Washington University in St. Louis

OLIN BUSINESS SCHOOL

Deep Learning for Prediction of Business Outcomes

Final Project
Lending Club Loan Default Detection

Project Group 28, Section 22 Group Members:

Fandi Sun, Yujian Chen, Yufan Zhou

May 7, 2024

Abstract

In the digital marketplace, the integrity of online reviews is undermined by the prevalence of fake reviews. This project proposes a hybrid deep learning model combining LSTM and CNN to enhance the detection of fake reviews. Utilizing a dataset with over 40,000 records from Kaggle, the model aims to surpass traditional detection limitations by accurately identifying fraudulent content, targeting an accuracy rate of over 90%. This approach promises to bolster consumer trust and ensure fair business practices by improving the authenticity of online reviews.

1 Introduction

In the world of online commerce, customer reviews play a crucial role in influencing purchasing decisions and establishing the reputation of a business. As such, the authenticity of these reviews is critical to both consumers and businesses. However, in the digital space, there has been a rise in the prevalence of fake reviews - manipulated feedback - that distort consumer perceptions and provide an unfair competitive advantage.

Traditional methods for detecting fake reviews, such as simple algorithmic analysis and manual validation, have proven to be inadequate due to their low accuracy and inability to cope with the exponential growth of online review data. These methods often struggle to distinguish between nuanced real content and fake content, leading to a large number of false positives and underreporting. The complexity of the language used in fake comments, which often mimics the emotions of real users, poses a significant challenge to existing models.

The goal of this project is to develop a system that more reliably detects fake comments, thereby restoring trust in online commenting systems and protecting the integrity of e-commerce interactions. By achieving over 90% accuracy, this project aims to set a new standard for review validation techniques.

2 Literature Review

3 Problem Description

Current methods for identifying fake comments face multiple challenges:

Inaccuracy: Many existing algorithms have difficulty distinguishing between complex human-like text generated by sophisticated methods and genuine comments, resulting in high rates of both false positives and misses.

Scalability: As the volume of online content grows, traditional detection methods, which often involve manual validation or simple automation techniques, are unable to effectively handle the large volumes of data generated every day.

Adaptability: Fake commenting tactics are constantly evolving, using more advanced techniques and nuanced language styles to evade detection. Existing systems are often unable to adapt quickly to these changes and therefore become less effective over time.

In addition, the dataset used to detect these reviews, while covering multiple product categories with over 40,000 entries, presents its own challenges. The diversity in the quality of reviews, the subtlety of fraudulent tactics, and the differences in linguistic expression between categories all add to the complexity of developing an effective detection model.

This project aims to address these issues by developing a robust, scalable and adaptable system capable of accurately identifying fraudulent comments. The system employs a hybrid model that combines the sequential data processing capabilities of LSTM with the pattern recognition benefits of CNN, aiming to significantly improve the detection of fake comments and restore the integrity of online commenting platforms.

4 Database Background and Data Preprocessing

The dataset is from Kaggle, which has 4 columns and over forty thousand rows, and the Ratio of true and false comments is about 50 to 50, containing 20k fake reviews and 20k real product reviews. For the four columns, we have category, rating, label, and reviews columns, and we will mainly focus on label and reviews column. In the label column, there are two unique values, OR and CG. OR = Original reviews (presumably human created and authentic); CG = Computer-generated fake reviews. The computer-generated reviews are generated by GPT 2, and original reviews is obtained from Amazon.

Methods

RNN

A Recurrent Neural Network (RNN) is a deep learning model that is particularly suited for sequence data analysis. In contrast to the unidirectional feedforward neural network, it is a bidirectional artificial neural network, meaning that it allows the output from some nodes to affect subsequent input to the same nodes. This makes them effective for tasks where the context of previous inputs significantly impact current predictions.

- Embedding layer:
  Converts each word in the reviews to a fixed-length word vector. Word vectors can capture the semantic information of words and map it into a continuous vector space. By specifying the max_features parameter, the model can process a predefined number of terms.
- RNN Layer (SimpleRNN):
  The model uses three stacked RNN layers, each containing multiple loop units. A cyclic unit is able to remember information previously input and use it to process the current

input, capturing contextual information and potential patterns in a sequence of text. The final layer of RNN passes the output to the dense layer for final classification.

- Dense layer:
  The final layer uses the sigmoid activation function, which outputs a probability value between 0 and 1 indicating the probability that the comment was generated by computer.
- Execute part：
  The sigmoid activation function is used for binary classification to determine whether a comment is machine-generated. Optimizer is ‘nadam’ that combines the benefits of Adam and NAG: the adaptive learning rate of Adam with the momentum correction of NAG for faster convergence and better generalization performance. The evaluated method is set as ‘accuracy’ which is the percentage of true positive and true negative to the whole result.
- Model limitations:
  In this part, we are facing over-fitting problem, there is a huge gap between the loss and accuracy value of training dataset and testing dataset. We will add penalty part to deal with this problem. Meanwhile, we need continue trying better model to keep improve the accuracy rate.

CNN

Steps is much similar with the RNN model except the RNN layers are change to one dimension CNN and pooling layer.
Combining embeddings and convolutional neural networks (CNNs) is a powerful technique for processing sequential data, especially for tasks in natural language processing. Word embeddings capture semantic information about words by representing them as dense vectors, while CNNs can detect patterns in the sequential data by convolutional filters.

LSTM with CNN

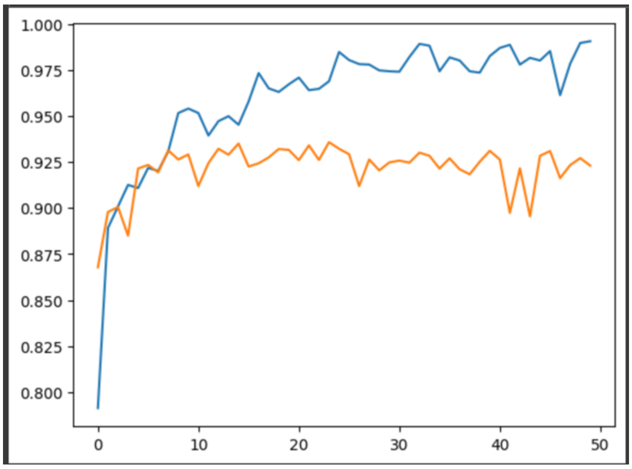
Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to deal with the vanishing gradient problem present in traditional RNNs. Its relative insensitivity to gap length is its advantage over other RNNs.

Bidirectional LSTM with CNN

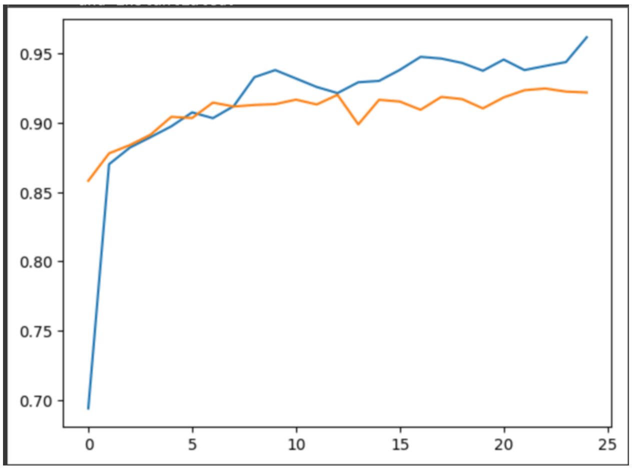
Bi-LSTM (Bidirectional Long Short-Term Memory) is a type of recurrent neural network (RNN) that processes sequential data in both forward and backward directions. Combining a bidirectional Long Short-Term Memory (LSTM) network with a Convolutional Neural Network (CNN) can enhance the model's performance to learn both spatial and temporal dependencies. This combination is useful to understand sequential relationships and identify patterns in the data simultaneously.

Results

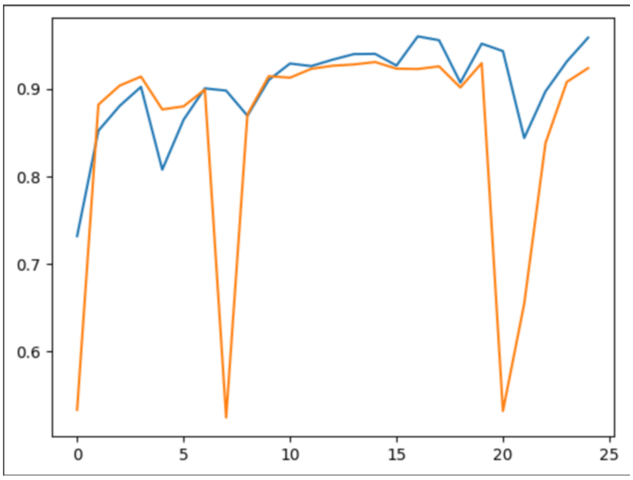
Performance



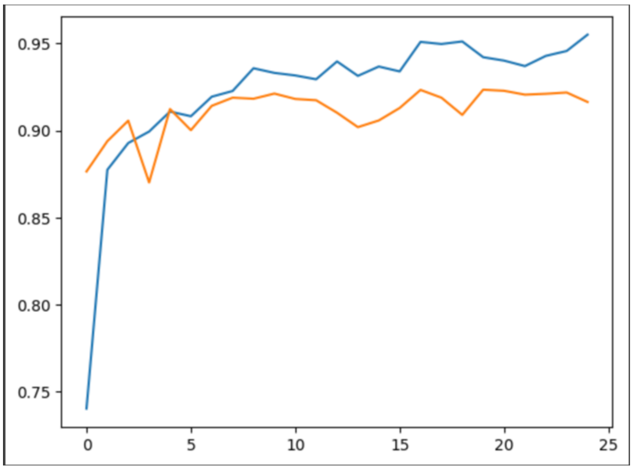
In the RNN model, the accuracy of training dataset is already raised to around 97.5%, the accuracy of testing dataset is stable at around 92.5% and couldn't improve. There is a gap between training and testing dataset, so we are facing overfitting problems, we need to add L2 or/and drop out some neurons from each layer.



Because we add L2 in model, the over-fitting problem has a further solved, the gap still exists but the distance become lower. Due to we just change RNN to CNN, we don't have much improve on model selection, the overall accuracy rate doesn't have increase. The training set accuracy rate still around 95%, and testing set is around 90%.



We combine LSTM with CNN in the model. Meanwhile, we also used dropout and recurrent_dropout function in the part, which we set is 20%. Therefore, the overfitting problem is also not observably. Both testing and training dataset is around or over 90%.



We combine two Bidirectional LSTM with one CNN in the model. We also add dropout and L2 function in the model, so the training and testing dataset accuracy rate are very close. Both testing and training dataset is over 90% and close to 92% and 95%.

Conclusion

In total, we implement 4 types of method to improve our model and try to solve the potential problem we faced especially under-fitting and over-fitting problems. Finally, we control our accuracy of test dataset stabled stay at over 90%. We think it is at a relatively high level. By get more than 90% accuracy in the model, the project provides a powerful solution that has the potential to have a significant impact on the online marketplace. Restoring trust of online reviews platform and enables consumers to make informed decisions. Ultimately, contributing to a more transparent and fairer online ecosystem.

Appendix

Date description:

Variables	**Description**
category	dropped
rating	dropped
label	CG/OR (OR = Original reviews (presumably human created and authentic); CG = Computer-generated fake reviews)
text	User reviews of products (string)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 50, 64)            640000

 simple_rnn (SimpleRNN)      (None, 50, 32)            3104

 simple_rnn_1 (SimpleRNN)    (None, 50, 16)            784

 simple_rnn_2 (SimpleRNN)    (None, 32)                1568

 dense (Dense)               (None, 1)                 33

=================================================================
Total params: 645,489
Trainable params: 645,489
Non-trainable params: 0
_________________________________________________________________
```

RNN model each layer and output shape detail

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 50, 64)            640000

 conv1d_2 (Conv1D)           (None, 46, 64)            20544

 max_pooling1d_1 (MaxPooling  (None, 15, 64)           0
 1D)

 conv1d_3 (Conv1D)           (None, 11, 32)            10272

 global_max_pooling1d_1 (Glo  (None, 32)               0
 balMaxPooling1D)

 dense_1 (Dense)             (None, 1)                 33

=================================================================
Total params: 670,849
Trainable params: 670,849
Non-trainable params: 0
_________________________________________________________________
```

CNN model each layer and output shapes detail

```
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 50, 64)            640000

 conv1d (Conv1D)             (None, 46, 64)            20544

 lstm (LSTM)                 (None, 32)                12416

 dense (Dense)               (None, 1)                 33

=================================================================
Total params: 672,993
Trainable params: 672,993
Non-trainable params: 0
_________________________________________________________________
```

CNN with LSTM each layer and shape detail

```
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 50, 64)            640000

 conv1d_1 (Conv1D)           (None, 46, 64)            20544

 bidirectional_2 (Bidirectio  (None, 46, 64)           24832
 nal)

 bidirectional_3 (Bidirectio  (None, 64)               24832
 nal)

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 710,273
Trainable params: 710,273
Non-trainable params: 0
_________________________________________________________________
```

CNN with two bidirectional each layer and shape detail

Reference

Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon-gyo Jung, Bernard J. Jansen,Creating and detecting fake reviews of online products,Journal of Retailing and Consumer Services, Volume 64,2022,102771,ISSN 0969-6989, https://doi.org/10.1016/j.jretconser.2021.102771 (https://www.sciencedirect.com/science/article/pii/S0969698921003374)

N. A. Patel and R. Patel, "A Survey on Fake Review Detection using Machine Learning Techniques," 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2018, pp. 1-6, doi: 10.1109/CCAA.2018.8777594. keywords: {Feature extraction;Support vector machines;Linguistics;Machine learning;Classification algorithms;Decision trees;Fake Review;Sentiment Analysis;Opinion Spam;Fake review detection technique;Machine learning},