(SAWA) Using Attention to Predict Sentiment of Reviews

Panshul Jindal

July 2025

Contents

1 Introduction to Attention Mechanisms

Attention Mechanisms are a type of neural network that helps a text model focus on specific parts of the data which hold the most relevance. This is important as the more important features are considered more heavily in the long run.

The aim of this project is to explore attention mechanisms and how they impact the performance of a model. For this purpose, we explore the following mechanisms:

- Bahdanau (Additive) Attention
- Luong Dot Attention
- Luong General Attention
- Luong Concat Attention

These four mechanisms are tested on the following models:

- Unidirectional RNN
- Unidirectional LSTM
- Bidirectional RNN
- Bidirectional LSTM

Before going into further details, here is a brief description of each attention mechanism.

1.1 Additive Attention

This mechanism computes attention scores by combining the encoder states and the previous decoder state through a feedforward network.

$$\text{score}(h_t, s) = v^T \tanh(W_e h_t + W_d s) \tag{1}$$

Where:

- h_t: Encoder output at time step t.
- W_e, W_d: Learnable weight matrices.
- s: Decoder state.

This mechanism has been implemented by using linear layers in PyTorch for each weight and the 'tanh' activation from PyTorch.

1.2 Dot Attention

Dot Attention computes the attention scores via a dot product of the encoder states and the decoder state. This measures how aligned they are.

$$\text{score}(h_t, s) = h_t^T s \tag{2}$$

If the dimensions of h_t and s differ, as in the case of bidirectional networks, then a linear projection layer is applied on h_t to match the dimensions of s. This layer is learnable.

1.3 General Attention

General Attention builds up on dot attention and adds a learnable transformation matrix on the encoder outputs.

$$\text{score}(h_t, s) = (W h_t)^T s \tag{3}$$

Although this is already sort of implemented in dot attention if the dimensions do not match.

1.4 Concat Attention

This attention concatenates the encoder and decoder states and passes them through a neural network to compute scores.

$$\text{score}(h, s) = v^T \tanh(W[h; s]) \tag{4}$$

Note that this is very similar to additive attention.

2 Approach

2.1 Model Architectures

The four base models used have the following architecture:

- **Unidirectional RNN:** 100 embedding dimensions, 128 hidden dimensions.
- **Unidirectional LSTM:** 100 embedding dimensions, 128 hidden dimensions.
- **Bidirectional RNN:** 100 embedding dimensions, 256 hidden dimensions.
- **Bidirectional LSTM:** 100 embedding dimensions, 256 hidden dimensions.

The embedding layers of each model are trainable.

2.2 Attention Mechanisms

- **Additive Attention (Bahdanau):** Uses a feedforward network to compute alignment scores between the encoder hidden states and a query vector. Implemented using learned projections for both encoder and decoder states.
- **Dot Product Attention:** Computes alignment using the dot product between encoder states and the decoder query. Optionally uses a linear projection to match dimensionalities.
- **General Attention (Luong):** Applies a learned linear transformation to encoder states before computing the dot product with the query vector.
- **Concat Attention:** Concatenates encoder and query states before projecting them down to a scalar score via a learned feedforward network.

2.3 Attention Classifier

For attention-based models, a modular `BaseBodelPlugPlayAttention` is made for each and every base model, and attention class can be passed as argument. The attention-weighted context vector is passed to a final fully-connected layer to produce logits.

The final setup includes:

- 4 encoder types $\times$ 4 attention mechanisms = 16 attention-based models.
- 4 vanilla baselines without attention.

2.4 Learning Methadalogy

All models were trained with a fixed learning rate of 0.01 and with binary cross entropy loss using adam optimizer.

Training was done using early stopping with a patience of 5 epochs and with maximum number of 30 epochs.

3 Pipeline

The main Pipeline

- Train Test Val Split (Cause only train data must be used to learn vocabulary)
- Preprocessing
  - Tokenization using spacy
    - In Tokenization, Space and punctuation except epojis were removed
    - Stop words were removed excluding (negation words) (This became a proble later) To reduce the size for better training
  - Adding padding of max length in train dataset
- Train Model Pipeline (Saves the log of val and train loss and then saves the model weights)
- Getting Results Pipeline (Loads the model and calculate the classfication metrics on test dataset and saves it

4 Results

The dataset given consisted of 50,000 entries; 40,000 were used to train the models, and 10,000 each for validation and testing respectively. The labels were binary. We will look at the results for each model, comparing the major metrics among its variants.

4.1 Evaluation Metrics on Test Data

Following are tables comparing the evaluation metrics for each model variant.

Table 1: Unidirectional RNN

Metric	**No Attn**	**Additive**	**Dot**	**General**	**Concat**
Accuracy	0.5602	0.7902	0.5052	0.6110	0.7798
Macro Precision	0.5671	0.7272	0.5051	0.6390	0.9289
Macro Recall	0.5457	0.9354	1.0000	0.5279	0.6107
Macro F1	0.5562	0.8183	0.6712	0.5782	0.7369

Table 2: Unidirectional LSTM

Metric	**No Attn**	**Additive**	**Dot**	**General**	**Concat**
Accuracy	0.8526	0.8586	0.8652	0.8620	0.8126
Macro Precision	0.8341	0.8875	0.8665	0.8517	0.8449
Macro Recall	0.8840	0.8246	0.8665	0.8800	0.7703
Macro F1	0.8583	0.8549	0.8665	0.8656	0.8059

Table 3: Bidirectional RNN

Metric	**No Attn**	**Additive**	**Dot**	**General**	**Concat**
Accuracy	0.4894	0.7790	0.6120	0.5588	0.8096
Macro Precision	0.4944	0.7133	0.5746	0.5835	0.7523
Macro Recall	0.4859	0.9402	0.8923	0.4416	0.9287
Macro F1	0.4901	0.8112	0.6990	0.5027	0.8313

Table 4: Bidirectional LSTM

Metric	No Attn	Additive	Dot	General	Concat
Accuracy	0.5090	0.8458	0.8646	0.8566	0.8596
Macro Precision	0.5103	0.8265	0.8505	0.8582	0.9071
Macro Recall	0.6875	0.8792	0.8879	0.8578	0.8044
Macro F1	0.5858	0.8520	0.8688	0.8580	0.8526

4.2 Visualization of Attention Weights

This section would contain the visualizations of attention weights for positive, negative, and mixed sentiment reviews. The heatmaps show how focus was distributed differently across words and how these words lead to high confidence in the model's predictions.

4.2.1 Positive Review

Text: = "I absolutely loved this movie. The performances were brilliant, the story was engaging from start to finish, and the cinematography was stunning. One of the best films I've seen this year!"

Note: You would have to zoom in to get to know at which word the attention is focused

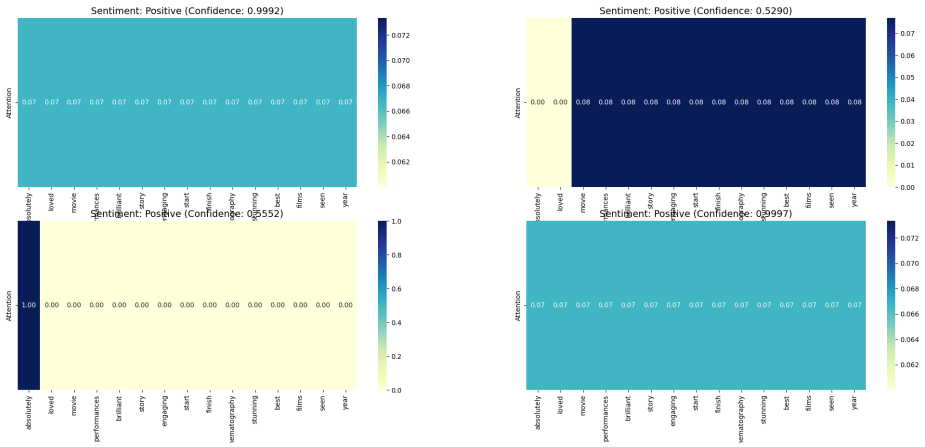

Figure 1: Unidirectional RNN

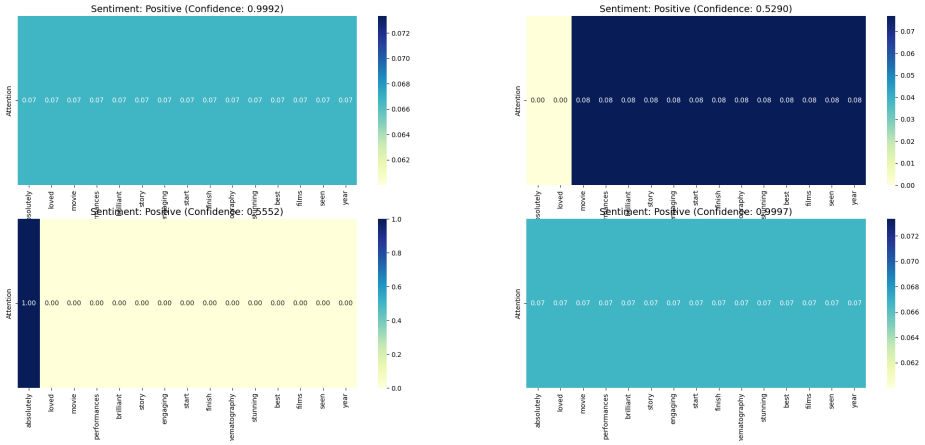

Figure 2: Unidirectional LSTM

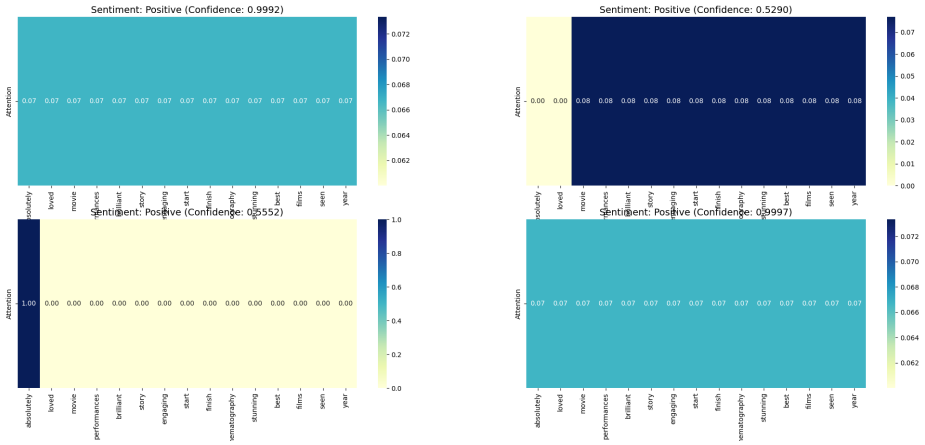

Figure 3: BiDirectional RNN

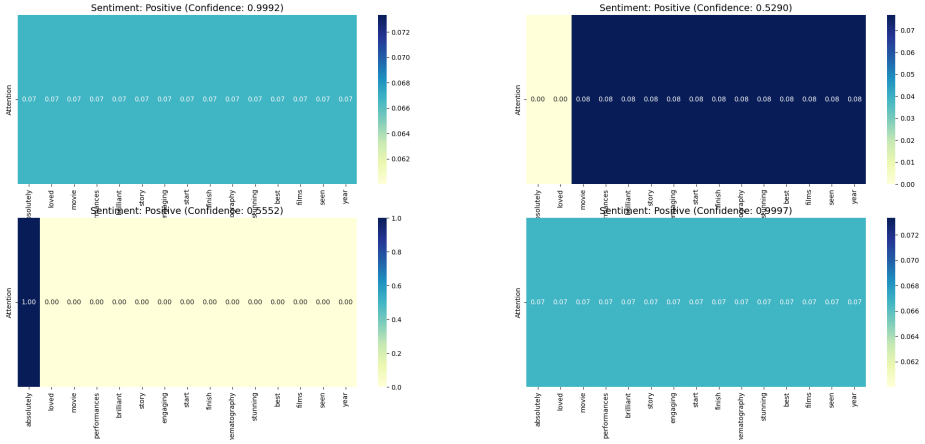

Figure 4: Bidirectional LSTM

4.2.2 Negative Review

Text: = "This was a complete waste of time. The plot made no sense, the acting was terrible, and the pacing was painfully slow. I couldn't wait for it to end"

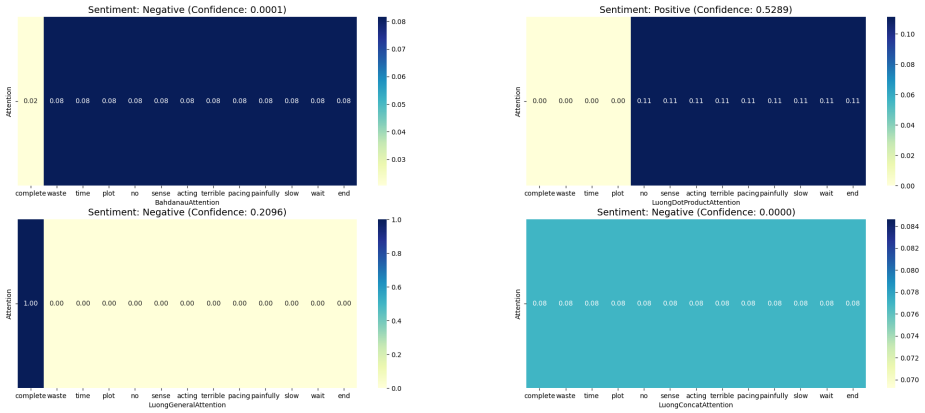

Figure 5: Unidirectional RNN

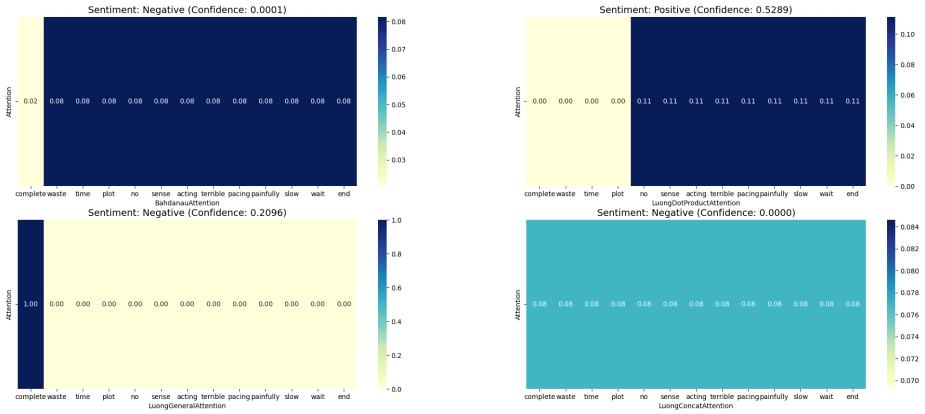

Figure 6: Unidirectional LSTM

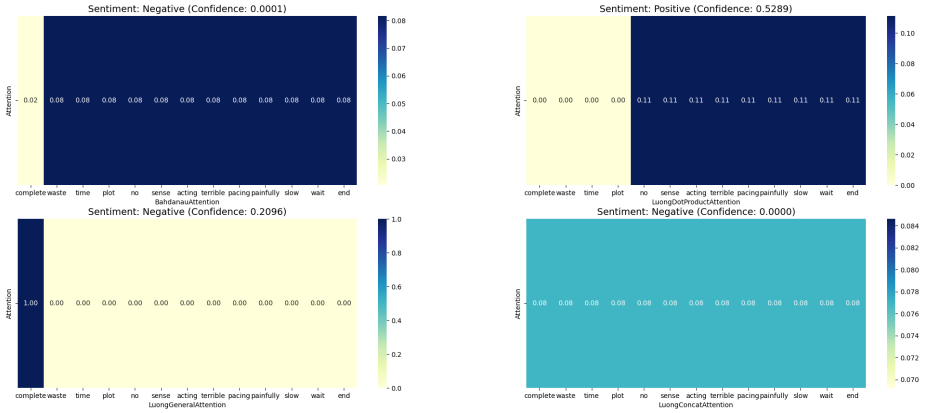

Figure 7: BiDirectional RNN

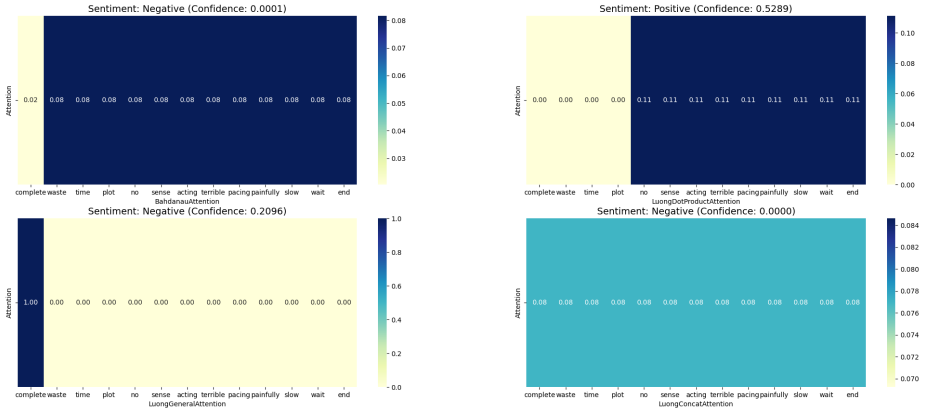

Figure 8: Bidirectional LSTM

4.2.3 Mixed Review

Text = "The movie started off strong with a great setup and some interesting characters, but it lost momentum halfway through. While the visuals were impressive, the ending felt rushed and unsatisfying"

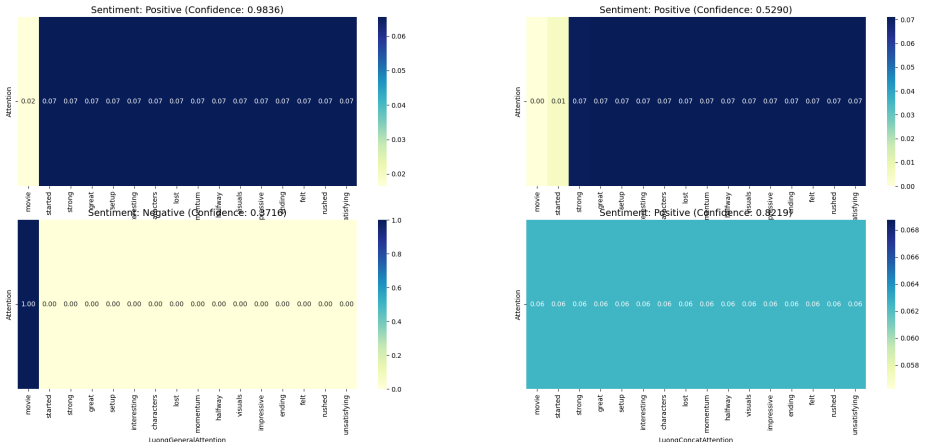

Figure 9: Unidirectional RNN

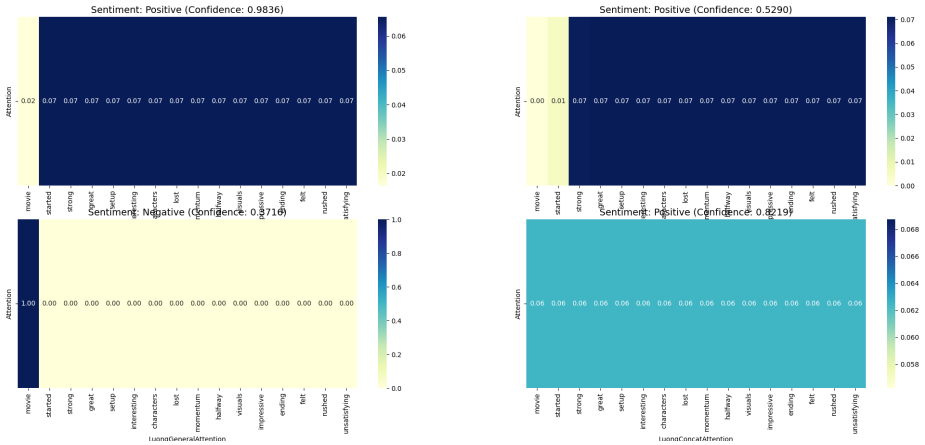

Figure 10: Unidirectional LSTM

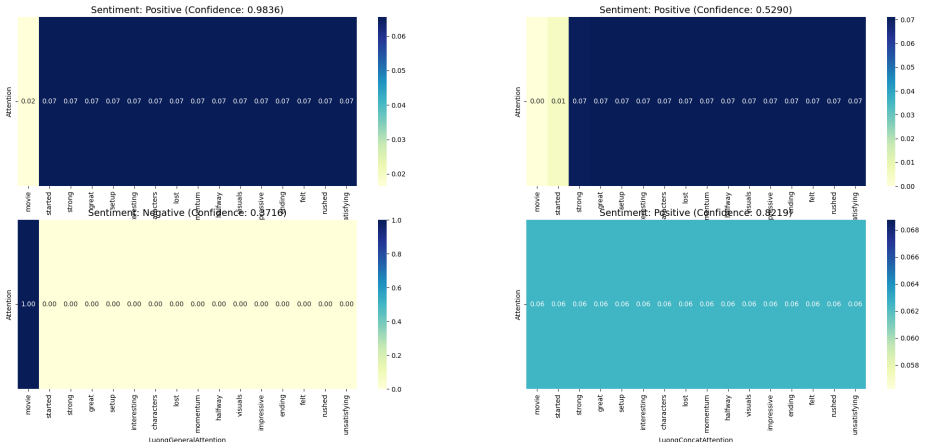

Figure 11: BiDirectional RNN

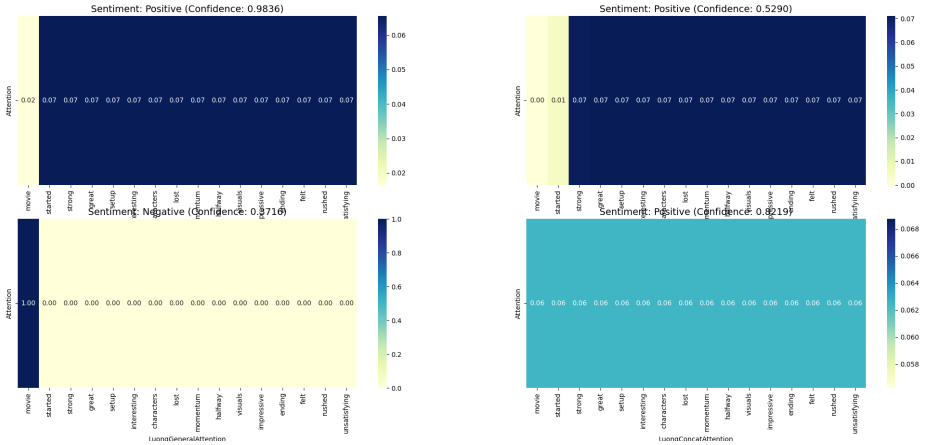

Figure 12: Bidirectional LSTM

5 Key Insights and Learnings

- **Effectiveness of Attention Mechanisms:** Attention significantly improved the performance across most models, particularly additive and concat attention . Models with these two consistently showed high precision, recall and F1 scores.It also lead to greater interpretability
- **Bi-directionality:** Bidirectional models outperformed their unidirectional counterparts in almost every combination, with the BiLSTM + Dot Attention achieving the best results overall.
- **Different types of Attention:** These dot and general mechanisms underperformed, probably due to their simplistic nature (less number of parameters). While concat and additive performed really well.
- **Attention Visualization:** Looking at the heatmaps, we can see how focus was distributed differently across words. And how these words lead to high confidence in the mixed reviews.
- **Learnings:** Implementing and testing multiple attention mechanisms provided deep insight into their mathematical foundations and practical behavior.