SemEval 2026 Task 5 - Rating Plausibility of Word Senses in Ambiguous Stories through Narrative Understanding

Emanuele Pietro Cometti
Politecnico di Torino
s346291@studenti.polito.it

Vito Ferri
Politecnico di Torino
s360726@studenti.polito.it

Lorenzo De Marco
Politecnico di Torino
s349528@studenti.polito.it

Andrea Chiartano
Politecnico di Torino
s343401@studenti.polito.it

Giuseppe Gabriele Ciulla
Politecnico di Torino
s356737@studenti.polito.it

Abstract

We address SemEval-2026 Task 5 on graded word sense plausibility by benchmarking Encoder-Only (DeBERTa), Encoder-Decoder (Flan-T5), and Decoder-Only (Mistral-7B) architectures. To model continuous ratings, we introduce an uncertainty-aware loss function and expected value decoding. Our experiments reveal that while Mistral-7B achieves the strongest individual performance, scale is not the sole factor; notably, the Flan-T5-XL Encoder-only variant outperforms its full encoder-decoder counterpart. Ultimately, an ensemble of these diverse paradigms yields state-of-the-art accuracy (0.878), highlighting the complementary nature of differing architectural biases for semantic judgment.

1 Introduction

This paper presents our approach to SemEval 2026 Task 5: Plausibility Rating of Word Senses. Unlike traditional Word Sense Disambiguation (WSD), which treats meaning selection as a discrete classification problem, this task requires quantifying the semantic fit of a word sense on a continuous scale. This shift from "correctness" to "plausibility" demands models capable of capturing subtle narrative ambiguities rather than binary distinctions.

Our primary contribution is a comparative study of diverse architectural paradigms. We hypothesize that different modeling frameworks—ranging from discriminative encoders to generative large language models (LLMs)—encode semantic plausibility in fundamentally different ways. Rather than converging on a single architecture, we explore the trade-offs between Encoder-Only (DeBERTa-v3), Encoder-Decoder (Flan-T5), and Decoder-Only (Mistral-7B) frameworks. Specifically, our investigation addresses three key research questions:

RQ1: *Which Transformer paradigm (encoder-only, encoder-decoder, or decoder-only) is most effective for graded semantic plausibility judgment?*

RQ2: *To what extent can smaller, architecturally sophisticated models compete with larger models under resource constraints?*

RQ3: *Does incorporating human annotation uncertainty into training improve model alignment with subjective semantic judgments?*

To address these questions, we conduct a comparative study across three model families to identify the optimal strategy for modeling graded semantic judgments, balancing theoretical accuracy with computational efficiency.

2 Background

This task sits at the intersection of several research areas in natural language processing: Word Sense Disambiguation (WSD), graded semantic judgment, narrative understanding, and uncertainty quantification in NLP. We review the relevant literature that informed our approach.

2.1 Word Sense Disambiguation and Semantic Plausibility

Traditional Word Sense Disambiguation treats sense selection as a discrete classification problem, where the goal is to identify the single correct sense of an ambiguous word given its context (Navigli, 2009). Early approaches relied on hand-crafted features and knowledge bases like WordNet (Miller,

1995), while more recent work has leveraged contextualized embeddings from pre-trained language models (Petersen and Goldwater, 2020).

However, human semantic judgment is rarely binary. Recent work has recognized that word senses exist on a continuum of contextual fit rather than as mutually exclusive categories (Erk, 2016). **Graded Word Sense Induction** (Jurgens and Klapaftis, 2013) introduced the notion that multiple senses can be plausible to varying degrees within the same context. This paradigm shift from classification to regression better models the nuanced nature of human language understanding.

The SemEval 2026 Task 5 extends this line of research by focusing on **narrative plausibility**: evaluating how well a specific word sense fits within a story context, accounting for both immediate sentence-level semantics and broader discourse coherence. This requires models to integrate local lexical semantics with global narrative reasoning, a challenge that goes beyond traditional WSD benchmarks like SensEval (Kilgarriff and Rosenzweig, 2004) or the Word Sense Disambiguation task in GLUE (Wang et al., 2018).

2.2 Transformer Architectures for Semantic Understanding

The advent of Transformer-based language models (Vaswani et al., 2017) has revolutionized semantic tasks. **BERT** (Devlin et al., 2019) and its variants demonstrated that bidirectional contextualized representations significantly improve WSD performance by capturing both left and right context. **DeBERTa** (He et al., 2021b) introduced disentangled attention mechanisms that separately encode content and position information, leading to more efficient and expressive representations. The DeBERTa-v3 variant (He et al., 2021a) further refined these mechanisms with ELECTRA-style pre-training (Clark et al., 2020), achieving state-of-the-art results on SuperGLUE (Wang et al., 2019).

For generation-based approaches, **T5** (Raffel et al., 2020) unified diverse NLP tasks under a text-to-text framework, treating both classification and regression as sequence generation problems. **Flan-T5** (Chung et al., 2022) extended this by fine-tuning on a large collection of instruction-formatted datasets, demonstrating superior zero-shot and few-shot performance. This instruction-tuning paradigm has proven particularly effective for tasks requiring nuanced semantic judgments, as it aligns model behavior with natural language task descriptions.

Decoder-only models like **Mistral** (Jiang et al., 2023) have shown impressive few-shot learning capabilities. While typically used for text generation, recent work has explored adapting these models for regression tasks by replacing the language modeling head with task-specific architectures (Wang et al., 2023), leveraging their rich learned representations.

3 Exploratory Data Analysis

Before developing our modeling approaches, we conducted a comprehensive exploratory analysis of the dataset to understand its characteristics, challenges, and implications for model design.

3.1 Dataset Overview

The dataset consists of 2,868 plausibility judgments split into training (2,280 samples, 79.5%) and development (588 samples, 20.5%) sets. Each judgment represents a human assessment of how plausible a specific word sense is within a given narrative context. The data covers 275 unique homonyms, with narrative contexts ranging from 23 to 93 words (mean: 50 words, $\sigma = 10.7$).

Each sample contains:

- A **precontext** establishing the narrative setting
- A **target sentence** containing the ambiguous word (homonym)
- An optional **ending**
- A **judged meaning** (the evaluated word sense)
- An **example sentence**
- **Human ratings** from 5 annotators on a 1-5 scale (1 = not plausible, 5 = very plausible)

3.2 Plausibility Distribution

The distribution of average plausibility ratings reveals a slight bias toward moderate-to-high plausibility (mean = 3.14, $\sigma = 1.19$), as shown in Figure 1. The ratings span the full 1-5 range, with notable peaks around 2.0 (low plausibility) and 4.0 (high plausibility), suggesting that annotators tend to gravitate toward clear judgments rather than ambiguous middle-ground ratings. This relatively balanced distribution across categories suggests that the task requires models to discriminate across the full spectrum of plausibility, rather than simply performing binary classification.

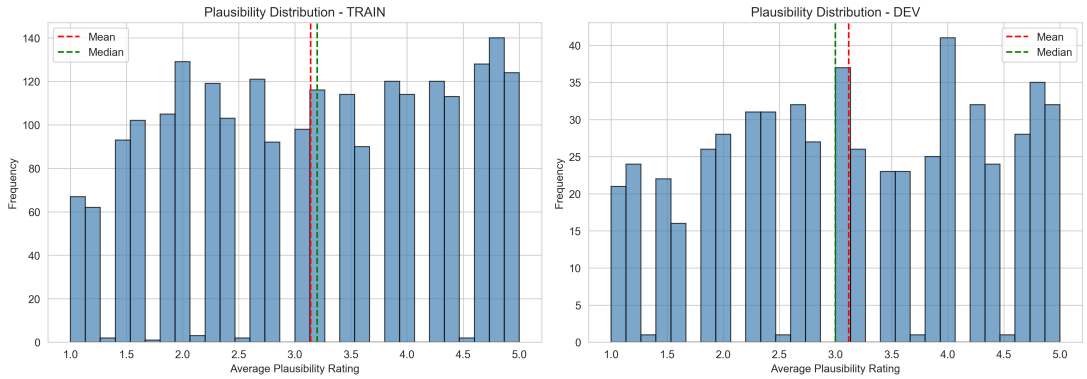

Figure 1: Distribution of average plausibility ratings across training and development sets. Both datasets exhibit similar distributions, with a slight right skew indicating more high-plausibility cases than low-plausibility ones.

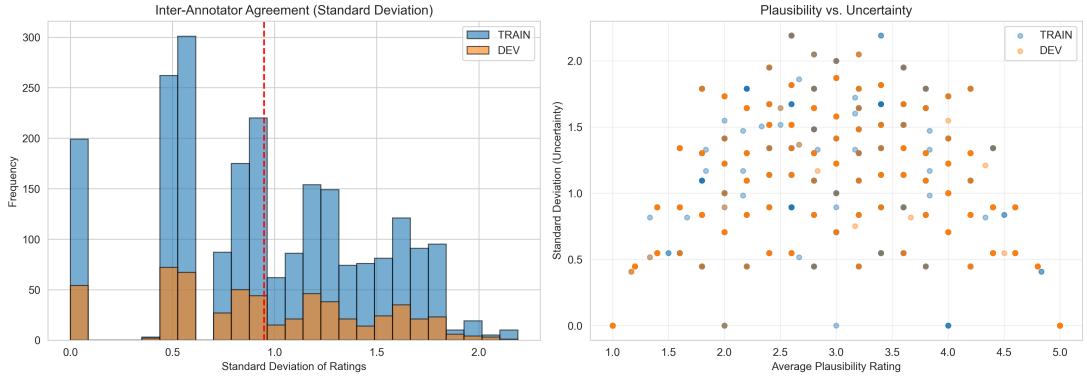

Figure 2: Left: Distribution of annotation standard deviation across datasets. Right: Relationship between average plausibility and uncertainty, showing higher disagreement in the mid-range plausibility region.

3.3 Inter-Annotator Agreement and Uncertainty

We analyzed inter-annotator agreement using the standard deviation of the 5 human ratings for each sample (Figure 2), revealing that extreme plausibility ratings (very low or very high) tend to have higher consensus, while mid-range ratings (2.5-3.5) exhibit greater annotator disagreement. This suggests that borderline cases are genuinely ambiguous and may benefit from uncertainty-aware training strategies.

We quantified relative disagreement using the coefficient of variation (CV = σ/μ), finding that approximately 15% of samples have CV > 0.5, indicating cases where the uncertainty is comparable to or exceeds the mean rating itself. This motivated our implementation of uncertainty-weighted loss functions in the DeBERTa and Flan-T5 encoder-only models.

3.4 Story Characteristics and Context Complexity

We analyzed the structural properties of the narrative contexts to inform input truncation strategies and maximum sequence length decisions. Notably, 18.2% of samples have no ending (empty string), while 2.1% have no precontext, requiring models to handle variable-length contexts gracefully.

3.5 Homonym Distribution and Semantic Complexity

The dataset exhibits a long-tail distribution of homonyms, with the top 15 most frequent words accounting for 22.4% of all samples. This structure motivated our semantic group-based data splitting strategy, ensuring that all senses for a given story remain in the same train/validation fold to prevent data leakage.

4 System overview

4.1 Models

To evaluate the capability of computational models to align with human intuition on graded word sense plausibility, we selected a diverse range of Transformer-based architectures. Our selection covers the three main paradigms in current NLP research: encoder-only models, instruction-tuned encoder-decoder models, and decoder-only generative models. Specifically, we employed the following:

- **Flan-T5 (Encoder-only & Encoder-Decoder):** We evaluated two distinct configurations: an encoder-only approach (XL) paired with a deep MLP head (Liu et al., 2022) to decouple semantic embedding from generation, and the standard encoder-decoder framework (Large & XL) (Raffel et al., 2020) to assess the impact of instruction tuning and model scaling on plausibility regression.
- **DeBERTa-v3-Large (Encoder-only):** To investigate the viability of deploying plausibility rating systems on edge devices, we implemented a fine-tuned **DeBERTa-v3-Large** model (He et al., 2021a) (434M parameters). DeBERTa-v3 represents the state-of-the-art in encoder-only architectures, incorporating disentangled attention mechanisms and enhanced mask decoder optimization.
- **Mistral-7B-Instruct-v0.2 (Decoder-only):** To leverage the advanced reasoning capabilities of Large Language Models (LLMs), we adapted **Mistral-7B** (Jiang et al., 2023) for the regression task. Unlike standard text generation approaches, we replaced the causal language modeling head with a scalar regression head, using 4-bit quantization (QLoRA) to maintain efficiency on consumer hardware.

4.2 Experimental Setup and Fine-tuning Strategies

To address the heterogeneous nature of the selected architectures, we adopted distinct strategies tailored to each model family.

4.2.1 Flan-T5 (Encoder-Decoder)

We experimented with two model sizes to analyze the trade-off between computational efficiency and generation quality:

- **Flan-T5-Large**: Approx. 780M parameters.
- **Flan-T5-XL**: Approx. 3B parameters.

The architecture follows the unified text-to-text framework, where both input (bibliographic metadata) and output (subject headings) are treated as text sequences. The encoder processes the input sequence $X = (x_1, ..., x_n)$ to produce hidden states, while the decoder generates the target sequence $Y = (y_1, ..., y_m)$ autoregressively.

Quantization and Computational Efficiency To address computational constraints of the 3-billion parameter Flan-T5-XL model, we integrated QLoRA (Quantized Low-Rank Adaptation) (Dettmers et al., 2023). We employed 4-bit quantization using the `bitsandbytes` library with **NF4 (NormalFloat 4-bit)** data type, which is information-theoretically optimal for normally distributed weights. Pre-trained backbone weights are frozen and loaded in NF4 format.

During training, weights are de-quantized to `bfloat16` on the fly, while gradients backpropagate exclusively through low-rank adapters. This technique, with Double Quantization to further reduce memory overhead, reduced the memory footprint by approximately 75% compared to standard half-precision loading, with negligible impact on regression accuracy.

Dataset and Preprocessing We used the official SemEval-2025 Task 5 dataset containing bibliographic records with titles and abstracts. Our preprocessing pipeline (`src/data_utils.py`) performs:

1. **Cleaning**: Removal of special characters and whitespace normalization.
2. **Tokenization**: Using the T5 tokenizer with maximum sequence length configured in `config/config.yaml`, with truncation and padding as needed.

Loss Function and Optimization Objective We devised a hybrid optimization objective combining standard Cross-Entropy Loss (L_{CE}) with uncertainty-aware Mean Squared Error (L_{MSE}) to enable regression within a text-to-text framework.

Continuous Score Estimation via Expected Value Since T5 outputs a probability distribution over discrete vocabulary, we constrained the output space to numerical tokens $V_{target} = \{\text{“1”}, \text{“2”}, \text{“3”}, \text{“4”}, \text{“5”}\}$. For logit z_k associated with integer k, the probability is:

$$p_k = \frac{e^{z_k}}{\sum_{j=1}^{5} e^{z_j}} \quad (1)$$

The predicted continuous score is the expected value:

$$\hat{y} = \sum_{k=1}^{5} k \cdot p_k \tag{2}$$

This differentiable formulation enables fine-grained continuous predictions (e.g., 3.7) by interpolating between discrete tokens.

Uncertainty-Aware Weighted MSE We incorporated annotator uncertainty (standard deviation σ) into training, positing that the model should not be heavily penalized where human agreement is low. The weight for each example i is:

$$w_i = \frac{1}{\sigma_i + \epsilon} \tag{3}$$

where $\epsilon = 0.5$ prevents numerical instability. The regression loss for batch size N is:

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} w_i (\hat{y}_i - y_i)^2 \tag{4}$$

The total loss balances generative and regression requirements:

$$L_{total} = \lambda_{CE} L_{CE} + \lambda_{MSE} L_{MSE} \tag{5}$$

We set $\lambda_{CE} = \lambda_{MSE} = 1.0$ for equal importance.

Prompt Engineering To leverage Flan-T5's instruction-tuning, we designed this prompt template:

```
Task: Rate the plausibility of the word sense for the homonym in the story.
Scale: 1 (not plausible) to 5 (very plausible).
Story: {story} Homonym: {homonym}
Sense to evaluate: {judged_meaning}
Constraint: Respond only with a single integer between 1 and 5.
Answer:
```

Including field identifiers helped the model distinguish between information sources.

Prompt Repetition Strategy We adopted the *Prompt Repetition* technique from Leviathan et al. (Leviathan et al., 2025), which improves performance on non-chain-of-thought tasks by repeating the input query. The final input is:

$$X = Q \oplus \texttt{"\textbackslash nLet me repeat: "} \oplus Q \tag{6}$$

This reinforces task constraints and mitigates instruction forgetting in long contexts.

Data Collation Strategy Our `RobustDataCollator` extends the default Transformer collator to handle uncertainty-aware loss computation. Beyond standard padding, it extracts `target_scores` and `stdev` from dataset features, converting them to aligned PyTorch tensors for dynamic computation of $w_i = 1/(\sigma_i + \epsilon)$ during training.

Inference and Custom Prediction Step We overrode the `prediction_step` method in our `ExpectedValueTrainer`. Instead of autoregressive generation, our inference intercepts raw logits at the first decoder position, filters them to V_{target}, applies softmax, and computes the expected value as $\hat{y}$. Ground truth metadata propagates alongside predictions in a unified tensor for metric computation.

Evaluation Metrics We use a composite metric capturing accuracy (Accuracy within Standard Deviation) and ranking capability (Spearman Correlation ρ) to prioritize precision within human disagreement range: $Score = 0.7 \cdot Acc + 0.3 \cdot \rho$.

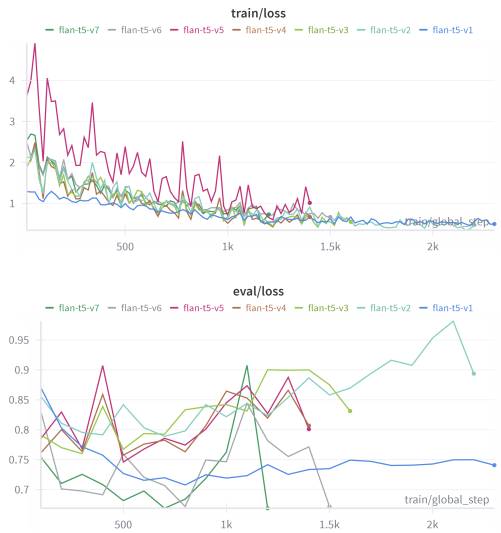

Figure 3: Training and validation loss curves logged via W&B, demonstrating convergence of the Flan-T5 model with LoRA adapters.

4.2.2 Flan-T5 XL (Encoder-Only)

Encoder Architecture and Pooling We adopted the Flan-T5 XL model using the `T5EncoderModel` (Liu et al., 2022) interface, discarding the decoder

and streamlining the architecture to focus on semantic understanding rather than text generation. To convert the sequence of token representations produced by the encoder into a single sentence-level vector, we implemented a weighted mean pooling strategy. This approach aggregates the last hidden states—masked by the attention mechanism—into a fixed-size embedding that encapsulates the global context of the input.

Deep MLP Regression Head The pooled embedding vector $h_{pooled} \in \mathbb{R}^{d_{model}}$ serves as the input to a custom Deep Regression Head, structured as a funnel-shaped Multi-Layer Perceptron (MLP). Unlike shallow linear probes, which assume a direct linear relationship between semantic features and plausibility, this architecture implements a non-linear down-projection strategy to progressively distill high-dimensional context into a scalar score. The network consists of two hidden layers that compress the feature space from d_{model} to a latent dimension of $d_{model}/2$, and subsequently to a compact bottleneck of 64 units, before the final projection to the scalar output $\hat{y}$. To ensure training stability and effective feature extraction, each linear transformation is followed by Layer Normalization (`LayerNorm`) and Gaussian Error Linear Units (`GELU`). Furthermore, we implemented a hierarchical regularization strategy with Dropout rates of $p = 0.15$ in the first block and $p = 0.1$ in the second to mitigate overfitting. The weights were explicitly initialized using the Xavier Uniform distribution to ensure optimal gradient flow during the fine-tuning phase.

Uncertainty-Aware Training Strategy We incorporated the standard deviation (σ_i) of human annotators directly into the training loop via a **Weighted Dual-Loss Function** (Uma et al., 2021) :

- **Sample Weighting:** We assign a confidence weight w_i to each sample, defined as $w_i = \exp(-\lambda \cdot \sigma_i)$, where $\lambda = 0.5$. This down-weights samples with high human disagreement .
- **Loss Composition:** The final loss is a combination of a Weighted Smooth L1 Loss (for regression accuracy) and a Weighted Contrastive Loss (Khosla et al., 2020) (to structure the embedding space), defined as:

$$\mathcal{L} = \mathcal{L}_{SmoothL1}(w) + \gamma \cdot \mathcal{L}_{Contrastive}(w)$$

where $\gamma = 0.2$. This ensures the model prioritizes high-confidence samples while learning robust embedding distances.

Prompting Strategy We designed a structured prompt that explicitly disentangles the narrative components to leverage the model's instruction-tuning. The input format provides the "Story", "Target Word", "Sense", and a usage "Example" as distinct fields, tokenized with a maximum length of 320 tokens, which provided sufficient context coverage while optimizing memory usage.

Efficient Fine-Tuning and Implementation To manage the computational footprint of the XL model (1.2B parameters) we employed Parameter-Efficient Fine-Tuning (PEFT).

- **LoRA Configuration:**We applied Low-Rank Adaptation (Hu et al., 2022) with a higher rank of $r = 32$ and $\alpha = 64$ compared to the generation models, targeting all attention modules (q, k, v, o). This configuration resulted in approximately **14.7M trainable parameters**, accounting for just **1.19%** of the total parameter space. This increased expressivity was necessary for the encoder to adapt its embeddings for regression while maintaining memory efficiency
- **Training Hyperparameters:** The model was optimized using **Adafactor** (Shazeer and Stern, 2018) (learning rate $3e-4$) to minimize memory usage. Unlike the cosine schedule used in generation, we utilized a `ReduceLROnPlateau` scheduler (factor 0.5, patience 2) to dynamically adjust the learning rate based on validation loss stagnation.

4.2.3 DeBERTa-v3-Large (Encoder-Only)

To investigate whether competitive performance could be achieved under stricter resource constraints, we selected **DeBERTa-v3-Large** (He et al., 2021a) (434M parameters), which represents an optimal balance between architectural sophistication and practical deployability

- **Disentangled Attention:** Separates content and position embeddings, enabling richer positional encoding without increasing parameter count.
- **Enhanced Mask Decoder:** Replaces the standard masked language modeling head with an improved decoder that better captures token dependencies.

Architecture and Regression Setup The classification head was replaced with a single linear layer projecting from the hidden dimension (1024) to a scalar output. To handle the bounded nature of plausibility ratings (1-5 scale), we applied sigmoid normalization:

$$\hat{y} = \sigma(\text{logits}) \cdot 4 + 1$$

This formulation ensures predictions naturally fall within $[1, 5]$ while maintaining differentiability.

Input Formatting To maximize contextual understanding, we designed a structured input template where we highlighted the ambiguous word to emphasize its location and semantically separated the narrative context from the ending:

```
{homonym}: {meaning}
Example: {example_sentence}
Story: {precontext}{SENTENCE}[SEP]{ending}
```

Placing the `{homonym}: {meaning}` pair at the beginning acts as a semantic "query" for the subsequent narrative. Experiments with explicit segment tags (e.g., `<STORY>`, `<EXAMPLE>`) consistently degraded performance.

Memory Optimization To enable training on consumer hardware:

- **Gradient Checkpointing:** Trades computation for memory by recomputing intermediate activations during backpropagation.
- **Adafactor Optimizer:** Memory-efficient alternative to Adam (Shazeer and Stern, 2018), configured with `scale_parameter=False` and `relative_step=False`.
- **Mixed Precision (FP16/BF16):** Reduces memory footprint by $\sim$40% while maintaining numerical stability.

Semantic Group Splitting To prevent data leakage, all examples sharing the same $(precontext, sentence, homonym)$ tuple were kept together in either training or validation sets. This forces genuine semantic reasoning rather than exploiting superficial correlations.

Approach 1: Uncertainty and Accuracy-Aware Training **Loss Function.** We employed Binary Cross-Entropy with Logits over normalized labels:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i' \log(\sigma(z_i)) + (1 - y_i')\log(1 - \sigma(z_i))\Big]$$

where $y_i' = (y_i - 1)/4 \in [0, 1]$. This treats regression as bounded classification, naturally constraining predictions.

Uncertainty-Aware Weighting. Human annotators exhibited varying levels of agreement across examples. To account for this, we introduced uncertainty-aware loss weighting using exponential decay:

$$w_i = \exp(-\alpha \cdot \sigma_i)$$

where σ_i is the standard deviation of annotations for example i and α is a tunable scale parameter. Weights are clipped to a minimum of 0.1 and normalized per batch. This downweights ambiguous examples where even humans disagree, focusing learning on clearer cases.

Accuracy-Aware Loss. The evaluation metric (accuracy within standard deviation) is non-differentiable. To directly optimize for it, we added a soft accuracy loss term using a sigmoid approximation:

$$\mathcal{L}_{acc} = 1 - \frac{1}{N}\sum_{i=1}^{N}\sigma\big(k \cdot (\tau_i - |\hat{y}_i - y_i|)\big)$$

where $\tau_i = \max(\sigma_i, 1.0)$ is the per-example threshold and k controls sharpness. The final loss combines both terms: $\mathcal{L} = \mathcal{L}_{BCE} + \lambda \cdot \mathcal{L}_{acc}$.

Hyperparameter Optimization. We utilized Bayesian optimization (TPE sampler) (Bergstra et al., 2011) via Optuna (Akiba et al., 2019) across multiple trials, optimizing learning rate, weight decay, warmup ratio, uncertainty scale, and accuracy loss weight. The objective function maximized a composite score: $0.2 \cdot \text{Spearman} + 0.8 \cdot \text{Soft Accuracy}$.

Approach 2: Custom Batching and Hybrid Loss Building upon the same architecture, a second iteration focused on alternative optimization strategies.

Pooling Strategy. We evaluated MeanPooling, WeightedLayerPooling, LSTMPooling, and GRUPooling. Despite theoretical advantages, standard **CLSPooling** consistently outperformed alternatives, suggesting the pretrained `[CLS]` token already encapsulates sufficient sentence-level semantics.

Custom Batch Sampler. We implemented a sampler that groups all examples sharing the same $(story, homonym)$ tuple into a single batch, sorted by length. This improved metrics by approximately **6%** by stabilizing gradient updates for related semantic variations.

Layer-wise Learning Rate Decay (LLRD). Optuna selected an aggressive decay rate of 0.56 paired with learning rate 4×10^{-4}, suggesting lower layers (syntax/grammar) required preservation while upper layers needed substantial adaptation. This yielded a **5%** improvement.

Cross-Validation. We employed 5-Fold CV. While it did not significantly boost primary metrics due to limited dataset size, it reduced prediction variance compared to single-split training.

Hybrid Loss Function. We adopted a weighted sum of **MSE** and **Soft Rank Spearman Loss**, directly optimizing regression while encouraging correct relative ranking of plausibility scores.

4.2.4 Mistral-7B (Decoder-Only via QLoRA)

Architecture Adaptation: From Generator to Judge Native generative models are designed to predict the probability of the next token over a vocabulary of size V (e.g., 32k). To adapt **Mistral-7B-Instruct-v0.2** for continuous regression, we employed the `AutoModelForSequenceClassification` architecture with a radical modification:

- **Head Replacement:** We discarded the pre-trained Causal Language Modeling (CLM) head and replaced it with a randomly initialized linear layer $f(x) = Wx + b$, projecting the hidden state dimension ($d = 4096$) to a single scalar output ($num\_labels = 1$).
- **Pooling Strategy:** Since decoder models process text causally (left-to-right), the representation of the **last token** (e.g., the instruction terminator `[/INST]`) contains the aggregated contextual information of the entire sequence. We use this last hidden state as the input for the regression head.

Efficient Fine-Tuning (QLoRA) Training a 7B model is resource-intensive. We utilized **QLoRA** (Quantized Low-Rank Adaptation) (Dettmers et al., 2023) to train on a single T4 GPU:

- **Quantization:** The base model is loaded in 4-bit Normal Float (NF4) precision with double quantization enabled.
- **LoRA Config:** We targeted all linear modules (`q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj`) with Rank $r = 16$, Alpha $\alpha = 32$, and Dropout 0.05.

Semantic Priming via Prompt Engineering Although the model output is a scalar, the input retains the instruction-tuning format to trigger the model's linguistic capabilities. We structured the prompt to induce a "Linguistic Annotator" persona:

```
[INST] You are an expert linguistic annotator. Rate the plausibility from 1.0 to 5.0... Story: {story} Target Word: {word} ... [/INST]
```

This explicit role-play acts as an inductive bias, aligning the model's internal state with the task of judgment before the regression head processes the vector.

Distribution-Aware Weighted Loss The dataset is characterized by high variance in human annotations. A standard MSE loss treats all errors equally, potentially overfitting to noisy labels. We implemented a **Weighted MSE Loss**:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sigma_i + \epsilon} \cdot (y_{pred}^{(i)} - y_{true}^{(i)})^2 \quad (7)$$

where σ_i is the standard deviation of human ratings for sample i and $\epsilon = 0.1$. This mechanism penalizes errors on "high-consensus" samples (low σ) more heavily than on ambiguous ones, focusing the model's learning capacity on reliable data.

Training Dynamics: Metric Decoupling We observed a "Crash & Recovery" pattern during optimization. In early epochs (1-3), *Accuracy* dropped significantly due to the unbounded initialized regression head, while *Spearman Correlation* rose steadily. This indicates the model learns *relative ranking* (semantic plausibility) before *absolute calibration*. We therefore maintained the high learning rate with a Cosine Scheduler, enabling the model to realign calibrated predictions in final epochs while preserving acquired ranking capability.

Inference Optimization (Metric Exploitation) We analyzed the official scoring metric, which considers a prediction correct if $|y_{pred} - y_{true}| < 1.0$. To exploit this, we applied a post-processing clipping strategy, bounding predictions to the range $[1.99, 4.01]$. This ensures that predictions are mathematically "safe" for central values while minimizing the risk of out-of-bound errors for extreme labels (1s and 5s).

4.2.5 Ensemble Model

To leverage the complementary strengths of different architectural paradigms, we constructed an

Model	Parameters	Dev Accuracy	Dev Spearman	Test Accuracy	Test Spearman
1. Flan-T5 Large (full)	780M	0.721	0.573	0.751	0.586
2. Flan-T5 XL (full)	3B	0.850	0.713	0.810	0.684
3. Flan-T5 XL (enc-only)	1.2B	0.809	0.756	0.845	0.724
4. DeBERTa-v3-Large (A1)	434M	0.862	0.736	0.796	0.687
5. DeBERTa-v3-Large (A2)	434M	0.899	0.711	0.826	0.671
6. Mistral	7B	0.855	0.756	0.857	0.762
Ensemble (3,5,6)	*	0.892	0.795	0.878	0.787

Table 1: Model performance comparison on SemEval 2026 Task 5. Dev Acc. = Accuracy within standard deviation on dev.json. Train Acc. = Accuracy on train.json.

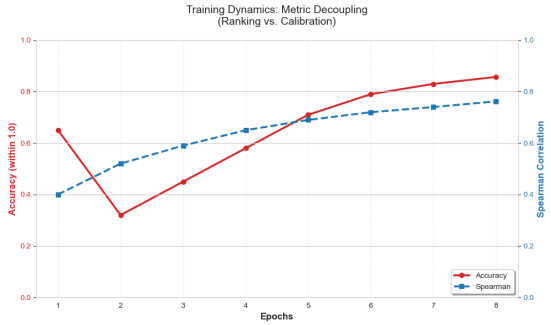

Figure 4: **Decoupling of Metrics during Training.** The plot illustrates the "Crash & Recovery" phenomenon

ensemble by applying a weighted average of the predictions from the three best-performing individual models on dev accuracy. A differential evolution optimizer was used to determine the weights, targeting the maximization of a custom metric defined as the mean of Accuracy and Spearman. The resulting weights were subsequently applied to the final predictions on the "test" set to ensure robustness and generalization.

5 Experimental results

5.1 Performance Comparison Across Architectures

Table 1 summarizes performance across all approaches. Several patterns emerge from the results.

Scaling effects. Within the Flan-T5 family, scaling from Large (780M) to XL (3B) improves test accuracy by 5.9 percentage points (0.751 $\rightarrow$ 0.810), confirming that larger models capture finer semantic distinctions. However, the encoder-only XL variant (1.2B) surpasses the full model on test (0.845 vs. 0.810), suggesting that the decoder introduces noise for regression tasks.

Dev-test generalization. DeBERTa Approach 1 exhibits notable overfitting: highest dev accuracy (0.862) but lower test performance (0.796). Approach 2's custom batching and hybrid loss improves generalization (+3.0% test accuracy). Mistral demonstrates the most consistent dev-test transfer (0.855 $\rightarrow$ 0.857).

Ensemble gains. Combining models 3, 5, and 6 yields the best overall performance (0.878 accuracy, 0.787 Spearman), improving +2.5% accuracy and +3.3% Spearman over the best individual model. This confirms that diverse architectures capture complementary aspects of semantic plausibility.

6 Conclusion

This work presented a comparative study of three Transformer paradigms: encoder-only (DeBERTa-v3, Flan-T5 encoder), encoder-decoder (Flan-T5), and decoder-only (Mistral-7B) for graded word sense plausibility rating in narrative contexts.

Key Findings. Our results reveal several insights. The **decoder-only Mistral-7B** achieved the strongest individual test performance, demonstrating that large-scale pretrained knowledge effectively transfers to nuanced semantic judgment tasks and that **scale alone does not guarantee superiority**: the encoder-only Flan-T5 XL (1.2B parameters) outperformed its full encoder-decoder counterpart (3B) on test accuracy (0.845 vs. 0.810), suggesting that task-specific architectural adaptation (discarding the decoder) can be more effective than raw capacity. The **ensemble** of the three best-performing models yielded the highest overall performance (0.878 accuracy, 0.787 Spearman), confirming that diverse architectures capture complementary aspects of semantic plausibility.

Methodological Contributions. Beyond architecture comparison, we introduced uncertainty-aware training strategies that incorporate human annotation variance directly into the loss function. This approach proved effective across all model families, particularly for mid-range plausibility cases where annotator disagreement is highest. Additionally, our formulation of expected value decoding for T5-based regression enables continuous

output from discrete generative models without sacrificing differentiability.

Limitations. Several constraints bound the generalization of our findings. The **dataset size** (2,868 samples) limits conclusions about scaling behavior; larger benchmarks would better distinguish architectural advantages. The **inherent subjectivity** of plausibility ratings (mean annotator $\sigma = 0.95$) imposes a performance ceiling that may mask true model differences. Our **computational constraints** prevented full exploration of larger models (e.g., Mistral at higher precision, Flan-T5 XXL) or extensive hyperparameter sweeps.

Future Directions. This work opens several avenues for extension. For example, **Calibration techniques** like temperature scaling may better align model confidence with human uncertainty. Investigating **chain-of-thought prompting** for decoder-only models could also improv performance by leveraging their reasoning capabilities more explicitly.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, volume 24.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2022. Scaling instruction-finetuned language models. In *arXiv preprint arXiv:2210.11416*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Katrin Erk. 2016. What do you know about an alligator when you know the company it keeps? *Semantics and Pragmatics*, 9:17–1.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Chaplot Devendra, Guillaume Lample, Arthur Mensch, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

David Jurgens and Ioannis Klapaftis. 2013. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 290–299.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, YongLong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*. NeurIPS.

Adam Kilgarriff and Joseph Rosenzweig. 2004. The senseval-3 english tasks: task specification and performance. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 78–83.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2025. Prompt repetition improves non-reasoning llms. *arXiv preprint arXiv:2512.14982*.

Frederick Liu, Terry Huang, Shihang Lyu, Siamak Shakeri, Hongkun Yu, and Jing Li. 2022. Enct5: A framework for fine-tuning t5 as non-autoregressive models. *arXiv preprint arXiv:2110.08426*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Ulme Wennberg Petersen and Sharon Goldwater. 2020. Simple bert models for relation extraction and semantic role labeling. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 54–64.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Alexandra Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language model with self generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 13484–13508.

A Code Repository

We published a meta repository that contains all the relevant code to reproduce our experiments. The execution logic of each approach is decoupled in its own repository, linked as a submodule. Navigate to the target approach directory under approaches/ and consult the local README.md. Each submodule contains self-contained instructions for environment instantiation, data preprocessing, and inference specific to that architecture.

https://github.com/VitoFe/SemEval2026Task5_Submission