Quote Attribution Using Deep Learning

Sayan Das (RA1411003010485)
Shubham Agrawal (RA1411003010458)

What is Quote Attribution?

- Given a literary text as the form of novel or television or movie screenplay, our task is to classify the dialog to their correct speaker.

- This can be done by understanding the context of the dialog e.g. actions, text surrounding it. Along with extracting the features and how he/she speaks.

Dataset

- Dataset has been collected from http://imsdb.com for Futurama television series.

- Dedicated scrapers has been written using python which scrapes the data and store in the file of raw text files.

- These raw text files has been splitter into 70:30 ratio of training and test files respectively.

Cleaning and Preprocessing

- All the dialogues have been cleaned by using the following regular expressions:
    - /(\.\.+)|”|:|;/g: Removes the special characters and replaces with single space
    - /\s*\([^)]*\)/g: Removes the content present in brackets.
    - /\s+/g: Removes multiple spaces and replaces with single space.
- All the dialogues has been tokenised using nltk.word_tokenise package.

Word Embedding

- Word embedding is a parametrised function mapping of some language to high-dimensional vector.
- Example: W(‘cat’) = [0.2 -0.3 0.4 …]
- This useful in extracting the sense of the word in quantitive way.
- For this we have used pre-trained datasets by Stanford having 6Billion tokens with 50 dimension each.

Quote Vector

- To reduce the dimension of dense vectors formed using the word embedding over a particular dialogue we have 2 methods to achieve this:

  - Simple Averaging: Takes the mean of the dense vector vertically.

  - Sentence Level Embedding: Tries to infer pattern from the word vectors this can be done by using dynamic recurrent neural networks with LSTM cell.

Machine Learning Models

n-Way Multi Layered Perceptron Model

Given a quote q consists of $\{w_1, w_2, w_3 \ldots w_n\}$ word vectors and the number of words (n) in each quote varies. Each word vector $w_i$can be represented by $\{f_1^i, f_2^i, f_3^i \ldots f_n^i\}$. here in this case the dimension of the quote is of $n \times d$, to reduce the dimension we have taken the summary of each quote and thus reducing the dimension to d.

Drawbacks

- This model focuses on specifically the features of the dialogue spoken.
- On the basis of single dialogue it tries to assign the speaker. Which leads to low accuracy.
- If the quote vector dimension is smaller than the number of characters present then, the model will never stabilise.

Recurrent Neural Network Model

- Recurrent Neural Network has been used having LSTM cells has been used. GRU Cells are used to extract the features of the statement and pass to the other nodes.

- This is many-to-many type network e.g. for every quote vector as input there is one speaker as output.

Drawbacks

- Only the features are extracted and aware of the adjacent speakers only.
- If the character is not in the context then it shows high error.

Deep Recurrent Neural Network Model

- This is multilayered Recurrent Neural Network. Every layer has it’s own purpose.
- Layer 1: Provides the sentence level embedding to generate quote vector for a single node.
- Layer 2: This layers captures the context of the previous sentence. This is remembers the vital features and forgets the irrelevant features from the previously passed signals. LSTM cells are used to serve the purpose.
- Layer 3: Till now our model is contextually aware but it cannot characterise by sentence feature. GRU cells are used to serve the purpose.

Drawbacks

- Because of multiple and too many weight adjustments it take more time to train.
- Too many hyper parameters to tune manually.

Future Works

- Bi-Directional RNN’s can be introduced which can capture the context of above dialogues but also from the dialogues below it.

- High Dimensional word vectors which provide informations to the models.

- Use of GPU instances for faster training.