README: NLP_FINAL_PROJECT

Group-15

Aishwarya Kumar	Aarish Chhabra	Gandharv Mohan	Abhishek Rajgaria
2017011	**2017212**	**2017232**	**2017276**

Dataset

MUStARD dataset, which contains test conversation of popular TV shows like Friends, Big Bang Theory etc.
https://github.com/soujanyaporia/MUStARD/blob/master/data/sarcasm_data.json

A balanced dataset of 690 samples (samples which are present in json format and is labelled).

Each sample contains: utterance text, speaker name, context text, context speakers name, show, label.

Glove Embeddings used for word level representations:
http://nlp.stanford.edu/data/glove.840B.300d.zip

Google Drive Link for the trained models, code files and dataset files:
Google Drive Folder

Models

The trained models are divided into 4 folders:

1. Word ngram models
2. Vectorize model
3. LSTM-Based models
4. Attention models

Within each type there are several models with different variations of the data. Total of 30 varying models have been picked.

Code Files

Word ngram

1. Input: 'sarcasm_data.json'
2. The configuration to be considered (Eg: utterance + context + show) can be specified by changing the value boolean variables: b_utterance, b_context, etc. in the file
3. Converts the data into feature vectors (word ngram for utterance and context and one hot encoding for the rest)
4. It then uses SVM and stratified k-fold cross validation on the feature vectors for classification and scoring.

Vectorize

1. Input: 'sarcasm_data.json'
2. Speaker and context features were taken as in the word ngram approach above with the difference that instead of taking ngrams of utterance and context, utterance and context sentences were vectorized using sent2vec.
3. Model trained on only utterances features gave the best result, SVM and stratified K-fold cross validation was used. This model was saved in the 'Vectorize_utterance_model.sav' file.
4. Model trained on utterance + context features produced poorer results as compared to the only utterance counterpart.

LSTM

1. Inputs - Glove Embeddings pre-trained text file, Sarcasm.json
2. A number of variations were done using this architecture, majorly focusing on finding the optimal amount of context to be included, the different representations of the context, the manner in which speakers and context speakers must be included, the size required to encode the information.
3. Out of all these models, best 6 models have been saved in drive which include Utterances + speaker + context based models & Utterance + speaker + context + context speaker based models. All other variations performed quite less in comparison to these, so those haven't been saved.
4. Description of Saved Models: (These models are saved in LSTM-Based folder inside Trained Models)
    a. Utterance + Context + Speaker
        i. Model1.pb - Using context of length 50 with pre-padding
        ii. Model2.pb - Using context of length50 with post-padding (finalised this)
        iii. Model3.pb - Using total context (length 100)
    b. Utterance + Context + Speaker + Context Speaker
        i. Model4.pb - Using Hidden vector size 64 (LSTM)
        ii. Model5.pb - Using Hidden vector size 128 (LSTM)
        iii. Model6.pb - Combining Speaker & Context Speaker

Attention

1. It takes in three inputs: Sarcasm.json, glove_dictionary and args.
2. Args: It used to specify the combination of the attributes from the sample to take into consideration.
3. Each sample attribute is first converted into embedded vectors.
4. Utterance and context text using glove embedding and the rest via one hot encoding.
5. Utterence_text (ut), Speaker (sp), Context text (ct), Context speaker (cp), Show (sh).
6. '-' has been used to show combinations.

Evaluation Metric Used:

1. Accuracy
2. Precision
3. Recall
4. F1_score