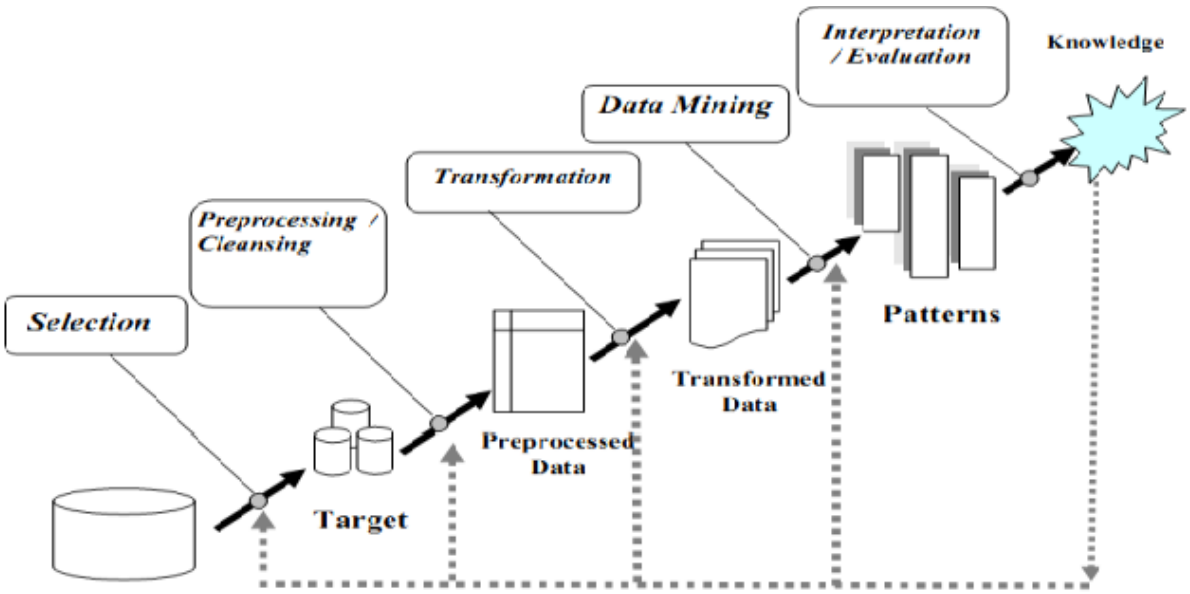
Project KDD and Feature Engineering

Project Team Members- Avinash Ramesh, Charu Cheema, Cory Randolph
Project Team Name- Insight Finders

CMPE 256 - Fall, 2021
11/12/2021

Knowledge Discovery for Data Mining (KDD)

Knowledge Discovery for Data Mining (KDD) is an iterative process for selecting and analyzing data so that useful patterns can be found that add business value. As part of our question answering chatbot/system we have applied the KDD process to ensure repeatability of data cleaning while also comparing various options and the quality of the results achieved.



Steps of the KDD Process (Fayyad et al, 1996)

1. Data Selection

Since we were interested in creating a question answering chatbot/system that would be able to seek out answers from a very wide range of topics, we decided to use the contents of Wikipedia. Only the relevant wiki page contents of top-N pages are gathered in the data selection phrase based on the user's query. The reason we designed our data selection process to have multiple sources (top few pages from wiki) was to provide the best chance at locating the most relevant context for a user's query. To implement this, we'll build a phrase matcher to extract matching context from each wiki page based on the user query.

2. Data Cleansing

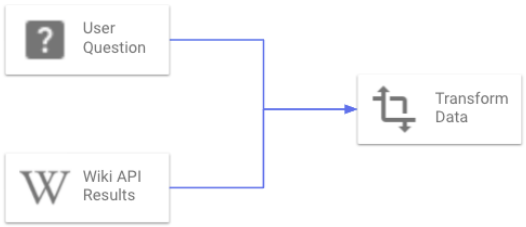
In early iterations of our model we tried passing in the raw query and context into our system, but the noise led to less accurate results. To increase model accuracy and reduce background noise, data cleansing will be performed on the retrieved Wikipedia pages to remove punctuation and special characters, as well as on the user's queries. From this point we applied typical NLP techniques like stop word removal, lemmatization, and word embedding techniques like TF-IDF and word2vec to tokenize and understand the gathered data.

3. Data Transformation

We parse the entire wiki page content, turn it into individual paragraphs, and save the results in a csv file in data transformation. Next, on the user question and extracted paragraph contexts, we utilize sentence embedding techniques like Universal Sentence Encoder/Sentence Transformers to understand semantics of the context and compute cosine similarity. This encoding is the primary transformation from text data to numerical values that can be used as inputs into a machine learning model. Lastly , we filter out only the top-N relevant contexts and store the final results in the required format.

4. Data Integration

There are two primary sources of our data four our system: the users question/query and the context gathered from wikipedia. Once the user's question/query is received and transformed it is passed into the Wiki API to search out the context. Then this context is retrieved and parsed so that everything can be passed into the pretrained NLP model at the same time.



Data Integration Diagram

5. Data Mining

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique developed by Google for natural language processing (NLP) pre-training. For our use case, we will primarily rely on the RoBERTa model, which has been pre-trained on the SQUAD2.0 (The Stanford Question Answering Dataset) dataset. We send all of the TopN relevant context to the model along with the user query (Results of Data Integration section). The model is designed to predict the answers to the user's query based on the context provided. Once the predictions are in place, we list only the TopN relevant (probably 3-5) predictions for the user query based on the model scoring parameter.

6. Pattern evaluation

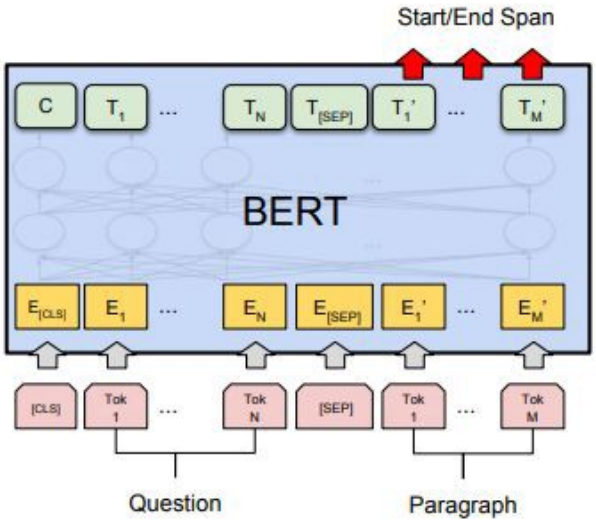
We evaluated the predictions of the RoBERTa model with our use case. The predictions were relevant and per user requirements. The majority of the complex pattern evaluation involved with the context semantic extraction takes place within the RoBERTa model based on the pre-trained model, the benefit is that these complex patterns are easily applied to the question/query and the context our system has provided.

7. Knowledge Presentation

The end result of our project is a simple User Interface where a question can be typed in and then relevant answers along with matching context/wikipedia page url are displayed.

Feature Engineering

One of the most important steps in machine learning is feature engineering. It is the process of using domain knowledge of data to create features that allow machine learning algorithms to function. In our use case, if we can use contexts as features and feed them to our model, so that the model will be able to better understand the sentence. The number of words, number of capital words, number of punctuation, number of unique words, number of stopwords, average sentence length, and so on are some of the common features that we can extract from a sentence. See the below BERT diagram for how text is engineered into numerical vectors form Bert to the Rescue [https://towardsdatascience.com/bert-to-the-rescue-17671379687f]



(c) Question Answering Tasks: SQuAD v1.1

In our problem statement, we transform each wiki page based on the user query into multiple paragraphs, and each paragraph, along with the user query, is embedded using sentence embedding techniques to understand the semantics of the context. Following that, each context is assigned a similarity score based on the user question prior feeding to the ML model.

References

- https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2749028.pdf
- https://chatbotslife.com/chatbot-qa-implementing-chatbot-solution-the-chatc-group-49acef43aaa9
- https://medium.com/@christophberns/using-crisp-dm-to-predict-car-prices-f15eb5b14025
- http://web.stanford.edu/class/cs224n/project/default-final-project-handout.pdf
- https://web.cse.ohio-state.edu/~bair.41/616/Project/Example_Document/Req_Doc_Example.html
- https://towardsdatascience.com/bert-to-the-rescue-17671379687f