Jeffrey Chen
Zhansaya Abdikarimova

Final Project Report

Introduction

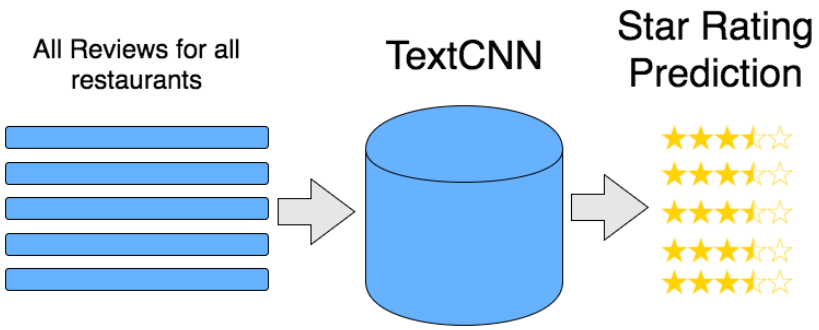
Enormous amounts of data are generated and collected in all shapes and sizes. Nowadays people more and more rely on reviews and ratings before using any product or going some places. User-submitted text reviews are among the most difficult to interpret due to hidden contexts and intentions. Any single given review may contain a number of opinions on various aspects of a product or service. Programmatically extracting the overall sentiment of a review would allow each piece to be more easily quantified in an efficient manner without the need of human examination. This will be incredibly helpful to consumers who seek to find the best product or service with minimal searching on their own part. Currently, overall product ratings are a simple average of all of the user-submitted ratings. By developing an algorithm that can determine the underlying attitude of a text review overall ratings can become a more accurate representation than a simple average.

Jeffrey Chen
Zhansaya Abdikarimova

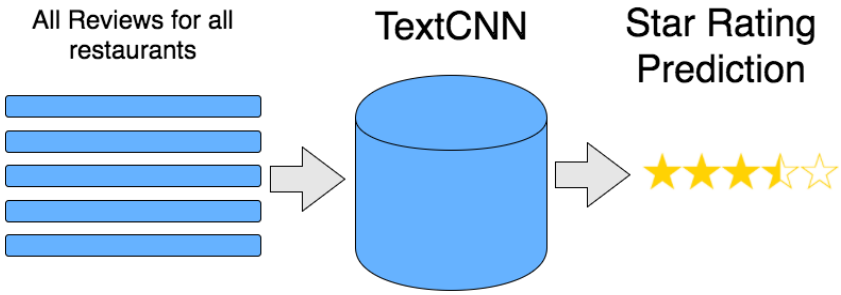
Goal Description

The ultimate goal of this project is to build a model that predicts the overall rating for a restaurant given a set of features. We have a model that predicts a specific star rating and compares it to the true rating in three different experiments:

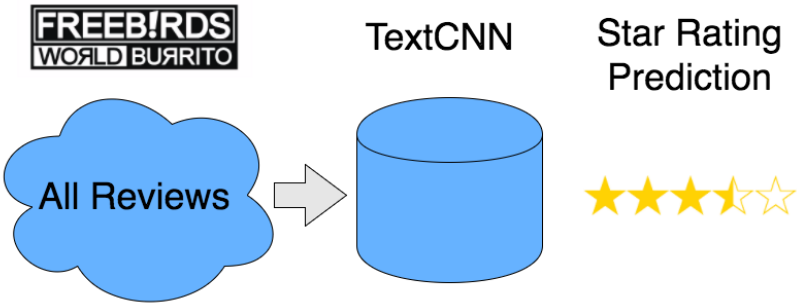
A. Predict rating of each individual review.



B. For a given restaurant, predict the ratings for each of its reviews and average them to get a predicted overall rating. Then, we record the difference of this rating from the true overall rating.



C. Similar to experiment B, except every review for each business is combined into one single review and then fed into the CNN for training. The output is predicted overall rating rounded to the nearest half star. No averaging necessary.



Jeffrey Chen
Zhansaya Abdikarimova

Dataset Cleaning and Preprocessing

Yelp has been holding a data mining challenge where they have provided datasets on businesses, reviews, users, check-ins, and tips. We are only interested in features of the first two datasets for now: businesses and user reviews. Each dataset is nicely formatted in JSON, rich in features, and composed of thousands of records. This saved us the trouble of having to do extensive cleaning and sorting, allowing more time to work on the algorithm instead.

The features from the business dataset that we are interested in include:

- Unique business ID code
- Attributes: We will only use records that have the "restaurant" attribute
- Overall star rating: From 1 to 5, in half-star increments

The features from the reviews dataset that we are interested in include:

- Business ID code with which this review is associated
- Rating given by this review: From 1 to 5, in whole-star increments
- Text content of the review: a single string

We clean the data in the following way:

1. Scan through business JSON file for business IDs with a "restaurant" category.
    a. 25,071 restaurants out of 77,445 businesses
2. From review JSON file, extract reviews of restaurants.
    a. 1,363,242 reviews out of 2,225,213 total
3. Delete unused features to save space and time when importing.
4. Export to new JSON files.

Preprocessing for Experiment A:

1. Read in all restaurant reviews from JSON file.
2. Tokenize each review using Regular Expression operations.
3. Pad each review to the length of the longest review using <pad> tokens.
4. Create vocabulary index that maps each word to an integer index from 0 to the vocabulary size.
5. Split list of reviews into training, validation, and testing sets.

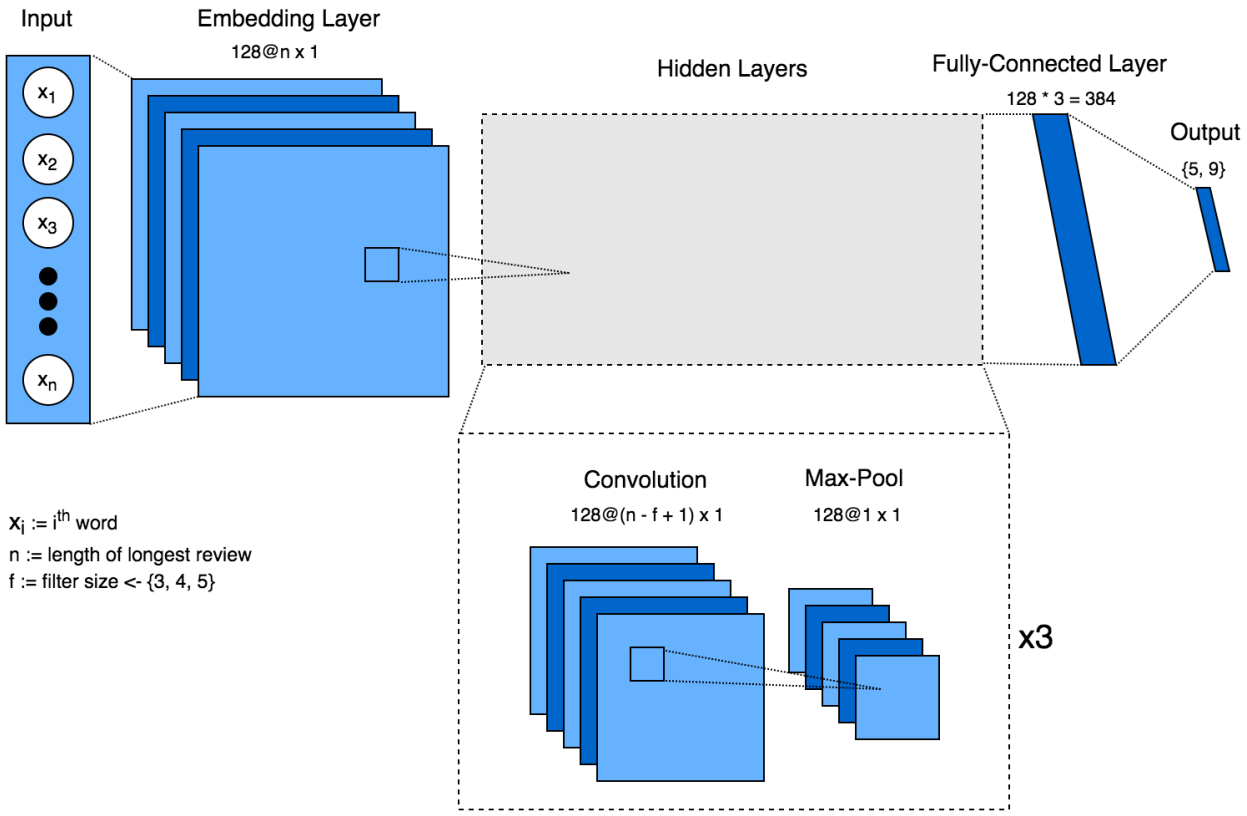
Preprocessing for Experiment B and C:

1. Read in business IDs instead of individual reviews.
2. Split list of business IDs into training, validation, and testing sets.
3. For each business ID, retrieve all corresponding reviews.
    - Exp B: Store all reviews in one list.
    - Exp C: Combine all reviews into one review.

Jeffrey Chen
Zhansaya Abdikarimova

Model Description

Since we have been mostly focused on Convolutional Neural Networks in class, we decided to use it as well for our task.



1. Input: One review, where each *x* is a tokenized word. All reviews are padded to be of length *n*.
2. Embedding: The word in the input vector is replaced by its index in the vocabulary vector. A vector of indices is output from this layer.
3. There are a series of hidden layers with three different filter sizes. The convolution layer and max-pooling layer are repeated three times each.
    a. Convolution: A filter is passed over the vector of indices to perform activations on neurons.
    b. Max-Pool: The activations are downsampled.
4. Fully-connected layer: The output from the hidden layers is flattened and dropout is performed.
5. Output: L2 regularization loss and softmax cross entropy calculations are performed on the flattened activations to produce an output size of 5 or 9, depending on the experiment.

Jeffrey Chen
Zhansaya Abdikarimova

Implementation

The two main technologies we used for our implementation are Python and the TensorFlow library. We created a TextCNN class that takes hyper parameters as values for its constructor. Such parameters include:

Parameter	Description
Review Length	Length of longest review or sentence
Number of Classes	5 stars (or 9 for overall)
Vocabulary Size	Number of unique words
Embedding Size	Dimensions of Weight Vectors
Filter Sizes	Convolutional Layer Filter Size {3, 4, 5}
Number of Filters	128
L2 Regularization	0.5
Dropout	0.5
Learning Rate	1e-3
Batch Size	100 reviews or businesses

Jeffrey Chen
Zhansaya Abdikarimova

Results

Set	Train	Validation	Test
A	~70%	~55%	~55%
B	~70%	~55%	~55% (Off by ~0.25 stars)
C	~60%	~60%	~60%

Team Member Contributions

Zhansaya	Jeffrey
• Data Processing • Neural Network building • Experiment A • Experiment C	• Data Processing • Experiment B • Architecture diagram

Challenges & Future Improvements

One of our initial ideas that we mentioned in our proposal was to use multiple features to have a better prediction accuracies. For example, along with text reviews we wanted to analyse the influence of the location to the star rating. In order to add it we were thinking of having additional row to our review matrices that would represent the location. However, due to time constraints and not enough evidence that our idea would work we were not able to implement and test it. Also, the data we have is very big so it overall training and trying out different hyperparameters took a lot of time.

References

- ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION
  - https://arxiv.org/pdf/1412.6980.pdf
- Convolutional Neural Networks for Sentence Classification
  - http://arxiv.org/pdf/1408.5882v2.pdf
- UNDERSTANDING CONVOLUTIONAL NEURAL NETWORKS FOR NLP
  - http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/
- IMPLEMENTING A CNN FOR TEXT CLASSIFICATION IN TENSORFLOW
  - http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/