SMAI PROJECT / SQUAD_7/Team-37

NAME	ROLL NO	CONTRIBUTION
ANVITA REDDY	2019115009	Sentence Level Model Implementation, Attempting Review Level, Embedding
SWAMY NAIDU	2019115007	Text Preprocessing, Attempting Review Level, High level design
STELLA SRAVANTHI	2019101101	Simple RNN baseline model creation, embeddings
DEEKSHITH AKULA	2019112022	Simple RNN baseline model creation, debugging

TITLE: Hierarchical Model of Reviews for ABS

- DATASET
    - Taken from https://github.com/magizbox/semeval-2016-task-5/blob/eda/data/ABSA16_Restaurants_Train_SB1_v2.xml
    - A XML which is converted in to pandas for making feature extraction simpler
    - Main Attributes:
        - **REVIEW**
            - contains many sentences and each sentence has aspect and polarity
        - **ASPECT**
            - Defined for each sentence
            - contains entity and a attribute.
            - Eg: FOOD#QUALITY
            - can be found in the category attribute of a sentence

- **TEXT**
    - A sentence which is present in a review

- **POLARITY**
    - Defined for a sentence
    - positive(1), negative (0)

- PREPROCESSING:

DATA EXTRACTION

- We extract data from the XML file using element tree
- Required items are extracted and stored in arrays. Sentences have more than one aspects are repeated along with their other aspects
- We created a pandas data frame from better application of preprocessing methods out of these arrays

TEXT CLEANING

- We used regular expressions for removing emotes, links, punctuations from all the sentences
- Using NLTK we removed all the stop words such as "of", "for", "in", "the" etc.

PADDING and TOKENISATION

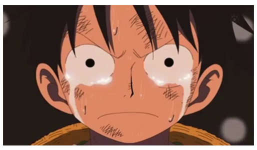
- The maximum number of words in sentence are 35
- Created a corpus for every unique word present in all the sentences
- We tokenized total words on a huge word index so that no word misses out
- After tokening we padded the text with zeros in the end so that every sentence contains 35 words
- Problems in padding Reviews:
    - Different review has different number of sentences
    - We padded maximum count of sentences (44) to all the reviews by using a polarity of 0.5 for each sentence
    - But due to low accuracy, we designed in a way so that each review can have different number of sentences but normalized this in the sentence level LSTM
- Now we use Glove word embeddings for further preprocessing

EMBEDDINGS

- We used Glove 27B , 100 dimensional pre-trained word embeddings

  - Use the glove model to create word embedding vectors for every word in the corpus

- MODEL
  - Contains a embedding layer and a Bi-LSTM layer for sentence level architecture
  - This Sentence Level Bi-directional LSTM captures meaning from both the chronological orders at a time-stamp
  - The output h_t at a given time step is then the concatenation of the corresponding states of the forward and backward LSTM.
  - For each sentence the last state of the Bi-LSTM is taken and padded to a vector of $1 \times$M where M represents the maximum number of sentences present in a review
  - Each h_t state of each sentence is concatenated with aspect vector $a = \frac{1}{2}(x_e + x_a)$ where x_e, x_a represents the 15-dimensional word embedding vectors of entity and review of an aspect
  - This new concatenated vectors are passed on to a review level LSTM which is trained based on average values of all the sentences in that review.
  - The parameters used for the layers are mentioned in the paper.
  - We are done with training, let us test our data set
  - Now the test-data set is evaluated with the help of the predictions from the model and below is the accuracy score.
- What Happened (some backlogs too)



  - Everything went well exactly as described in the paper until review level.
  - At review level architecture, we faced a serious problem of lack of literature.
  - We were able to create a word embedding aspect vector $a = \frac{1}{2}(x_e + x_a)$ for all the sentences and successfully concatenated them with h_l output of previous Bi-LSTM layer.
  - We created a second Bi-LSTM model based on the output of the first model making use of the embedding layer.
  - We faced issues in fitting the data of the model due to some issues which happened because of the inadequacy of the paper.
  - We assigned review level polarities on the basis of majority sentences' polarity and used them to compare the accuracy predicted by both the

models (Simple RNN and Bi-LSTM)

- RESULTS
    - Simple RNN has a validation accuracy of 0.45 on an average.
    - Sentence level Bi-LSTM has an validation accuracy of 0.84 on an average
    - While normal LSTM has an validation accuracy of 0.6 on an average