Learning Journals And Project Report For Project 6

Data source : https://www.kaggle.com/c/quora-question-pairs/data

Problem Statement And Problem Formulation

- For this problem statement, we are taking the use case of Quora Similar Question Recommendation System.
- The goal of this project is to predict which of the provided pairs of questions contain two questions with the same meaning.
- Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded questions.
- Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question.
- The main idea of this project is to create a ranking algorithm to recommend images or text based on the user's query.

Objectives

For solving this problem statement, we first need to divide this problem statement as :

  - **Sentence Encoding(Converting available text corpus data into vector representation).**
  - **Encoding the user's query into similar vector representation as above.**
  - **Calculate similarity between two vector representations.**
  - **Rank the sentences according to the similarity scores.**
- **Recommending top 5 predictions based on sentence similarity.**

Implementation

Data Requirement :

Dataset is provided in kaggle, via source link :
https://www.kaggle.com/c/quora-question-pairs/data

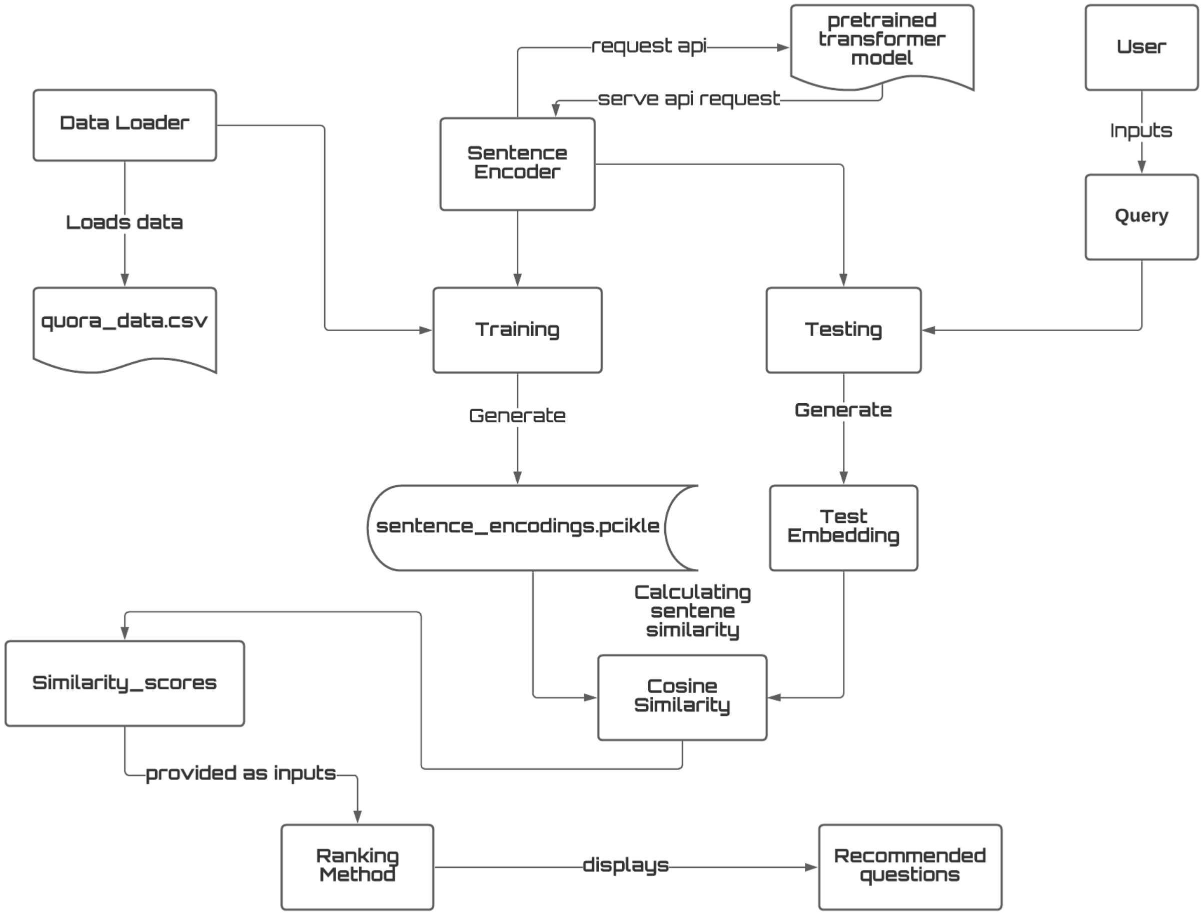
- The shape of the dataset is **(404290, 6)**.
- The dataset consists of **404290 rows and 6 columns (id, qid1, qid2, question1, question2, is_duplicate).**
- **id** column indicates the unique entry whose value is unique in every row of the dataset.
- **id1 and id2** are the unique ids of questions in **question1 and question2** respectively.
- **is_duplicate is 1** if the **two questions have similar intent** and **0 if they don't have a similar intent.**

Data Preprocessing For Selecting Questionnaire Data

- There are **290456 unique questions in question1** column in our datasets.
- There are **299174 unique questions in question2** column in our datasets.
- After combining, there are **589630 total unique questions** in the datasets.
- After preprocessing, the **shape** of the dataset is **(589630, 2)**.
- The **two features** after preprocessing are **Questions and id.**

Methodology

System WorkFlow Chart



1. Data Loader :

- The preprocessed data is available in **quora_data.csv.**
- The **csv data file is read and loaded using the data loader.**
- In our project, this functionality is obtained by **data_loader.py** file.

2. Sentence Encoder:

- Sentence encoder is a model that is used to convert the given input sentence into a 768 dimensional column vector.
- It does so by making an api request to the pretrained transformer model.
- The name of the pretrained transformer model used in this project is bert model.

3. Training

- The project has two main phases. One is Training and another is Testing.
- In training, each sentence in the csv file is sent through a sentence encoder to get a 768 dimensional vector for each question.
- Here, in this project, we have used only 500 rows out of the whole csv file for minimizing the cost and speed of training.
- After training, the embeddings along with their respective question id is stored in a dictionary.
- Then, the dictionary is dumped into a pickle file as a reference so that we can use it during the inference or testing process.

4. Testing

- During testing, the user is allowed to ask any question.
- The asked questionnaire is passed through the sentence encoder to generate 768 dimensional column vectors called as sentence embeddings.
- Then, we are reading to calculate the sentence similarity score by using cosine similarity algorithm.

5. Cosine Similarity

- Cosine similarity measures the similarity between two vectors of an inner product space.
- One vector is the vector embedding generated for the user input query while the others are each vector generated for the training datasets.
- It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction.
- It is often used to measure sentence similarity in text analysis.
- The higher similarity, the lower distances.
- The sentences or questionnaires with lower distances are said to have the similar meanings or intent.

6. Ranking Method:

- This is the final step of our project lifecycle.

- After applying cosine similarity, we get a similarity score of the query set with other all available questionnaires in the corpus.
- Then, we sort the questionnaires in the training corpus according to the similarity score in descending order.
- The higher the similarity score, the higher they are ranked.
- Finally, a table is constructed and displayed to show the recommended questions which can have similar meanings and intent with the query set from the available training datasets.

Learnings And Journals

- There are different pretrained models that we can apply to create sentence embedding for ranking sentences with similar intents.
- However, among all other models, the pretrained bert model is the best model that we can use to get our best results.
- Other models like Word2Vec, Tfidf, CountVectorizer are used to generate word and sentence embeddings but these models are not able to understand the context of the sentence.
- Transformers are the state of the art algorithms when it comes to performing the natural language tasks.
- Transformers can solve sequence-to-sequence tasks and handle long-range dependencies with ease.
- The idea behind Transformer is **to handle the dependencies between input and output with attention and recurrence completely**
- During ranking, we have adopted an approach in which it loops over each sentence/question in the corpus and generates the vector embeddings for each of them.
- Then, we calculate cosine similarity of input query with all available data in the corpus due to which it can cost a huge amount of time if the dataset is in millions.
- So, for solving such issues, we can use advanced ranking algorithms to solve this case.
- Facebook uses EdgeRank algorithm to determine what articles should be displayed in a user's News Feed.