COMP4332/RMBI4310 Project 1 – Sentiment Analysis Report

CHENG, Ho Sing (20513950), LIU, Yat Long (20520953), WAN, Kwok Kit (20519679) and
YEUNG, Man Yin Michael (20603418) [Project Group 26, 2021 Spring Term, HKUST]

I. Abstract

Sentiment analysis is implemented on a text reviews dataset with stars rating (1-5) labels. Explorative data analysis is first performed on features given in and derived from the columns. Then the TFIDF-BiLSTM-Numerical-MLP Model built in this project, in which vectorized TF-IDF & BiLSTM of text data and stacked numerical features selected in explorative data analysis are concatenated to train MLP model, outputs predictions with 63% valid accuracy.

II. Explorative Data Analysis

(Refer to: explorative_data_analysis.ipynb)

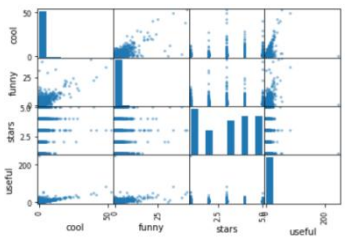
Exploration on the Given Columns

Besides the text reviews (*'text' column)* and the stars ratings (*'stars' column*), some other columns are explored in this following analysis. We do not include columns containing IDs in our analysis as they do not convey useful information. As we can see in the third row of the correlation matrix, the three feature columns do not have a high correlation with 'stars'. Instead, we observe that 'funny' and 'cool' has a high correlation. We also visualize the pair relationships in the scatter matrix.

Correlation and Scatter Matrices

	cool	funny	stars	useful
cool	1.000000	0.735803	0.087558	0.542500
funny	0.735803	1.000000	-0.061346	0.507746
stars	0.087558	-0.061346	1.000000	-0.085296
useful	0.542500	0.507746	-0.085296	1.000000

Correlation between different columns. The third row shows correlation with the 'stars' label.



This figure visualizes the relationship between different columns. Observe that 'funny' and 'cool' has a strong correlation.

Derived 'text_len' & 'day_of_week' Columns

The *'text_len' column* contains the number of characters in reviews. Number of tokens was once considered instead but showing worsened validation accuracy thus abandoned. The *'day_of_week'* column contains weekday[1].

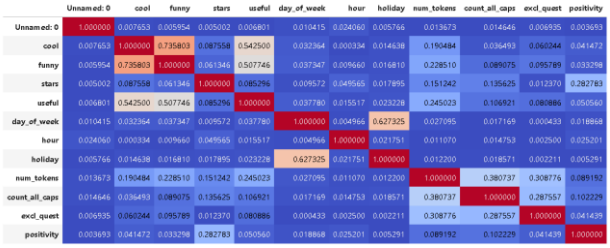
Other Columns Abandoned

(Refer to: abandoned_variation.ipynb)

(i) *Derived 'excl_quest'* [2] *column*:
Median is 0 or 1 when grouped by 'stars'.

(ii) *'cool'; and derived 'count_all_caps'* [3], *'holiday'*[4], *'hour'*[5], and *'positivity'*[6]*:*
Trials of training show reduction or no improvement in validation accuracy.

Correlations Including All Columns



	Unnamed: 0	cool	funny	stars	useful	day_of_week	hour	holiday	num_tokens	count_all_caps	excl_quest	positivity
Unnamed: 0	1.000000	0.007653	0.005954	0.005002	0.006801	0.010415	0.024060	0.005766	0.013673	0.014646	0.006935	0.003693
cool	0.007653	1.000000	0.735803	0.087558	0.542500	0.032364	0.000334	0.014638	0.190484	0.036493	0.060244	0.041472
funny	0.005954	0.735803	1.000000	0.061346	0.507746	0.037347	0.009660	0.016810	0.228510	0.089075	0.095789	0.033298
stars	0.005002	0.087558	0.061346	1.000000	0.085296	0.009572	0.049565	0.017895	0.151242	0.135625	0.012370	0.282783
useful	0.006801	0.542500	0.507746	0.085296	1.000000	0.037780	0.015517	0.023228	0.245023	0.106921	0.080886	0.050560
day_of_week	0.010415	0.032364	0.037347	0.009572	0.037780	1.000000	0.004966	0.627325	0.027095	0.017169	0.000433	0.018868
hour	0.024060	0.000334	0.009660	0.049565	0.015517	0.004966	1.000000	0.021751	0.011070	0.014753	0.002500	0.025201
holiday	0.005766	0.014638	0.016810	0.017895	0.023228	0.627325	0.021751	1.000000	0.012200	0.018571	0.002211	0.005291
num_tokens	0.013673	0.190484	0.228510	0.151242	0.245023	0.027095	0.011070	0.012200	1.000000	0.380737	0.308776	0.089192
count_all_caps	0.014646	0.036493	0.089075	0.135625	0.106921	0.017169	0.014753	0.018571	0.380737	1.000000	0.287557	0.102229
excl_quest	0.006935	0.060244	0.095789	0.012370	0.080886	0.000433	0.002500	0.002211	0.308776	0.287557	1.000000	0.041439
positivity	0.003693	0.041472	0.033298	0.282783	0.050560	0.018868	0.025201	0.005291	0.089192	0.102229	0.041439	1.000000

Observe the fourth row containing correlations of columns with stars.

Selection of Columns in the Finalized Model

All columns showing sufficient correlations to the "stars" label is at first used to train the model to be introduced later. Some columns shows correlations but are then abandoned when attempting to enhance the model. Finally only four columns, *'funny', 'useful', 'text_len' and 'day_of_week'*, remains in the model as **numerical features** after multiple trials of enhancement for better validation accuracy.

---

[1] E.g., '0' represents Monday and '1' represents Tuesday, etc.
[2] Total frequency of "!" and "?" used in the text reviews.
[3] Frequency of "all-caps" words used, e.g., APPLE
[4] 1 (True) for Fed holiday or Sunday, else 0 (False)
[5] Hour in day of review, e.g., 18 for a review in 6:30 p.m.
[6] Defined as sum of scores of the adjectives in 'text'; score is its frequency in 4-5 stars train reviews minus that in 1-2 stars'.

III. Algorithm of Finalized Model

(Refer to: finalized_model.ipynb)

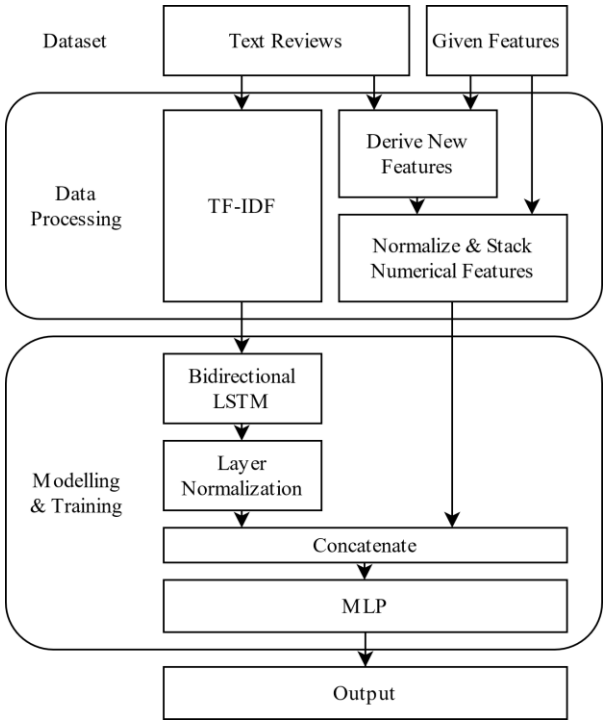
An Overview of the Main Idea

Concatenate (i) TF-IDF & Bi-Directional LSTM on text data, and (ii) derived numerical features to train MLP model accuracy.

(i) **TF-IDF of Text:** Transform the valid and test set using the TF-IDF built by train set.

(ii) **Numerical Features:** Stack normalized numerical features including ***'funny', 'useful', 'text_len' and 'day_of_week'*** using `numpy.hstack()`

Outline of Major Procedures (Flowchart)



Data Processing

1. **TF-IDF Features:** Natural Language Processing of Text, including: tokenizing the text, filtering stopwords, stemming, and n-gram (by TFidfVectorizer).
2. **Numerical features:** Normalization and stacking of all selected numerical features as mentioned previously.
3. One-hot encoding for stars. (`to_categorical`)

Modelling

4. Define Model Architecture:
   (i) Pass TF-IDF Vector into **BiLSTM**;
   (ii) Perform Layer Normalization on the output of **BiLSTM**, and concatenate stacked feature columns to one matrix;
   (iii) Pass the matrix into **Multi-Layers Perceptron (MLP)**.

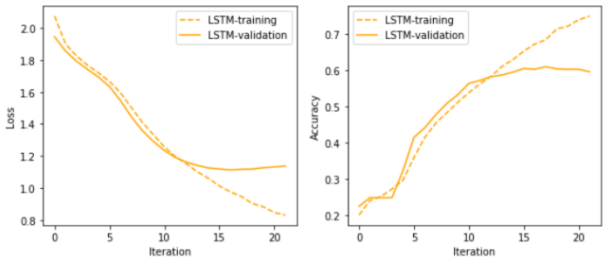
Training

5. **Early-stopping:** Preventing over-fitting Monitor validation loss with patience 5.
6. **Scheduler:** lower down the learning rate Halve the learning rate every 20 epochs.

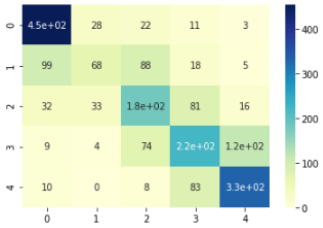
IV. Prediction Results

Training Accuracy: 81.7%

Validation Accuracy: 63% (from *evaluate.py*)



With help of the **confusion matrix** below, we see that the model reaches an accuracy of 63% in the validation set (frequency on diagonal indicates correctly predicted data), yet not predicting well with "stars = 2" in particular.



Predicted labels (stars) against actual labels. Observe that the second row (=2 stars) on the diagonal has a light color and the two neighboring cells have relatively high frequency (99 and 88 to 68).

Classification Report on Validation Data

	precision	recall	f1-score	support
0	0.75	0.88	0.81	517
1	0.51	0.24	0.33	278
2	0.49	0.53	0.51	344
3	0.54	0.52	0.53	427
4	0.70	0.77	0.73	434
accuracy			0.63	2000
macro avg	0.60	0.59	0.58	2000
weighted avg	0.62	0.63	0.61	2000

Shows high accuracy, precision, recall and f1-score on validation data.

END OF REPORT