is relevant to them. Manually searching for news content on news websites can be sometimes a time-consuming task as there are plenty of news websites, each with numerous articles containing similar content and most of the times irrelevant to the user query. This calls for a need of a rank-based search engine. Thus, to avoid the manual searching of news, our search engine aims to display only that news which are most relevant to the user's query, saving a lot of time. Other than this, our search engine would handle date related queries. One can search a news article related to a topic on a particular date by simply mentioning the topic and a date. The articles related to that date would be given higher priority and would be retrieved. Other date articles will have lower priority.

Algorithm Used:

For the rank-based retrieval we will be using a variant of BM25 which is BM25F. We will not use BM25 due to the following points:

1. BM25 considers documents as unstructured text but in our case, documents will be well structured.
2. There would be no easy way of interpreting the meaning of merging evenly weighted field types. For instance, due to non-linearity nature of term frequencies setting all field weights to 1 does not restore the unstructured scenario of equivalently merging all fields into a large unstructured field.
3. Another problem is that scoring term weights by combining field types opens the question of how to collect global field statistics like IDF values for the individual fields. For instance, titles are short fields and body large fields. Since frequently used terms in body fields may rarely occur in title fields these should receive a high weight in the title score. It turns out that because a frequently used term is defined in relation to a field type, the result would be very unstable IDF statistics.

Why BM25F:
An elegant treatment consists in weighting term frequencies accordingly to their field importance, combining them, and then using the resulting pseudo-frequencies. The above limitations are well handled by this algorithm.