Sentence Splitting: Encoder vs Decoder
Appendix: Complete Results and Analysis

Archit Rastogi **Ejaz Ahmed**
Sapienza University of Rome

A Complete Encoder Results

Table 1: Encoder Models: Full Metrics

Model	Val F1	OOD F1	Prec	Rec
XGB+IT-BERT	0.960	0.960	0.963	0.957
XGB+MiniLM	0.939	0.946	1.000	0.898
XLM-R+CRF	1.000	0.866	0.770	0.989
BERT-Italian	0.835	0.678	0.663	0.693
BERT-IT+CRF	0.776	0.620	0.594	0.648
XGB+TF-IDF	0.067	0.017	0.036	0.011

Table 2: Encoder Models: Accuracy (note: misleading due to 96.7% class imbalance)

Model	Val Acc	OOD Acc
XGB+Italian-BERT	0.997	0.997
XGB+MiniLM-L12	0.996	0.994
XLM-RoBERTa+CRF	1.000	0.982
BERT-Italian	0.991	0.963
BERT-Italian+CRF	0.987	0.955
XGB+TF-IDF	0.944	0.924

B Complete Decoder Results

Table 3: Decoder Models: Validation Set Performance

Strategy	Model	F1	Prec	Rec
Marker	Llama-3.1-8B	0.587	0.484	0.747
Marker	Llama-3.2-3B	0.310	0.481	0.228
Marker	Llama-3.2-1B	0.101	0.270	0.062
CoT	Llama-3.2-1B	0.470	0.555	0.407
CoT	Llama-3.1-8B	0.443	0.717	0.321
CoT	Llama-3.2-3B	0.068	0.429	0.037
JSON	Llama-3.1-8B	0.331	0.234	0.565
JSON	Qwen3-8B	0.186	0.130	0.330
JSON	Llama-3.2-3B	0.080	0.051	0.191
Iterative	Llama-3.1-8B	0.270	0.400	0.204
Few-Shot	Llama-3.2-3B	0.052	0.044	0.062
Sliding	All models	0.000	0.000	0.000

Table 4: Decoder Models: OOD (Pinocchio) Test Set

Strategy	Model	F1	Prec	Rec
Marker	Llama-3.1-8B	**0.837**	0.756	0.938
Marker	Llama-3.2-3B	0.187	0.909	0.104
Marker	Llama-3.2-1B	0.080	1.000	0.042
CoT	Llama-3.1-8B	0.503	0.766	0.375
CoT	Llama-3.2-1B	0.418	0.737	0.292
CoT	Llama-3.2-3B	0.151	0.800	0.083
JSON	Llama-3.1-8B	0.472	0.431	0.521
JSON	Qwen3-8B	0.288	0.226	0.396
JSON	Llama-3.2-3B	0.167	0.115	0.302
Iterative	Llama-3.1-8B	0.203	0.545	0.125
Few-Shot	Llama-3.2-3B	0.105	0.076	0.167
Sliding	Llama-3.1-8B	0.040	0.667	0.021

C Dataset Statistics

Table 5: Dataset Overview

Split	Tokens	Bounds	%Pos
Train (Manzoni)	74,765	2,447	3.27%
Dev (Manzoni)	9,344	324	3.47%
OOD (Pinocchio)	1,524	96	6.30%

The severe class imbalance (96.7% negative) makes accuracy misleading—a model predicting all non-boundaries achieves 96.7% accuracy but 0% F1. This explains why many decoder strategies show high accuracy but near-zero F1.

D Error Analysis

D.1 Encoder Errors

XLM-RoBERTa+CRF achieves perfect validation F1 (1.000) but drops to 0.866 on OOD, indicating overfitting to Manzoni's specific patterns. The CRF layer learns transition probabilities that don't generalize (e.g., specific dialogue markers unique to Manzoni).

XGBoost models show remarkable consistency (0.960 validation and OOD), suggesting that

embedding-based features capture more transferable patterns than fine-tuned weights.

TF-IDF failure (0.017 F1) confirms that bag-of-words features cannot capture the contextual dependencies required for sentence boundary detection.

D.2 Decoder Errors

Systematic under-prediction: Most decoder strategies have higher precision than recall, meaning they miss many true boundaries. This is particularly severe for sliding window (0.021 recall) and few-shot (0.021 recall).

Output parsing failures: JSON and iterative strategies suffer from malformed outputs. When the model generates invalid JSON or misaligned markers, the entire chunk fails.

Model size effects: Llama-3.1-8B consistently outperforms smaller models (1B, 3B), suggesting that token-level classification via prompting requires larger model capacity.

D.3 Recurrent Error Patterns

Both approaches struggle with:

- **Abbreviations**: Periods in abbreviations (e.g., "sig.", "dott.") are sometimes misclassified as sentence boundaries.
- **Dialogue**: Italian literary dialogue uses unconventional punctuation that differs from modern conventions.
- **Ellipses**: Multiple consecutive periods create ambiguity.

Mitigation suggestions: (1) Add abbreviation lists as features for encoders; (2) Include more dialogue examples in decoder few-shot prompts; (3) Pre-process ellipses into single tokens.

E Strategy Descriptions

Strategy 1 - Sliding Window: Local context window around each punctuation token, independent YES/NO decision. Failed due to lack of global context.

Strategy 2 - Next-Token Probability: Uses token probabilities to infer sentence starts. Requires local model access; essentially failed.

Strategy 3 - Marker Insertion: Model rewrites text inserting `<EOS>` markers. Best decoder strategy due to simple, robust output format.

Strategy 4 - Structured JSON: Model outputs boundary indices in JSON. Brittle to formatting errors.

Strategy 5 - Few-Shot Hard: Curated edge cases as examples. Insufficient for small models to generalize.

Strategy 6 - Chain-of-Thought: Explicit reasoning before prediction. Second-best strategy; reasoning adds reliability but also verbosity.

Strategy 7 - Iterative Refinement: Two-pass correction. Marginal improvement over single-pass.