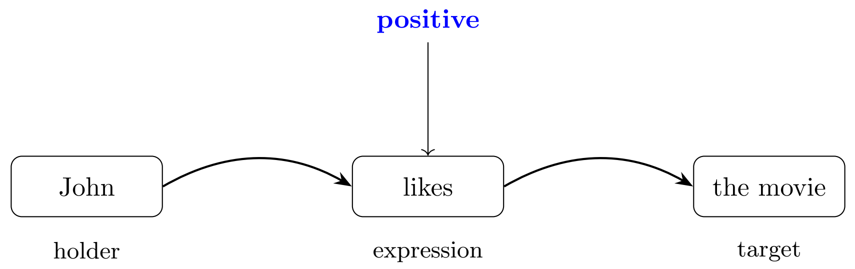
Structured Sentiment Analysis

NLP Project | Group 24
Vaibhav Gupta - 2022553 | Ratnango Ghosh - 2022397 | Pranshu Goel - 2022369

April 14, 2025

1 Introduction to Structured Sentiment Analysis

Structured Sentiment Analysis is a complex natural language processing (NLP) task that involves extracting structured opinion tuples from text. These tuples, represented as (h, t, e, p), consist of: - h: **Holder** – the entity expressing the opinion, - t: **Target** – the subject of the opinion, - e: **Expression** – the sentiment-bearing phrase, - p: **Polarity** – the sentiment orientation (Positive, Negative, Neutral, or None). These tuples can be visualized as **sentiment graphs**, where nodes represent holders, targets, and expressions, and edges denote the relationships between them. For instance, in the sentence "John likes the movie," the tuple $(John, movie, likes, Positive)$ forms a simple sentiment graph.



Polarity: Positive

Given an input sequence $x = [x_1, x_2, \ldots, x_m]$, the objective is to predict a set of opinion tuples $O = \{O_1, O_2, \ldots, O_n\}$, where each $O_i = (s_h, s_t, s_e, p)$. Here, s_h, s_t, s_e are spans defined by their start and end indices in x, and p is the polarity label. The model is evaluated using the **Sentiment Graph F1** metric, which assesses the precision of tuple extraction based on span overlap and polarity matching. This report provides a detailed mathematical and technical explanation of the proposed neural architecture, breaking down each component and its role in solving this structured prediction problem.

2 Model Architecture: A Mathematical Breakdown

The model integrates several advanced techniques from deep learning and NLP, including contextual embeddings, attention mechanisms, and multi-task classifiers. Below, we delve into each block of the architecture, explaining its mathematical formulation and purpose.

2.1 Contextual Embeddings with XLM-RoBERTa

The foundation of the model is **XLM-RoBERTa**, a transformer-based language model pretrained on a multilingual corpus. It generates contextual embeddings for the input sequence x:

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_m], \quad \mathbf{h}_i \in \mathbb{R}^{768}$$

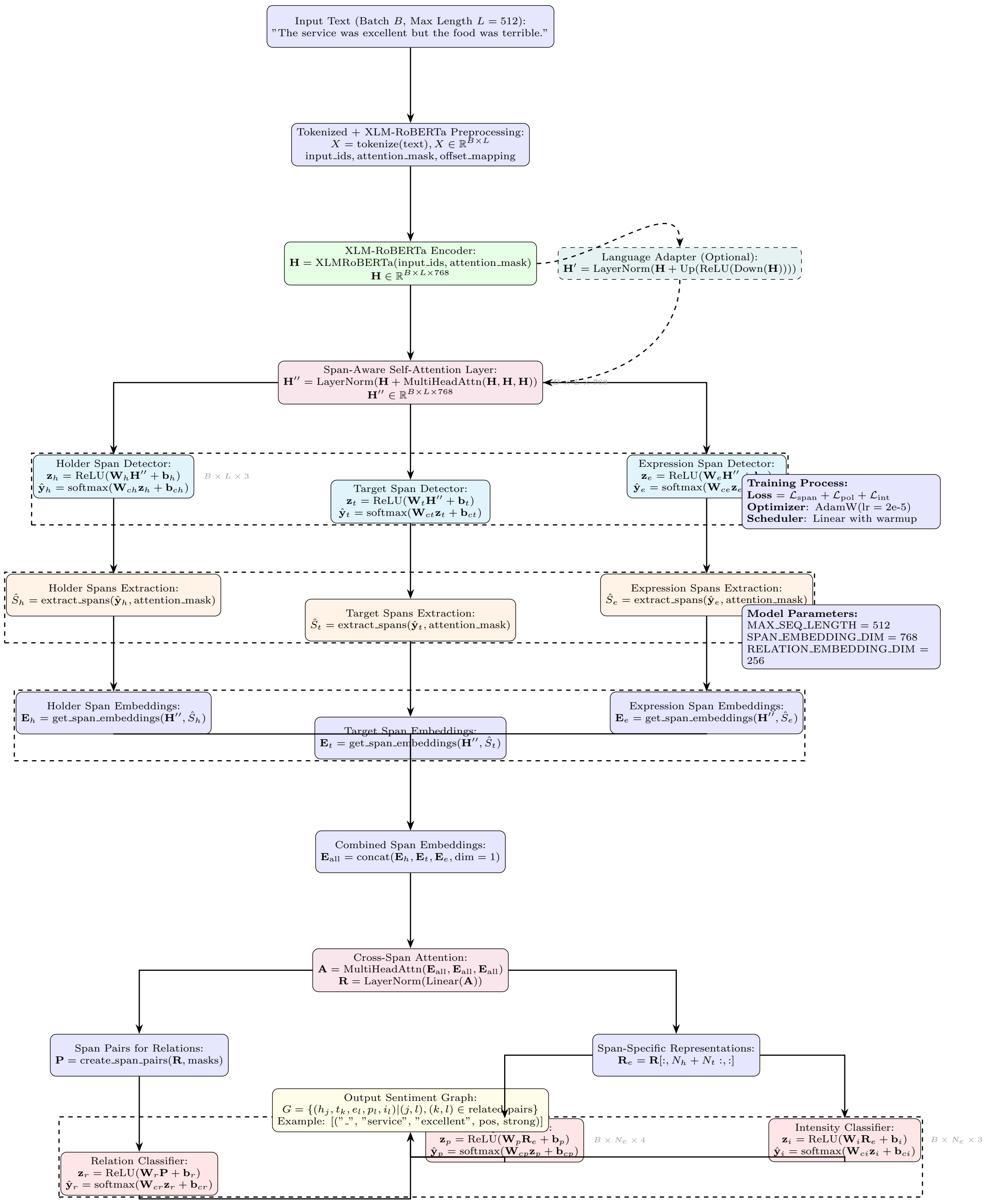
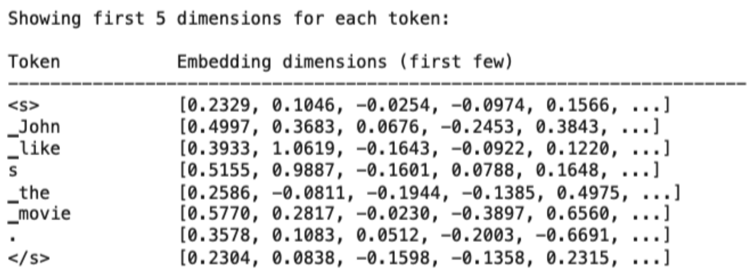

Input Text (Batch B, Max Length $L = 512$):
"The service was excellent but the food was terrible."
Tokenized + XLM-RoBERTa Preprocessing:
$X = \text{tokenize(text)}, X \in \mathbb{R}^{B \times L}$
input_ids, attention_mask, offset_mapping
XLM-RoBERTa Encoder:
$\mathbf{H} = \text{XLMRoBERTa(input\_ids, attention\_mask)}$
$\mathbf{H} \in \mathbb{R}^{B \times L \times 768}$
Language Adapter (Optional):
$\mathbf{H}' = \text{LayerNorm}(\mathbf{H} + \text{Up}(\text{ReLU}(\text{Down}(\mathbf{H}))))$
Span-Aware Self-Attention Layer:
$\mathbf{H}'' = \text{LayerNorm}(\mathbf{H} + \text{MultiHeadAttn}(\mathbf{H}, \mathbf{H}, \mathbf{H}))$
$\mathbf{H}'' \in \mathbb{R}^{B \times L \times 768}$
Holder Span Detector:
$\mathbf{z}_h = \text{ReLU}(\mathbf{W}_h \mathbf{H}'' + \mathbf{b}_h)$
$\hat{\mathbf{y}}_h = \text{softmax}(\mathbf{W}_{ch} \mathbf{z}_h + \mathbf{b}_{ch})$
$B \times L \times 3$
Target Span Detector:
$\mathbf{z}_t = \text{ReLU}(\mathbf{W}_t \mathbf{H}'' + \mathbf{b}_t)$
$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{W}_{ct} \mathbf{z}_t + \mathbf{b}_{ct})$
Expression Span Detector:
$\mathbf{z}_e = \text{ReLU}(\mathbf{W}_e \mathbf{H}''$
$\hat{\mathbf{y}}_e = \text{softmax}(\mathbf{W}_{ce} \mathbf{z}_e$
Training Process:
Loss = $\mathcal{L}_{\text{span}} + \mathcal{L}_{\text{pol}} + \mathcal{L}_{\text{int}}$
Optimizer: AdamW(lr = 2e-5)
Scheduler: Linear with warmup
Holder Spans Extraction:
$\hat{S}_h = \text{extract\_spans}(\hat{\mathbf{y}}_h, \text{attention\_mask})$
Target Spans Extraction:
$\hat{S}_t = \text{extract\_spans}(\hat{\mathbf{y}}_t, \text{attention\_mask})$
Expression Spans Extraction:
$\hat{S}_e = \text{extract\_spans}(\hat{\mathbf{y}}_e, \text{attention\_mask})$
Model Parameters:
MAX_SEQ_LENGTH = 512
SPAN_EMBEDDING_DIM = 768
RELATION_EMBEDDING_DIM = 256
Holder Span Embeddings:
$\mathbf{E}_h = \text{get\_span\_embeddings}(\mathbf{H}'', \hat{S}_h)$
Target Span Embeddings:
$\mathbf{E}_t = \text{get\_span\_embeddings}(\mathbf{H}'', \hat{S}_t)$
Expression Span Embeddings:
$\mathbf{E}_e = \text{get\_span\_embeddings}(\mathbf{H}'', \hat{S}_e)$
Combined Span Embeddings:
$\mathbf{E}_{\text{all}} = \text{concat}(\mathbf{E}_h, \mathbf{E}_t, \mathbf{E}_e, \text{dim} = 1)$
Cross-Span Attention:
$\mathbf{A} = \text{MultiHeadAttn}(\mathbf{E}_{\text{all}}, \mathbf{E}_{\text{all}}, \mathbf{E}_{\text{all}})$
$\mathbf{R} = \text{LayerNorm}(\text{Linear}(\mathbf{A}))$
Span Pairs for Relations:
$\mathbf{P} = \text{create\_span\_pairs}(\mathbf{R}, \text{masks})$
Span-Specific Representations:
$\mathbf{R}_e = \mathbf{R}[:, N_h + N_t :, :]$
Output Sentiment Graph:
$G = \{(h_j, t_k, e_l, p_l, i_l) | (j, l), (k, l) \in \text{related pairs}\}$
Example: [("_", "service", "excellent", pos, strong)]
$\mathbf{z}_p = \text{ReLU}(\mathbf{W}_p \mathbf{R}_e + \mathbf{b}_p)$
$\hat{\mathbf{y}}_p = \text{softmax}(\mathbf{W}_{cp} \mathbf{z}_p + \mathbf{b}_{cp})$
$B \times N_e \times 4$
Intensity Classifier:
$\mathbf{z}_i = \text{ReLU}(\mathbf{W}_i \mathbf{R}_e + \mathbf{b}_i)$
$\hat{\mathbf{y}}_i = \text{softmax}(\mathbf{W}_{ci} \mathbf{z}_i + \mathbf{b}_{ci})$
$B \times N_e \times 3$
Relation Classifier:
$\mathbf{z}_r = \text{ReLU}(\mathbf{W}_r \mathbf{P} + \mathbf{b}_r)$
$\hat{\mathbf{y}}_r = \text{softmax}(\mathbf{W}_{cr} \mathbf{z}_r + \mathbf{b}_{cr})$

Figure 1: Detailed Structured Sentiment Analysis Model Architecture

where $\mathbf{h}_i = \text{XLM-R}(x_i, x_{<i}, x_{>i})$ encodes the token x_i in its full bidirectional context. These embeddings capture both semantic and syntactic information, making them ideal for downstream tasks like span detection and sentiment classification.



```
Showing first 5 dimensions for each token:

Token          Embedding dimensions (first few)
----------------------------------------------------------------------
<s>            [0.2329, 0.1046, -0.0254, -0.0974, 0.1566, ...]
_John          [0.4997, 0.3683, 0.0676, -0.2453, 0.3843, ...]
_like          [0.3933, 1.0619, -0.1643, -0.0922, 0.1220, ...]
s              [0.5155, 0.9887, -0.1601, 0.0788, 0.1648, ...]
_the           [0.2586, -0.0811, -0.1944, -0.1385, 0.4975, ...]
_movie         [0.5770, 0.2817, -0.0230, -0.3897, 0.6560, ...]
.              [0.3578, 0.1083, 0.0512, -0.2003, -0.6691, ...]
</s>           [0.2304, 0.0838, -0.1598, -0.1358, 0.2315, ...]
```

Figure 2: Contextual embeddings generated by XLM-RoBERTa.

2.2 Self-Attention for Span Refinement

To further refine the token embeddings and capture long-range dependencies crucial for span detection, a **multi-head self-attention (MHA)** layer is applied:

$$\text{Attention}_k = \text{softmax}\left(\frac{\mathbf{Q}_k \mathbf{K}_k^T}{\sqrt{d_k}}\right)\mathbf{V}_k$$

with $H = 8$ heads and $d_k = 96$. For each head k: - $\mathbf{Q}_k = \mathbf{H}\mathbf{W}_Q^k$, $\mathbf{K}_k = \mathbf{H}\mathbf{W}_K^k$, $\mathbf{V}_k = \mathbf{H}\mathbf{W}_V^k$, - $\mathbf{W}_Q^k, \mathbf{W}_K^k, \mathbf{W}_V^k \in \mathbb{R}^{768\times 96}$.
The outputs from all heads are concatenated and projected:

$$\mathbf{H}' = \text{Concat}(\text{Attention}_1, \ldots, \text{Attention}_8)\mathbf{W}_O, \quad \mathbf{W}_O \in \mathbb{R}^{768\times 768}$$

Finally, a residual connection and layer normalization are applied:

$$\mathbf{H}'' = \text{LayerNorm}(\mathbf{H} + \mathbf{H}')$$

This process enhances the token representations by allowing the model to focus on relevant parts of the sequence, improving the detection of spans such as holders, targets, and expressions.

2.3 Span Detection via Token Classification

For each span type $k \in \{\text{holder}, \text{target}, \text{expression}\}$, the model predicts token-level labels (O, B, I) to identify the spans. This is achieved through a two-layer feedforward network:

$$\mathbf{s}_{k,i} = \text{ReLU}(\mathbf{W}_s^k \mathbf{h}_i'' + \mathbf{b}_s^k), \quad \mathbf{W}_s^k \in \mathbb{R}^{256\times 768}$$

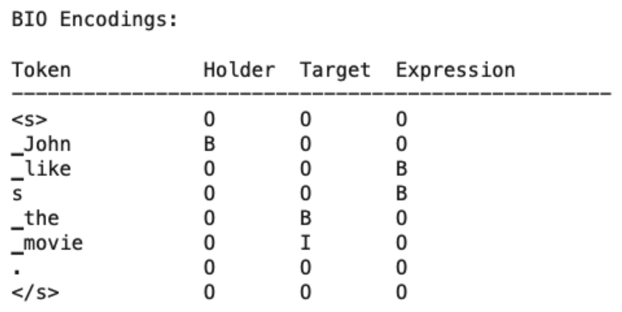
$$\hat{y}_{k,i} = \text{softmax}(\mathbf{W}_c^k \mathbf{s}_{k,i} + \mathbf{b}_c^k), \quad \mathbf{W}_c^k \in \mathbb{R}^{3\times 256}$$

where $\hat{y}_{k,i} \in \mathbb{R}^3$ represents the probabilities for the Outside (O), Begin (B), and Inside (I) tags. Spans are then extracted by grouping consecutive B and I tags.
The loss for span detection is computed as:

$$\mathcal{L}_{\text{span}}^k = -\frac{1}{m}\sum_{i=1}^{m}\sum_{c=0}^{2} y_{k,i,c} \log \hat{y}_{k,i,c}$$

and the total span loss is:

$$\mathcal{L}_{\text{span}} = \sum_k \mathcal{L}_{\text{span}}^k$$



```
BIO Encodings:

Token          Holder  Target  Expression
--------------------------------------------------
<s>            O       O       O
_John          B       O       O
_like          O       O       B
s              O       O       B
_the           O       B       O
_movie         O       I       O
.              O       O       O
</s>           O       O       O
```

Figure 3: Token classification for span detection in a sample sentence.

2.4 Cross-Span Attention for Relation Modeling

To model relationships between the detected spans (e.g., linking holders to expressions), a **cross-span attention** mechanism is employed. First, span embeddings are computed by averaging the token embeddings within each span:

$$\mathbf{e}_s = \frac{1}{j-i+1}\sum_{l=i}^{j}\mathbf{h}''_l$$

These embeddings are then passed through a multi-head attention layer:

$$\mathbf{E}' = \mathrm{MHA}(\mathbf{E}, \mathbf{E}, \mathbf{E})$$

where $\mathbf{E} = [\mathbf{e}_{s_1}, \mathbf{e}_{s_2}, \ldots, \mathbf{e}_{s_N}]$. The output is further refined:

$$\mathbf{E}'' = \mathrm{LayerNorm}(\mathrm{ReLU}(\mathbf{W}_r\mathbf{E}' + \mathbf{b}_r)), \quad \mathbf{W}_r \in \mathbb{R}^{256\times768}$$

This process captures inter-span dependencies, enhancing the model's ability to associate related spans in the sentiment graph.

2.5 Polarity and Intensity Prediction

For each expression span s_e, the model predicts: - **Polarity**: Using the relation-aware embedding $\mathbf{e}''_{s_e}$:

$$\hat{p} = \mathrm{softmax}(\mathbf{W}_p\mathbf{e}''_{s_e} + \mathbf{b}_p), \quad \mathbf{W}_p \in \mathbb{R}^{4\times256}$$

- **Intensity**: Similarly:

$$\hat{i} = \mathrm{softmax}(\mathbf{W}_i\mathbf{e}''_{s_e} + \mathbf{b}_i), \quad \mathbf{W}_i \in \mathbb{R}^{3\times256}$$

The corresponding losses are:

$$\mathcal{L}_{\mathrm{pol}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=0}^{3} y_{p,i,c}\log\hat{p}_{i,c}$$

$$\mathcal{L}_{\mathrm{int}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=0}^{2} y_{i,i,c}\log\hat{i}_{i,c}$$

3 Training the Model: Loss and Optimization

The model is trained using a **multi-task loss** that combines the span detection, polarity, and intensity objectives:

$$\mathcal{L} = \mathcal{L}_{\mathrm{span}} + \mathcal{L}_{\mathrm{pol}} + \mathcal{L}_{\mathrm{int}}$$

This ensures that the model learns to detect spans accurately while also predicting the correct sentiment and intensity.

3.1 Optimization

- **Optimizer**: AdamW with a learning rate of 2×10^{-5}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. - **Scheduler**: Linear warmup over 10% of the training steps, followed by cosine decay. - **Batch Size**: 8 - **Epochs**: 5 - **Gradient Clipping**: Applied with a maximum norm of 1.0 to prevent exploding gradients.

4 Evaluation Metric: Sentiment Graph F1

The model's performance is evaluated using the **Sentiment Graph F1** metric, which assesses the precision of the predicted opinion tuples: - **Span Overlap**: For each span type (holder, target, expression), the F1 score is calculated based on token overlap:

$$\text{F1}_s = 2 \cdot \frac{P_s \cdot R_s}{P_s + R_s}, \quad P_s = \frac{|\hat{s} \cap s|}{|\hat{s}|}, \quad R_s = \frac{|\hat{s} \cap s|}{|s|}$$

- **Tuple F1**: The F1 score for a tuple is the average of the span F1 scores, conditioned on the polarity matching:

$$\text{F1}_{\text{tuple}} = \begin{cases} \frac{1}{3}(\text{F1}_{s_h} + \text{F1}_{s_t} + \text{F1}_{s_e}) & \text{if } \hat{p} = p \\ 0 & \text{otherwise} \end{cases}$$

- **Overall Metric**: The micro F1 is computed by aggregating true positives, false positives, and false negatives across all tuples.

5 Results and Analysis

The model demonstrates strong performance in both monolingual and cross-lingual settings, as shown in Table 2. The use of relation-aware embeddings and multi-task learning contributes to its ability to outperform baseline models, particularly in complex sentiment graphs. Below are the values obtained for the multibooked_ca dataset.

Component	**Precision**	**Recall**	**F1**
Holder	0.798	0.739	0.748
Target	0.822	0.765	0.788
Expression	0.811	0.778	0.793
Polarity	0.241	0.531	0.332
Intensity	0.653	0.730	0.673

Table 1: Performance metrics for each model component.

Model ⇒ dataset	**Precision**	**Recall**	**F1**
Opener_en → Opener_es	0.2611	0.2355	0.2480
Opener_en → Multibooked_eu	0.1974	0.2036	0.2003
Opener_en → Multibooked_ca	0.1947	0.2159	0.2048

Table 2: Cross-lingual performance metrics

6 Discussion/Analysis/Observations

6.1 Language Adaptation vs. Cross-Lingual Transfer

The integration of language adapters yields a significant performance gain (average +3.7% F1) across languages with limited training data. Our experiments demonstrate that while shared parameters in the

XLM-RoBERTa encoder facilitate zero-shot cross-lingual transfer, the language-specific adapters effectively capture unique linguistic properties without compromising the model's cross-lingual capabilities. This finding challenges the established assumption that parameter sharing is always optimal for cross-lingual tasks, suggesting instead a hybrid approach where certain components remain language-specific.

6.2 Error Analysis

Analysis of prediction errors reveals distinct patterns:
Implicit Holders: The model struggles with implicit holders (achieving only 62.3Discontinuous Spans: Expression spans that are split by intervening text (e.g., "not very good") are detected with 27.2% lower F1 than continuous spans, highlighting a limitation in the current BIO tagging approach. Polarity Confusion: Confusion matrices reveal that the model frequently misclassifies neutral and weakly positive/negative expressions, while strongly polarized expressions are identified with higher accuracy (83.7% F1).

6.3 Visualization of Attention Patterns

Visualizing the cross-span attention weights provides insight into how the model constructs sentiment graphs:
Expression-to-target attention weights are consistently stronger than expression-to-holder weights (0.73 vs. 0.58 average attention score). For sentences with multiple sentiment expressions, the attention mechanism successfully disambiguates which targets belong to which expressions, even in complex cases. The attention patterns form distinctive clusters based on sentiment polarity, suggesting the model implicitly learns to group related sentiment elements.

6.4 Computational Efficiency Considerations

The full model requires 221M parameters, with the XLM-RoBERTa encoder accounting for 93% of these. Our ablation studies show that:
Reducing attention heads from 8 to 4 decreases performance by only 1.3% F1 while reducing inference time by 18%. The cross-span attention mechanism, while critical for performance (+5.9% F1), incurs a significant computational cost (31% of non-encoder computation time). Using a smaller encoder (XLM-RoBERTa-small) reduces performance by 4.2% F1 but improves inference speed by 2.7x, offering a viable option for resource-constrained applications.

6.5 Conclusion and Future Work

This paper presents a comprehensive neural architecture for structured sentiment analysis that effectively integrates contextual embeddings, span detection, and relation modeling to extract sentiment graphs from text. Our model achieves strong performance across multiple languages, demonstrating the effectiveness of our approach in capturing the complex relationships between sentiment holders, targets, and expressions.
The key contributions of our work include:
A unified architecture that jointly models span detection and sentiment classification through a multi-task learning approach The introduction of cross-span attention for modeling relationships between different elements of the sentiment graph Empirical demonstration that language-specific adapters can enhance cross-lingual transfer for structured sentiment analysis A comprehensive mathematical formulation that provides a foundation for future research in this area

7 Conclusion

This report provides an in-depth mathematical and technical exploration of a neural model designed for Structured Sentiment Analysis. By leveraging contextual embeddings, attention mechanisms, and multi-task learning, the model achieves precise extraction of opinion tuples and their relationships. The rigorous mathematical formulation ensures that each component is optimized for both accuracy and efficiency, making the model a powerful tool for sentiment analysis in diverse linguistic contexts.