**Problems Statement:**

How many times have you forgotten the movie name you saw only a couple of days back. You still remember the plot and some character names and possibly some locations. With this information we will help you find out the movie you saw.

**Dataset Scraped:**

There are very few places that have all the details of the characters, places etc that occur in a movie. One of them is subtitles and the other is movie plots. We choose to use subtitles as they were directly related to things spoken out by characters.

We cleaned the subtitles file (SRTs) to get the dialogues. At the end, we also appended metadata like movie actors, genre etc.

**Approach:**

After cleaning the SRTs, we tokenize them to get an array, After tokenizations, for every movie , a dict is made. The keys are the tokens and the value the number of times it has occured. Once all the dicts are obtained, we calculate the tf-idf values. All of them are stored as dicts. We save them to disk as an json object, and read from it whenever required.

Since we are using dicts for one movie, the lookup time is practically constant. Hence, finding the movies takes O(n\*|q|) time, where n = number of documents and |q| is number of terms in queries

The indexing takes some time. We first tokenize the entire files, followed by the counting and stuff.

From this, we also generate the list for the autocomplete feature

**Best Parameters**

After some experimentation, we realized that we got the best results for only single word tokenisation, with no stemming for our particular case.

**The Novelty**

The novelty is the data that we have scraped to fulfill the purpose. We have augmented the metadata into the corresponding srt file for each movie. This allows the user to search with genre. In the case (s)he does not remember it, the result will not be necessarily excluded, as in the case of using a relational database alongside.