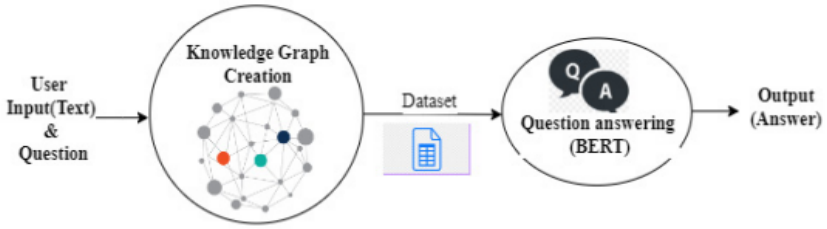
Knowledge Graph based QnA Agent

Karthic Sreenivas A(20pd12), Santhosh V(20pd25) & Vishnupriya G(20pd39)

- **Dataset Description**:
  - As the first step towards preparing the required data for our project, we have resorted to Wikipedia data that contains information about various Films and its associated cast and crew details such as:
    - Music directors
    - Producers
    - Year of release
    - Genre of the movie
    - Languages in which it was released
    - Achievements of the movie
    - Reviews corresponding to the movie, etc.
  - All this data was web scraped using the BEAUTIFULSOUP library after which we were able to model it as a properly organized csv file and use it to build knowledge graphs and subsequently a QnA bot.



- **NLP Techniques Used**:
  - **Tokenization**:
    - Splitting the sentence into individual tokens (words and punctuation).
    - The entries present in the column - 'sentence' of the dataframe were tokenized. These tokenized words are then used further for PoS Tagging.

- **Part-of-speech (POS) Tagging**:
  - Assigning grammatical categories (tags) to each token.
  - We have used the 'StanfordTagger' from the nltk library to perform PoS Tagging.
  - The PoS tags furnished will later be used to generate customized rules that will help identify subjects and predicates.
- **Building Wordclouds**:
  - As the first step, stop words removal is done. Punctuations are also removed from every row entry.
  - Wordcloud for MUSIC DIRECTORS:
    - Iteration is done through all the row entries(sentences), to check if there is presence of the word 'COMPOSED', and if present all these rows are taken into consideration and a wordcloud is built on that.
  - Wordcloud for DIRECTORS/LYRICISTS/GENRE/LANGUAGE:
    - Iteration is done through all the row entries(sentences), to check if there is presence of the word 'WRITTEN', and if present all these rows are taken into consideration and a wordcloud is built on that.
  - Wordcloud for PRODUCERS/PRODUCTION HOUSE:
    - Iteration is done through all the row entries(sentences), to check if there is presence of the word 'PRODUCED', and if present all these rows are taken into consideration and a wordcloud is built on that.

➢ **<u>Entity Pair Extraction</u>**:

- The function extracts entities (ent1 and ent2) from the sentence based on specific syntactic patterns and dependency relations using spaCy.
- It constructs ent1 when it finds a subject and ent2 when it finds an object in the sentence.

- The function uses information from previous tokens (prefix, modifier) to construct the entities in cases of compound words or modifiers.
- The extracted entities are returned as a list containing ent1 and ent2.
- The custom rules used here are:
  - It skips tokens that are punctuations.
  - If a token is a compound word, it updates the prefix accordingly, combining it with the previous token if it's also a compound.
  - If a token is a modifier (ends with "mod" in dependency), it updates the modifier, again combining it with the previous token if it's part of a compound.
  - If a token is identified as a subject (subj in dependency), it constructs ent1 by combining the modifier, prefix, and the token's text.
  - If a token is identified as an object (obj in dependency), it constructs ent2 in a similar way

➢ **Relation Extraction**:

- The Relation Extraction function then defines a pattern using a list of dictionaries. Each dictionary represents a token pattern to be matched in the sentence.
- In this case, the pattern consists of:
  - The **root** of the dependency tree ('DEP':'ROOT'), which typically represents the main verb of the sentence.
  - Optionally, a **preposition** ('DEP':'prep','OP':'?') that may follow the root verb.
  - Optionally, an **agent** ('DEP':'agent','OP':'?') that may appear in the sentence.
  - Optionally, an **adjective** ('POS':'ADJ','OP':'?') that may modify the relation.
  - These patterns are based on the syntactic structure of the sentence, aiming to capture the relationship between entities.

- A pattern-based approach is used along with spaCy's tokenization and dependency parsing capabilities to identify and extract the relation from the input sentence.

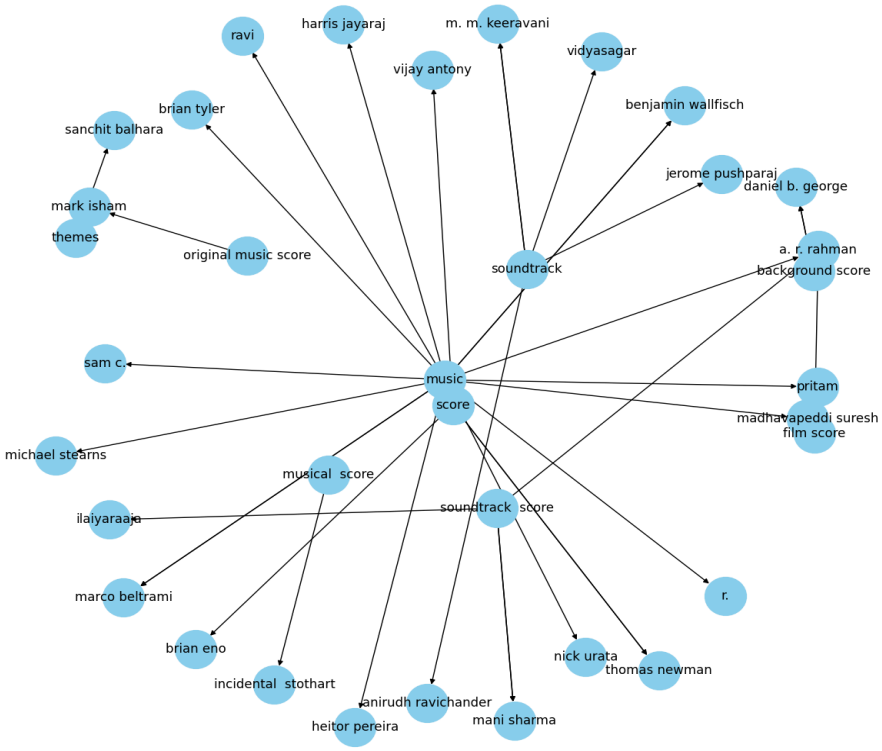
➢ **<u>Constructing Knowledge Graphs</u>**:

- As the first step, we need to tailor-make a dataframe so that knowledge graphs can be constructed with ease.
- The dataframe formulated consists of the columns :
  - Source and Target - derived from Entity Extraction process
  - Edge - derived from Relation Extraction process

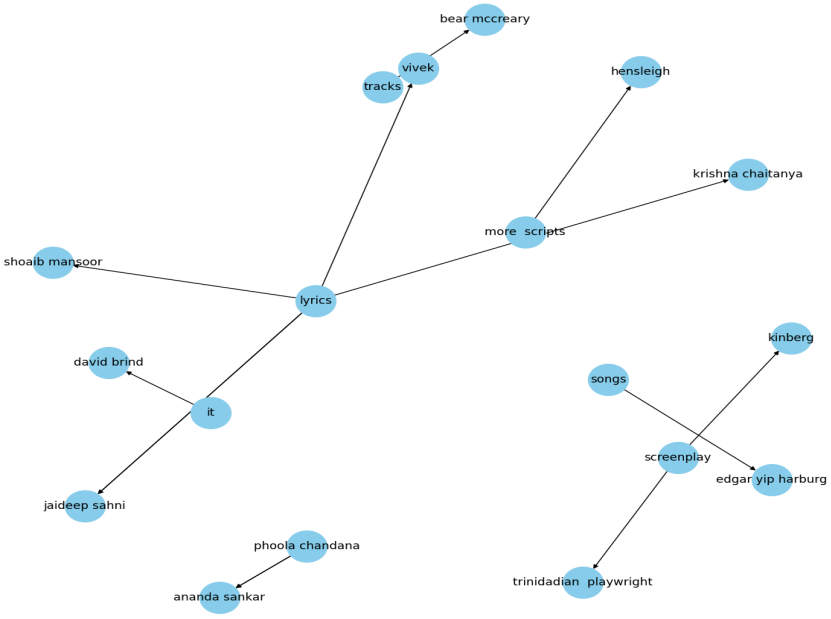
	source	target	edge
731	griffith	nation	's
2795	notable openings	germany	witnessed in
3483	film	excellent reviews	gathered
74	national film awards	prominent film award india	is
4288	lomography	110 film 2011	re-

- In the next step, we have constructed the knowledge graph from the entire dataframe.
- Subsequently, for easier understanding, we have visualized the entire knowledge graph as 3 separate graphs:
  - The first graph contains only the single relation - "**Composed by**"
  - The first graph contains only the single relation - "**Written by**"
  - The first graph contains only the single relation - "**Released in**"

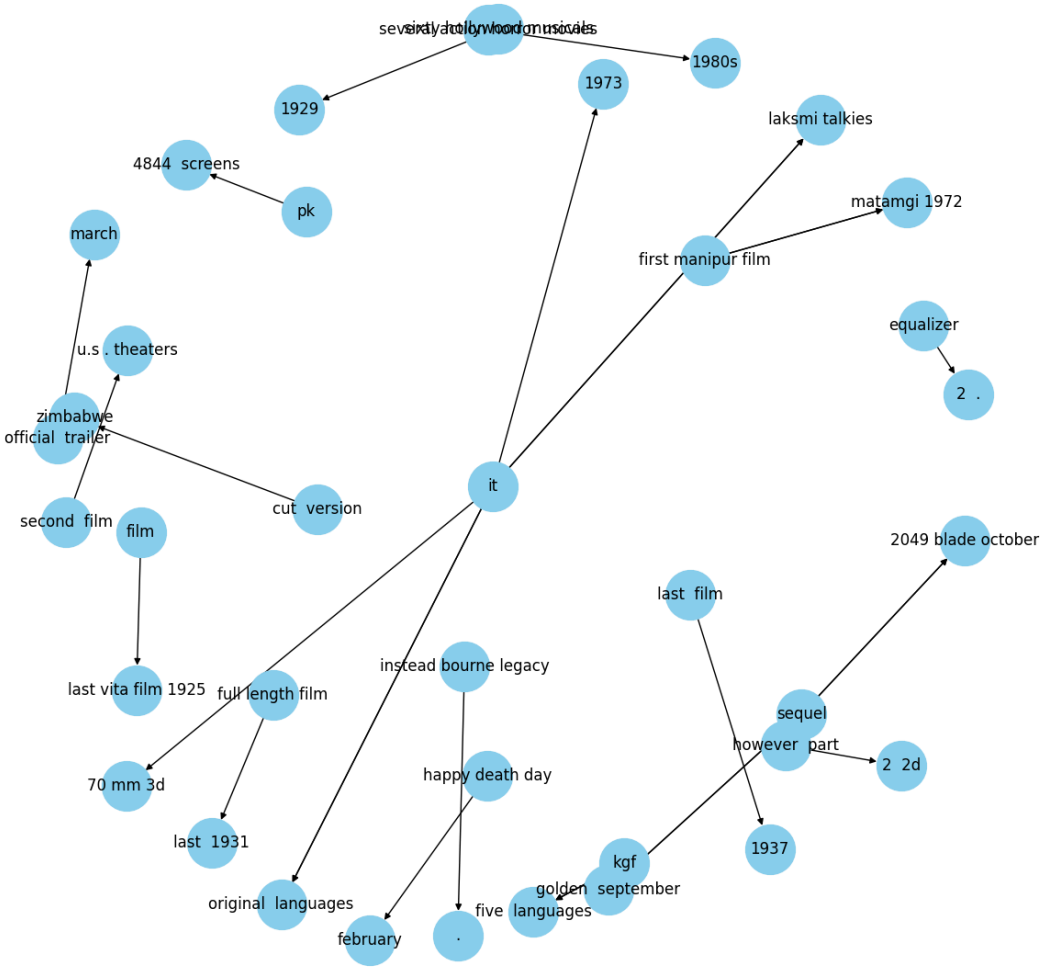
Knowledge Graph corresponding to - “Composed by”



Knowledge Graph corresponding to - “Written by”



Knowledge Graph corresponding to - “Released in”



- ➢ **<u>Fine Tuning RoBERTa - QnA Agent</u>**:
  - What is BERT?
    - BERT, short for Bidirectional Encoder Representations from Transformers, is a Machine Learning (ML) model for natural language processing.
    - What is BERT used for?
      - Can determine how positive or negative a movie’s reviews are. (Sentiment Analysis)
      - Helps chatbots answer your questions. (Question answering)

- Predicts your text when writing an email (Gmail). (Text prediction)
- Can write an article about any topic with just a few sentence inputs. (Text generation)
- Can quickly summarize long legal contracts. (Summarization)
- Can differentiate words that have multiple meanings (like 'bank') based on the surrounding text. (Polysemy resolution)

- **RoBERTa**:
  - It builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates.
  - This implementation is the same as BertModel with a tiny embeddings tweak as well as a setup for Roberta pretrained models.
  - RoBERTa has the same architecture as BERT, but uses a byte-level BPE as a tokenizer (same as GPT-2) and uses a different pre training scheme.
  - Same as BERT with better pre training tricks:
    - dynamic masking: tokens are masked differently at each epoch, whereas BERT does it once and for all
    - together to reach 512 tokens (so the sentences are in an order than may span several documents)
    - train with larger batches
    - use BPE with bytes as a subunit and not characters (because of unicode characters)

- **Finetuning of RoBERTa**:
  - The domain we have chosen to work on is a FILM DATASET scraped from Wikipedia data.
  - The output obtained previously using the knowledge graphs is converted into a triplet form (source, target and edge).
  - The training data is the output obtained from the knowledge graphs (triplet form).

        - On the existing base model for RoBERTa, i.e. roberta-base-squad2, we have performed tuning of the training arguments.
        - Our model is named - roberta-finetuned-subjqa-movies_2, and is stored in our HuggingFace account.
        - Upon supplying the same test data along with the context, we were able to furnish better answers (in terms of relevancy) specifically under the FILM domain in comparison to the base model.

➢ **RESULTS**:

- We have provided the same question along with the context for both the base model as well as the tuned model.

- **Question** : Why is the movie soo confusing?
- **Context** : Inception is an interesting movie but might not be everyone's cup of tea. Dom Cobb (Leonardo DiCaprio) is a man who specializes in dream extractions (think corporate espionage) - going into a shared dream state using a military derived technique with a team and a subject to extract a piece of information from that subject's subconscious mind. This can sometimes involve going into a dream within a dream (or more). I get that these particular dreams were tailor made but they were so orderly it was a bit disconcerting. Much was made of the complexities of this movie and indeed it is complex but for a generation growing up watching Matrix movies (especially the latter two) this is positively straight forward - and that's not a bad thing.

- **Answer (from tuned model)** : is an interesting movie but might not be everyone's cup of tea
- **Answer (from base model)** : they were so orderly

➢ **ADDITIONAL REFERENCE (Wandb results for the neural network):**