**ABSTRACT**

In this project we implemented 3 types of models (SVM, XGBoost, LSTM) for the fine-grained sentiment analysis of user reviews about restaurants in Chinese language. There are 20 elements, and 4 labels (positive, neutral, negative, not mentioned) for each of them. We trained one model of each type for each of the elements. On the whole, XGBoost has the best performance based on the accuracies, weighted f1 scores, and efficiencies.

**DATASET**

We used the datasets provided by AI challenger official. The training and validation data are manually labelled. We created validation and test sets from the original validation set. Training set (105,000), validation set (7500), and test set (7498) have roughly same class distributions. For the limited space, we just list the 6 categories for the all 20 elements:

1. Location
2. Service
3. Price
4. Environment
5. Dish
6. Others

Within “Others”, the 2 elements are “overall experience” and “willing to consume again”.

**RESULTS AND DISCUSSIONS**

We used accuracies and weighted F1 scores as our metrics. The graph shows a very fluctuating test accuracies across 20 elements. In general, the XGBoost models have better results.

**LSTM FOR CLASSIFICATION**

We constructed a two-layer LSTM (Long Short Term Memory) neural network. This model takes the 105,000 outputs of size (1, 400) from feature extraction step as the input. The embedding matrix takes weights from the Word2Vec model, and is not trainable. Apart from hyperparameters shown in the graph, we also modified class weights according to class distributions in each element. We used categorical cross entropy as the loss function:

\[ L(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{4} y_{ij} \log(p_{ij}) \]

**SVM & XGBOOST FOR CLASSIFICATION**

**SVM**

We first extract the embedding vectors for each word in a sentence through indices, to form a sentence matrix. Then we created sentence feature vectors for training, validation, and test sets by averaging the sentence matrix along the vertical axis to get a vector. Each observation is a vector of size (1, 300). We tuned gamma, learning rates, kernel type through cross validation, and used linear SVC for our final model.

**XGBOOST**

The inputs are the same with SVM. We used GridSearchCV to tune for parameters including learning rate and max depth.

**PREPROCESSING & FEATURE EXTRACTION**

We constructed a Word2Vec model using gensim library. The corpus is from the original training, validation, and test datasets. The embedding size is 300, and the words with frequency < 3 are eliminated. The vocabulary length is 6573.

We transformed each sentence into a vector of indices, which map the words to this vocabulary, and further to the embedding matrix. We padded each sentence to a length of 350 for SVM and XGBoost, and 400 for LSTM.

**FUTURE WORKS**

1. Collect better-labelled data;
2. Improve the quality of language representation models; Implement contextual representation, e.g. BERT;
3. Apply Attention mechanism for long input texts.

**REFERENCES**

https://github.com/AIChallenger/AI_Challenger_2018