Presentation by **Group 25**

Project 4

Tweet Sentiment Phrase Extraction

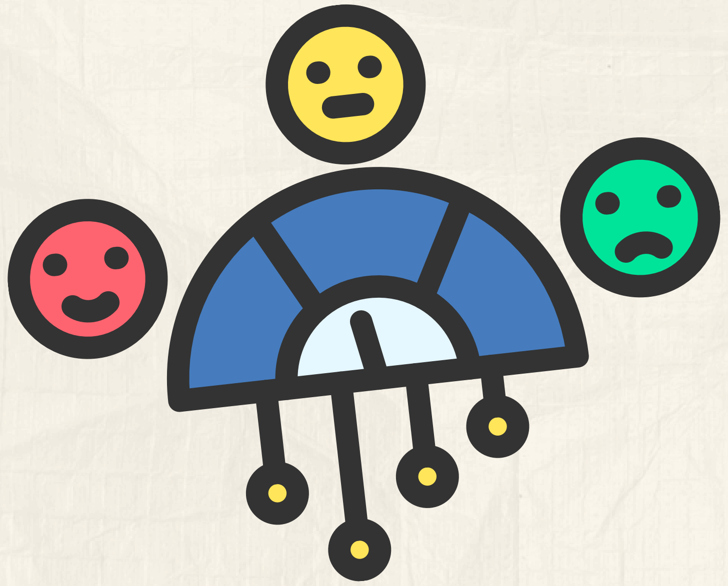
ATI

Dinh Thi Tu
Nguyen Thi Hien
Nguyen Thanh Phuong

ATI

TABLE OF CONTENT

Tweet Sentiment Phrase Extraction



lovely

LOVE

ATI

01. Problem Formulation

Problem

Traditional sentiment analysis models only predict an overall label (positive, negative, or neutral) for a tweet.

=> Lack of interpretability

Scope

Extracting the exact phrase (span) in the tweet that justifies the given sentiment.

ATI

01. Problem Formulation

Task Definition

We formulate this problem as an Extractive Question Answering (QA) task.

- Context: The tweet text.
- Question: Which part of the text expresses the positive/negative/neutral sentiment?

=> Predict the start and end positions of the answer span (the sentiment phrase) within the tokenized text.

Challenges

Noisy and informal (eg. slang, abbreviations, typos, emojis, hashtags, URLs), sarcasm and mixed emotions

=> Handling noise and accurately identifying boundaries.

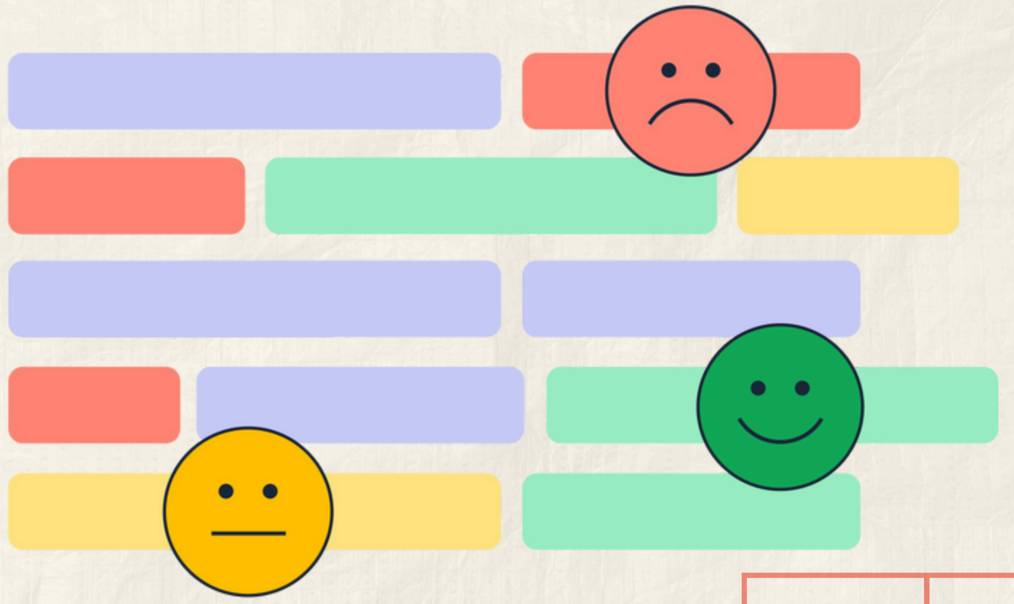
02. Input & Output

Input

Component	Description
Tweet Text	Raw tweet string (noisy, short, with emojis, slang)
Sentiment Label	Given label: positive, negative, or neutral
Format	Combined as QA prompt: "Extract the {sentiment} phrase: [tweet]”

Output

- Predicted phrase (selected_text): Exact span from tweet
- Format: String substring justifying the sentiment
- Additional: Start/end character positions (internal for decoding)



ATI

03. Data preparation & preprocessing

Train dataset

textID	text	selected_text	sentiment
27481 unique values	27481 unique values	22464 unique values	neutral 40% positive 31% Other (7781) 28%

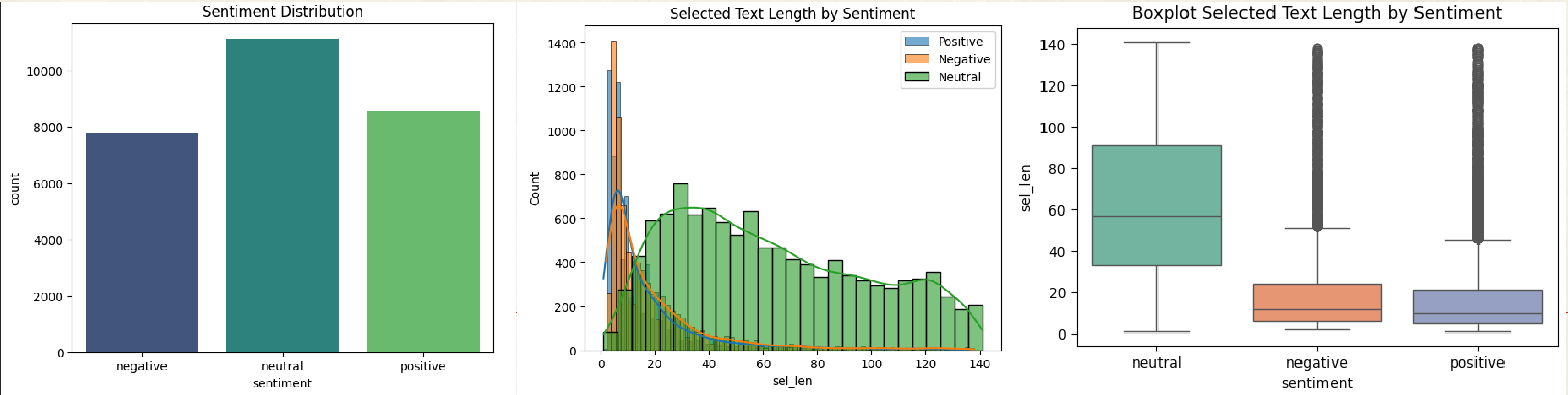
Test dataset

textID	text	sentiment
3534 unique values	3534 unique values	neutral 40% positive 31% Other (1001) 28%

EDA Summary - Key Insights:

- No severe class imbalance, neutral cases are the most common.
- Neutral tweets often use the full tweet as selected_text, Positive and negative tweets tend to have shorter, more focused phrases.
- Sentiment distribution is consistent between train and test

→ Reduces risk of distribution shift during inference.



ATI

3. Data preparation & preprocessing

The Core Challenge in Preprocessing: Token-Character Alignment

The misalignment between human annotation and model tokenization.

- Humans select phrases by characters in the raw text.
- Models (RoBERTa) process text as subword tokens.

When we add the QA prompt ("Extract the {sentiment} phrase:"), token positions shift → direct mapping causes boundary errors.

Solution – Robust Offset Mapping

- Enable return_offsets_mapping=True during tokenization
- Compute offsets for the full input (including QA prompt)
- Adjust offsets by subtracting the length of the prompt prefix
- Map predicted start/end token indices back to exact character positions in the original tweet

- Precise boundary recovery, even with emojis, punctuation, and added prompt
- Critical for achieving high word-level and character-level Jaccard scores

TOKEN ID	TOKEN	START CHAR	END CHAR
0	[CLS]	-	-
1	Extract	-	-
2	the	-	-
3	positive	-	-
4	phrase	-	-
5	:	-	-
6	I	0	1
7	love	2	6
8	this	7	11
9	phone	12	17
10	!	17	18
11	!	18	19
12	!	19	20
13	[SEP]	-	-

Special Tokens (CLS, SEP) **Prompt Tokens** (no char positions) **Original Text Tokens** (with positions)

📍 Character Position Mapping

I love this phone!!!

I	_	l	o	v	e	_	t	h	i	s	_	p	h	o	n	e	!	!	!
0-1	1-2	2	3	4	5-6	6-7	7	8	9	10-11	11-12	12	13	14	15	16-17	17-18	18-19	19-20

ATI

04. Optimization techniques

1. Training Optimization

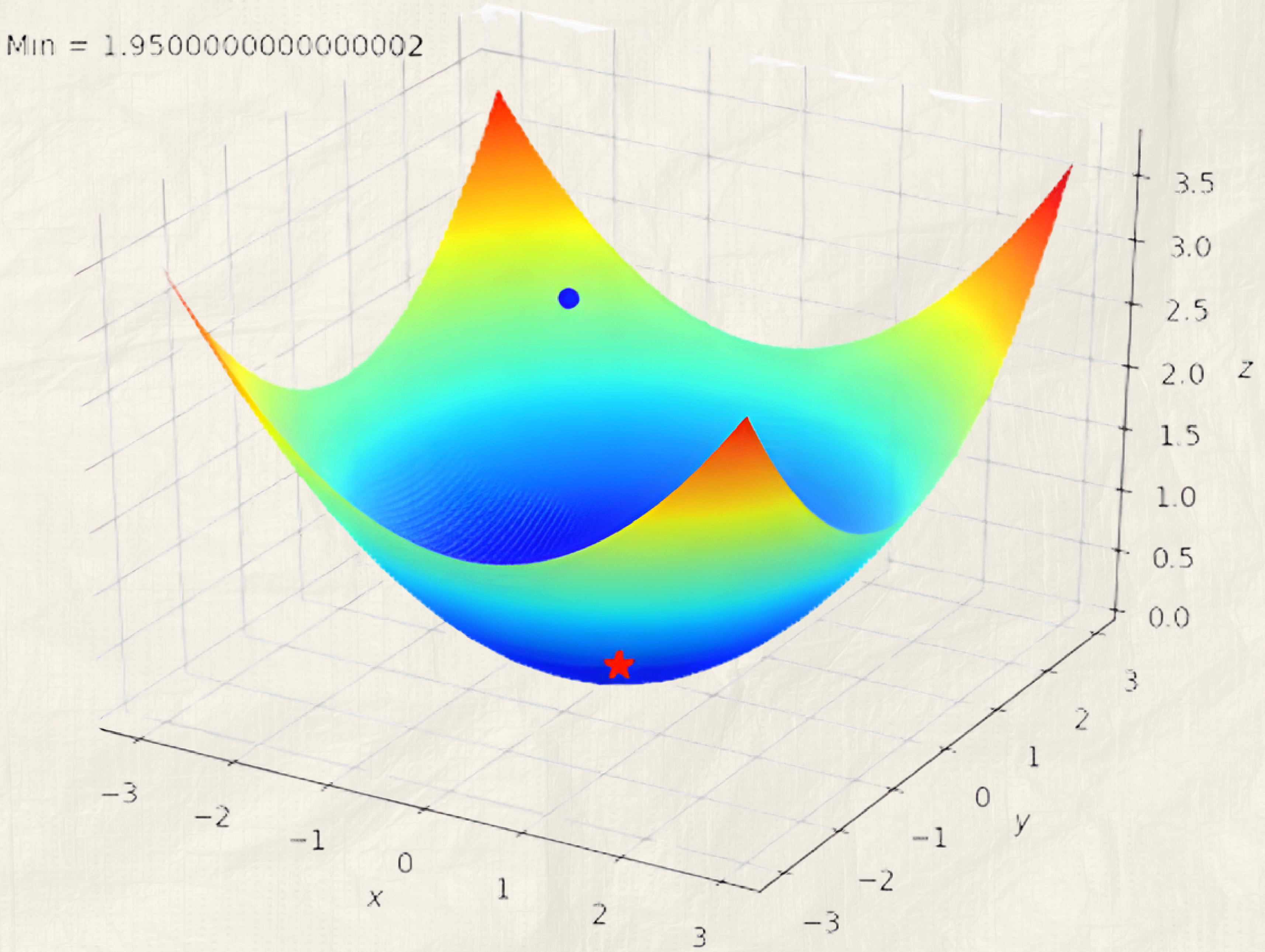
Includes: AdamW + Weight Decay, Linear Scheduler, Gradient Clipping, Label Smoothing, Long Training + Early Stopping, 5-fold Stratified K-Fold.

Ensure stable training, reduce overfitting, and achieve strong generalization on noisy tweet data.

2. Inference Optimization

Includes: Top-K Decoding, Precise Offset + Valid Token Mask, Neutral Trick, 5-fold Ensemble (majority vote).

Maximize character-level accuracy, overcome argmax errors & variance → significantly boost Jaccard score.



ATI

05. Architecture & training strategy

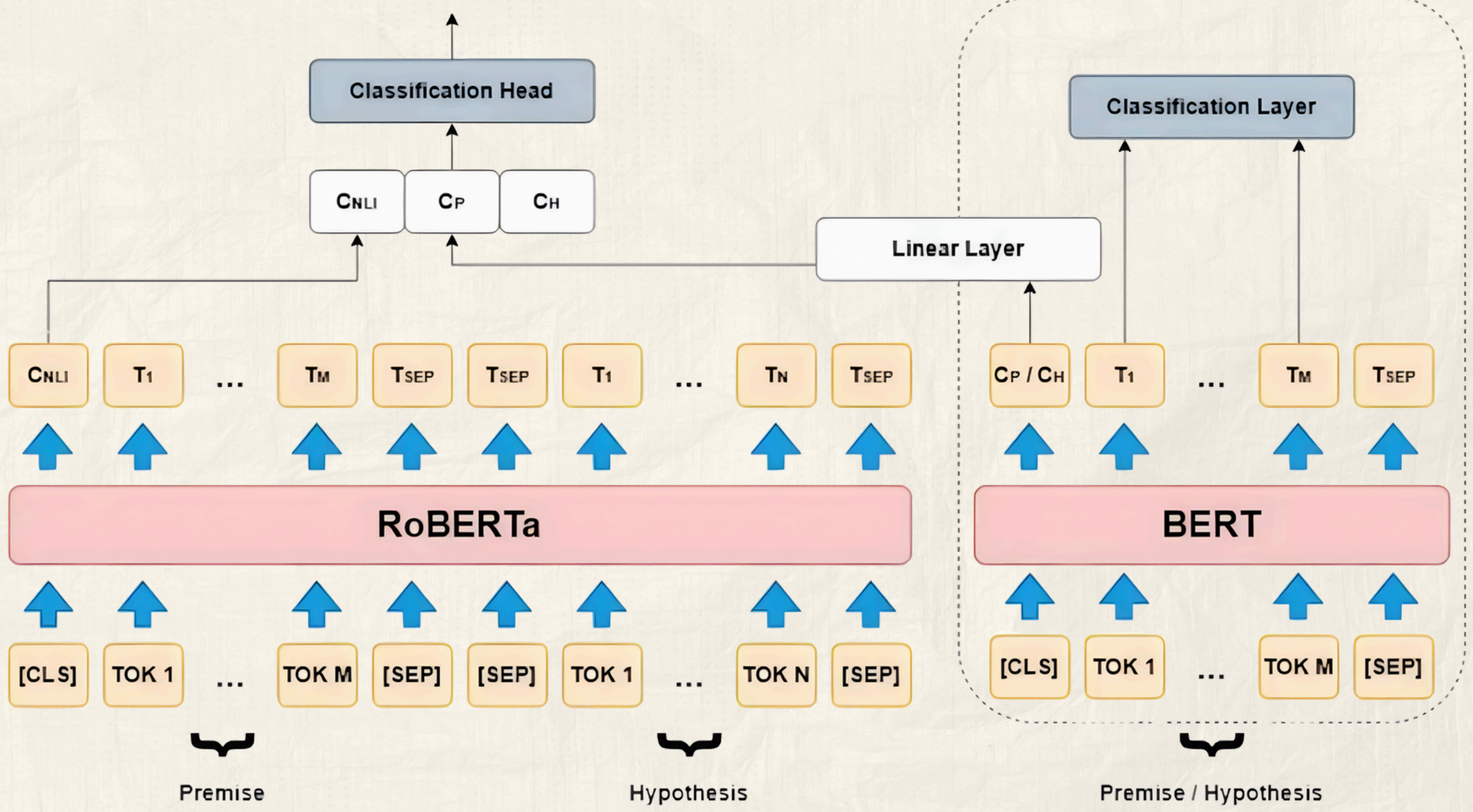
✳ RoBERTa is over BERT

- Trained on 10x more data (160GB diverse text: books, news, web) vs BERT's 16GB
- Longer training with larger batches
- Dynamic masking (patterns change every epoch)
- Removed Next Sentence Prediction (NSP) objective

→ Result: Stronger contextual representations and better downstream performance

✳ Why RoBERTa excels on Tweets

- Tweets are short, noisy, full of slang, emojis, typos, and unusual characters
- Standard tokenizers often mark rare/emoji as [UNK] → loss of information
- RoBERTa uses Byte-Level BPE tokenization: Breaks text into bytes → handles ANY character/emoji perfectly without [UNK]



❌ BERT (WordPiece)

```
"I love this 😍🔥"
→ ["I", "love", "this", [UNK], [UNK]]
```

⚠️ INFORMATION LOSS!

```
"amaziiiing 💯"
→ ["am", "##az", "##i", "##i", [UNK], [UNK]]
```

⚠️ UNKNOWN TOKENS!

✅ RoBERTa (Byte-Level BPE)

```
"I love this 😍🔥"
→ ["I", "Ġlove", "Ġthis", "ĠðŸ˜, ðŸ"¥"]
```

✓ PERFECT ENCODING!

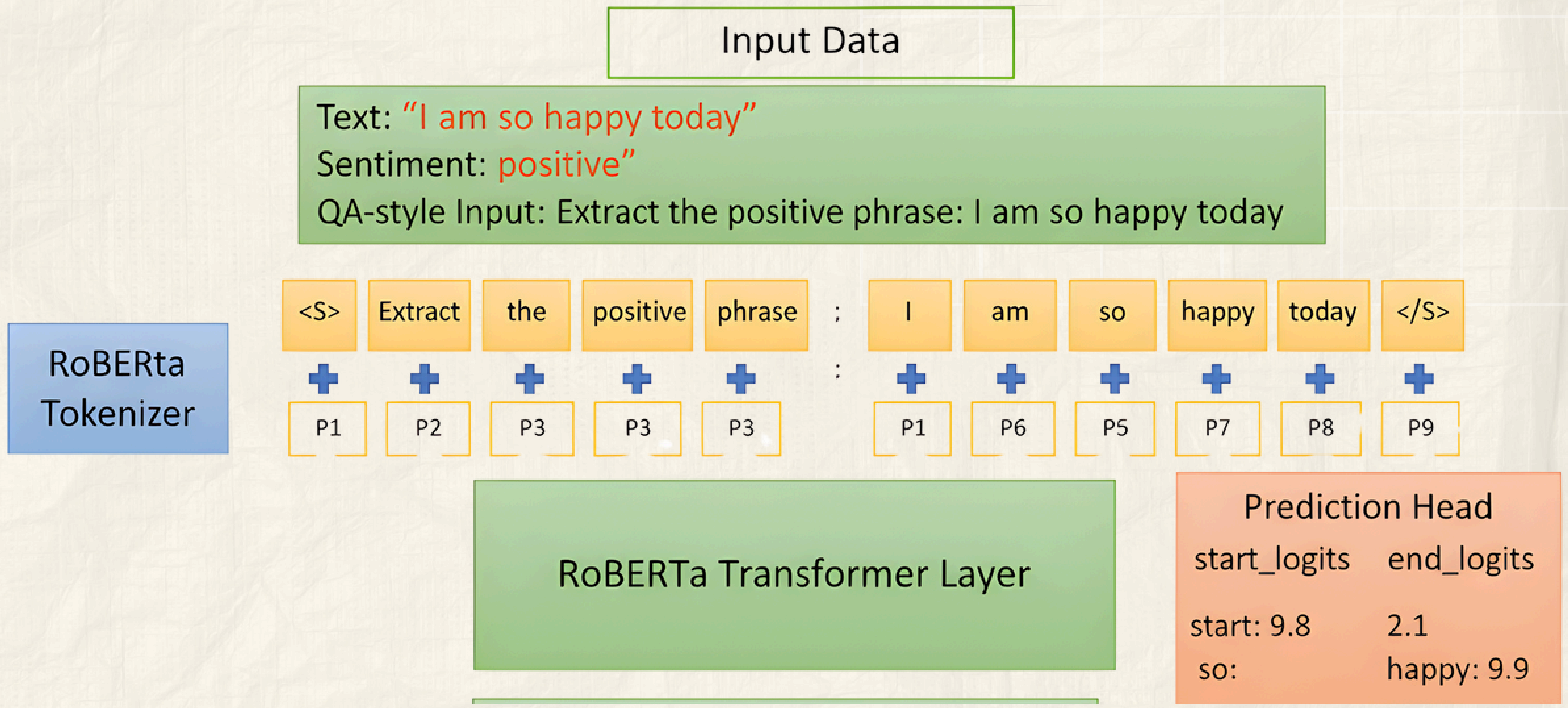
```
"amaziiiing 💯"
→ ["am", "az", "iii", "ing", "Ġðð"¯"]
```

✓ NO INFORMATION LOSS!

HA HA

ATI

05. Architecture & training strategy



1. Input: Convert to QA format "Extract the {sentiment} phrase: " + text
2. Tokenize: Get tokens + offset_mapping (token → char positions)
3. Model Forward: RoBERTa → contextual embeddings → start_logits & end_logits
4. Training: Char mask from selected_text → map to correct start/end positions → CrossEntropy loss
5. Inference: Top-K decoding → best start/end pair by combined score
6. Decode: Use offset_mapping → extract predicted text span
7. Neutral Trick: If neutral → return full text
8. Final: 5-fold ensemble → majority vote

ATI

06. Experimental Results & Evaluation

Jaccard Score

Measures overlap between predicted and true phrases

Word-level Jaccard

Evaluates overlap by words

Character-level Jaccard

Evaluates overlap by characters
→ More precise for short tweets

Evaluation Techniques

✳ Top-K Decoding:

Selects the best start–end token pair among top-K candidates
→ Reduces boundary errors.

✳ Neutral Trick:

For neutral sentiment, return the entire tweet
→ Improves accuracy.

```
 FINAL CV RESULTS
Fold scores: ['0.7831', '0.7901', '0.7876', '0.7834', '0.7870']
Mean ± Std: 0.7863 ± 0.0027
```

ATI

Evaluation Techniques

Fold	Word-level Jaccard	Char-level Jaccard
1	0.7831	0.7726
2	0.7901	0.7861
3	0.7876	0.7878
4	0.7834	0.7834
5	0.7878	0.7870
Mean ± Std	**0.7863 ± 0.0027**	**≈ 0.78 ± 0.002**

Cross-Validation Results

- Neutral: ~0.98 → easiest to predict (often the full tweet).
- Positive/Negative: ~0.52–0.59 → harder due to shorter, diverse spans.

ATI

07.

Inference On New Data

```
Text: Last session of the day  http://twitpic.com/67ezh...
Pred: Last session of the day http://twitpic.com/67ezh
Sentiment: neutral

Text: Shanghai is also really exciting (precisely -- skyscrapers galore). Good tweeps in China:  (SH)  (BJ.
Pred: exciting
Sentiment: positive

Text: Recession hit Veronique Branquinho, she has to quit her company, such a shame!...
Pred: such a shame!
Sentiment: negative

Text: happy bday!...
Pred: happy bday!
Sentiment: positive

Text: http://twitpic.com/4w75p - I like it!!...
Pred: I like it!!
Sentiment: positive
```

Inference Workflow

YEAH!

Preprocessing

Encode tweet + sentiment into QA format.

Prediction

Each fold model outputs a candidate phrase.

Ensemble

Majority vote across folds → stable prediction.

Neutral Trick

If sentiment is neutral → return the full text.

ATI

07. Inference On New Data

```
Text: Last session of the day  http://twitpic.com/67ezh...
Pred: Last session of the day http://twitpic.com/67ezh
Sentiment: neutral

Text: Shanghai is also really exciting (precisely -- skyscrapers galore). Good tweeps in China:  (SH)  (BJ.
Pred: exciting
Sentiment: positive

Text: Recession hit Veronique Branquinho, she has to quit her company, such a shame!...
Pred: such a shame!
Sentiment: negative

Text: happy bday!...
Pred: happy bday!
Sentiment: positive

Text: http://twitpic.com/4w75p - I like it!!...
Pred: I like it!!
Sentiment: positive
```

YEAH!

Output

- Final predictions saved in submission.csv (textID, selected_text).
- Ready for submission and evaluation.

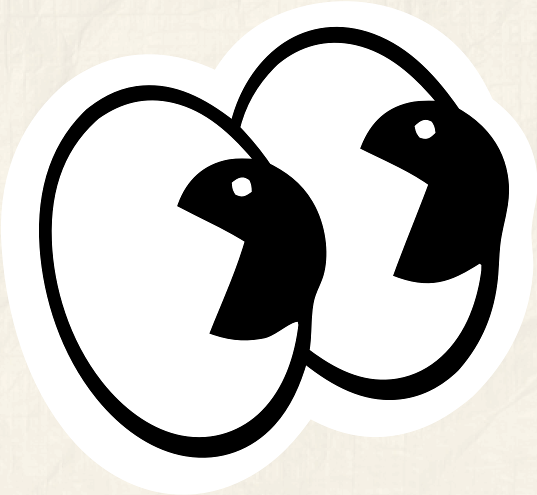
ATI

08. Conclusion

- Framed the task as Extractive Question Answering (QA) → improved interpretability.
- RoBERTa handled noisy tweets effectively with byte-level BPE tokenization.
- Optimization techniques (label smoothing, top-K decoding, neutral trick) boosted robustness.
- Data visualization clarified sentiment distribution and phrase length characteristics
- Jaccard scores demonstrated strong extraction ability.
- Ensemble predictions ensured stable results on unseen test data.

Presentation by **Group 25**

Thank You So Much!



ATI