

Comparative Analysis of Machine Learning and Deep Learning Algorithms for Sentiment Polarity Detection.

-Project By

Raveesh R- 21011101099- 21110282 - 3rd year AI&DS-B

Samyuktha S- 21011101106 -21110237 - 3rd year AI&DS-B

Introduction:

Sentiment analysis, or opinion mining, holds significant importance in understanding public sentiment and customer feedback. This research paper concentrates on the task of sentiment polarity detection and conducts a comparative study between conventional machine learning algorithms and cutting-edge deep learning methods. The study leverages the Amazon reviews dataset as its foundational data source. Following text data preprocessing, a range of machine learning techniques and hybrid models are applied to categorize sentiment. Notably, the study highlights the superiority of CNN, achieving an 85% accuracy rate. As part of future research, the paper suggests delving into more advanced deep learning techniques like GRU, Transformers and BERT Model to further enhance predictive accuracy.

Dataset:

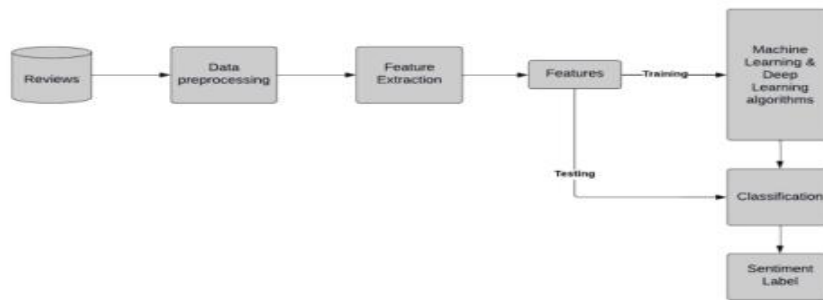
For this research paper the experimental dataset is the amazon reviews dataset, which includes the reviews from Amazon selected from Hugging Face. The dataset contains 36 million reviews out of which 18 million are positive comments and the remaining are negative comments. We will be using a subset of the dataset for our research purposes to reduce the computational complexity.

Link: https://huggingface.co/datasets/amazon_polarity

Dataset Sample:

| label (class label) | title (string) | content (string) |
|---------------------|--|---|
| 1 (positive) | 'Stuning even for the non-gamer' | "This sound track was beautiful! It paints the senery in your mind so well I would reconed it even to people who hate vid. game music! I have played the game Chzono. |
| 1 (positive) | "The best soundtrack ever to anything." | "I'm reading a lot of reviews saying that this is the best 'game soundtrack' and I figured that I'd write a review to disagree a bit. This in my opinio is Yasunori. |
| 1 (positive) | 'Amazing!' | "This soundtrack is my favorite music of all time, hands down. The intense sadness of 'Prisoners of Fate' (which means all the noze if you've played the game) and the. |
| 1 (positive) | 'Excellent Soundtrack' | "I truly like this soundtrack and I enjoy video game music. I have played this game and most of the music on here I enjoy and it's truly relaxing and peaceful.On disk. |
| 1 (positive) | 'Remember, Pull Your Jaw Off The Floor After Hearing It' | "If you've played the game, you know how divine the music is! Every single song tells a story of the game, it's that good! The greatest songs are without a doubt,. |

Proposed Workflow:



Base model and architecture:

This is the base model architecture taken from the research paper containing sets of Conv2D, BatchNormalisation and MaxPooling layers followed by a flattening layer. Accuracy of the base model: 85%

```
def build_model():
    sequences = layers.Input(shape=(MAX_LENGTH,))
    embedded = layers.Embedding(MAX_FEATURES, 64)(sequences)
    x = layers.Conv1D(64, 3, activation='relu')(embedded)
    x = layers.BatchNormalization()(x)
    x = layers.MaxPool1D(3)(x)
    x = layers.Conv1D(64, 5, activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.MaxPool1D(5)(x)
    x = layers.Conv1D(64, 5, activation='relu')(x)
    x = layers.GlobalMaxPool1D()(x)
    x = layers.Flatten()(x)
    x = layers.Dense(100, activation='relu')(x)
    predictions = layers.Dense(1, activation='sigmoid')(x)
    model = models.Model(inputs=sequences, outputs=predictions)
    model.compile(
        optimizer='rmsprop',
        loss='binary_crossentropy',
        metrics=['binary_accuracy']
    )
    return model

model = build_model()
```

Results of the base model:

The study compared machine learning and deep learning models with SVM and CNN achieving the highest accuracy, respectively. Naive Bayes outperformed Random Forest, and hybrid models performed better than BiLSTM. Deep learning showed superior learning capacity, with CNN being the best-performing model overall, with an accuracy of 85%.

Refined model:

We have implemented an improved model using CNN and RNN that has yielded an accuracy of 95%. The architecture of our model is presented below.

The model can be accessed using the link:

https://github.com/raveeshraj कुमार/Amazon_Sentiment_Analysis

CNN architecture used:

Convolutional Neural Net Model

```
def build_model():
    sequences = layers.Input(shape=(MAX_LENGTH,))
    embedded = layers.Embedding(MAX_FEATURES, 64)(sequences)
    x = layers.Conv1D(64, 3, activation='relu')(embedded)
    x = layers.BatchNormalization()(x)
    x = layers.MaxPool1D(3)(x)
    x = layers.Conv1D(64, 5, activation='relu')(x)
    x = layers.BatchNormalization()(x)
    x = layers.MaxPool1D(5)(x)
    x = layers.Conv1D(64, 5, activation='relu')(x)
    x = layers.GlobalMaxPool1D()(x)
    x = layers.Flatten()(x)
    x = layers.Dense(100, activation='relu')(x)
    predictions = layers.Dense(1, activation='sigmoid')(x)
    model = models.Model(inputs=sequences, outputs=predictions)
    model.compile(
        optimizer='rmsprop',
        loss='binary_crossentropy',
        metrics=['binary_accuracy']
    )
    return model

model = build_model()
```

```
preds = model.predict(test_texts)
print('Accuracy score: {:.4}'.format(accuracy_score(test_labels, 1 * (preds > 0.5))))
print('F1 score: {:.4}'.format(f1_score(test_labels, 1 * (preds > 0.5))))
print('ROC AUC score: {:.4}'.format(roc_auc_score(test_labels, preds)))
```

Accuracy score: 0.9451
F1 score: 0.946
ROC AUC score: 0.9868

The CNN model has yielded an accuracy of 94.5%.

RNN architecture used:

Recurrent Neural Net Model

```
: def build_rnn_model():
    sequences = layers.Input(shape=(MAX_LENGTH,))
    embedded = layers.Embedding(MAX_FEATURES, 64)(sequences)
    x = layers.CuDNNGRU(128, return_sequences=True)(embedded)
    x = layers.CuDNNGRU(128)(x)
    x = layers.Dense(32, activation='relu')(x)
    x = layers.Dense(100, activation='relu')(x)
    predictions = layers.Dense(1, activation='sigmoid')(x)
    model = models.Model(inputs=sequences, outputs=predictions)
    model.compile(
        optimizer='rmsprop',
        loss='binary_crossentropy',
        metrics=['binary_accuracy']
    )
    return model

rnn_model = build_rnn_model()
```

```
preds = rnn_model.predict(test_texts)
print('Accuracy score: {:.4}'.format(accuracy_score(test_labels, 1 * (preds > 0.5))))
print('F1 score: {:.4}'.format(f1_score(test_labels, 1 * (preds > 0.5))))
print('ROC AUC score: {:.4}'.format(roc_auc_score(test_labels, preds)))
```

```
Accuracy score: 0.9502
F1 score: 0.9504
ROC AUC score: 0.9881
```

The RNN model has yielded an accuracy of 95%.

Results of the refined model:

The model that we have built using CNN and RNN yields an accuracy of 95%. The model is able to predict the polarity of the amazon sentiment reviews accurately in most cases.

Possible Refinements in future:

Here are a few ways by which the accuracy of the model can be further improved:

Gated Recurrent Units(GRU):

GRU is a variation of recurrent neural networks (RNNs) that can effectively capture long-range dependencies in sequential data. Replacing LSTM with GRU in the model architecture might lead to better performance.

Transformers:

Transformers, particularly the Transformer architecture introduced in the "Attention is All You Need" paper, have shown remarkable success in various natural language processing tasks. Implementing a transformer-based model can potentially improve the model's understanding of context and relationships between words.

Bert Model:

Pre-trained BERT models have achieved state-of-the-art results in many NLP tasks. Fine-tuning a pre-trained BERT model on the sentiment analysis task can bring significant improvements to the accuracy.

Conclusion and remarks:

Our ongoing objective is to further enhance our model by leveraging advanced deep learning models. Nonetheless, it is essential to highlight our impressive achievement of 95% accuracy using CNN and RNN. We also recognize that there remains room for improvement and the need to explore additional facets while employing advanced techniques.