# DNS at SemEval 2026 Task 6: Encoder-Centric Approach for Classifying Clarity and Evasion in Political Interview Responses

**Nickolaus Jackoski** **Sungboo Park** **Dima Golubenko**
Department of Computer Science
University of Colorado Boulder
{Nick.Jackoski, Sungboo.Park, Dima.Golubenko}@colorado.edu

## Abstract

Political communication inherently exhibits lower clarity and a stronger tendency toward evasive responding compared to common discourse. In this work, we extend our focus from simple clarity judgment to the classification of fine-grained evasion types in political interview question–answer datasets. Based on insights from related studies, we explored a broad spectrum of approaches from traditional statistical machine learning methods to encoder-based architectures and current LLMs. We evaluated our system using macro-F1 score. Our ensemble method DeBERTa-large + RoBERTa-large achieved 0.650 on Subtask 1, and DeBERTa-v3-large model got 0.627 on Subtask 2.

## 1 Introduction

### 1.1 Background and Motivation

Political figures often use unclear or evasive answers during interviews. This recurrent theme leads to a disconnect between politicians and citizens, which can create confusion, misinformation, and, in critical circumstances, can even be dangerous. According to Bull (2003) meta-analysis of five studies on political interview questions and answers, politicians provide direct answers in only 39-46% of questions in televised interviews, well below the 70-89% typically observed among non-politicians. This difference in percentages between citizens and politicians suggests that evasiveness is a common behavior amongst politicians, not an isolated one. The purpose of SemEval Task 6 is to build two classification models that distinguish the various forms of ambiguity in politicians' unclear and evasive responses. As emphasized in their motivation for clarity and evasion classification, this task requires models to move beyond surface-level semantics and instead capture pragmatic cues, strategic indirectness, and the interactional dynamics between a question and its response. Based on this perspective, our work investigates how encoder-based models can be leveraged to perform fine-grained evasion classification using the QEvasion dataset introduced by Thomas et al. (2024).

### 1.2 Task Challenges

Both tasks, subtask 1 clarity level classification, and subtask 2 evasion level classification presents several challenges such as class imbalance, predicting the correct annotation, and technical modeling limitations. The task is inherently complex because it involves classifying response clarity, which requires bidirectional encoding and reasoning across a long conversational context, a task that traditional NLP models are not optimized for making the correct predictions. Furthermore, ambivalent replies are particularly susceptible to confirmation bias, as slightly indirect answers are difficult to classify as ambivalent rather than adequate responses. These subjective interpretations complicate the definition of category boundaries and weaken the clarity of label separability. Annotation itself introduces further challenges. As the number of labels increases, annotation errors naturally become more frequent, especially in distinguishing ambivalent categories from neighboring classes. Disagreements are common between General vs. Explicit replies and between Partial vs. Explicit replies.

Modern bidirectional encoders and large language models struggle to achieve adequate F1 scores when applied to response clarity and evasion classification, due to the need for a detailed understanding of context, especially in cases such as clear versus ambivalent replies. Model performance improves with larger models that include learned embeddings before fine-tuning and inference, while small models lacking a similar training background perform inadequately compared to larger models. These machine learning models continue to struggle with classifying clear and

ambivalent replies indicating sensible distinctions remain a barrier for computational systems.

### 1.3 Approach overview

Our approach begins with simple lexical baselines and incrementally progresses toward more sophisticated architectures. This development trajectory mirrors the iterative methodology commonly observed in prior studies, moving from classical feature-based classifiers toward larger transformer models capable of modeling long-range dependencies and pragmatic intent. By systematically exploring a broad spectrum of architectures from TF-IDF logistic regression (Sparck Jones, 1972; Cox, 1958) to DeBERTa (He et al., 2021) and llama3-8B model (Grattafiori et al., 2024) we attempt to establish a reliable performance landscape for evasion classification and identify model behaviors associated with particularly difficult categories.

| Label | Train | Development | Validation |
|---|---|---|---|
| Clear Reply | 947 | 105 | 86 |
| Ambivalent Reply | 1834 | 204 | 207 |
| Clear Non-Reply | 320 | 36 | 24 |

Table 1: Distribution of Instances Across Clarity Labels in Training, Development, and Validation Sets.

| Label | Train | Validation | Test |
|---|---|---|---|
| Explicit | 842 | 105 | 105 |
| Implicit | 390 | 49 | 49 |
| General | 308 | 39 | 39 |
| Partial/Half-answer | 66 | 8 | 8 |
| Dodging | 625 | 71 | 71 |
| Deflection | 332 | 38 | 38 |
| Clarification | 78 | 9 | 9 |
| Declining to answer | 126 | 15 | 15 |
| Claims ignorance | 105 | 12 | 12 |

Table 2: Distribution of Instances Across Evasion Labels in Training, Validation, and Testing Sets.

The challenge is further compounded by the dataset's long-tailed label distribution. As noted in prior work (Thomas et al., 2024), categories such as Explicit Reply appear frequently, whereas others including Partial/Half-Answer contain fewer than 80 examples. As shown in Tables 1 and 2, this imbalance contributes to substantial confusion among semantically adjacent classes such as General, Implicit, Dodging, and Deflection, reflecting the same annotation difficulties reported in earlier analyses (e.g., $\kappa$ as low as 0.58 for General vs. Explicit). Consequently, our system design places emphasis on handling skewed distributions, stabilizing training under label sparsity, and improving representational robustness across neighboring categories.

Our final system integrates targeted preprocessing, clarity-informed input formatting, weighted loss functions, and seed-averaged ensembling. Through comprehensive experimentation across multiple encoder families—including RoBERTa-large (Liu et al., 2020), XLNet-large (Yang et al., 2019), SetFit (Pannerselvam et al., 2024), and LLaMA-3-8B (Grattafiori et al., 2024) with LoRA (Hu et al., 2022)—we find that DeBERTa-v3-large (He et al., 2021) consistently provides the strongest performance. This paper details the components of our system and the insights gained from these experiments.

## 2 Related Work

### 2.1 Baseline Selection

In Yusupujiang et al. (2022)'s response space classification experiment, feature-based classical Machine Learning algorithms were employed to classify responses. Their work reported that classical machine learning methods achieved classification performance comparable to BERT-based models (Devlin et al., 2019). This observation motivated us to adopt a TF-IDF–based Logistic Regression model (Sparck Jones (1972); Cox (1958)) as our baseline.

### 2.2 Encoder-based approach

Alvarez and Morrier (2025) used a Sentence-BERT (Reimers and Gurevych, 2019) and semantic similarity to clarify evasive answers. Their work demonstrated that clarity classification using encoder embeddings and cosine similarity proved to be an effective approach. We also focused on experiments conducted on earnings conference call Q&A datasets[1] from financial-sector CEOs, where evasive responses frequently appear, like in political discourse. In particular, Nuaimi et al. (2025)'s research that utilized ALBERT (Lan et al., 2019) and cosine similarity between question and answer embeddings successfully predicted next-quarter earnings. Alongside Alvarez and Morrier (2025)'s work, this result provided additional evidence supporting the capability of encoder-based methods.

[1] https://www.kaggle.com/datasets/gautiermarti/earnings-calls-qa-evasive-answers

In addition, deep encoders such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2020), and XLM-R (Conneau et al., 2019 present strong capabilities in capturing contextual and pragmatic

aspects of meaning (Wang et al., 2023), enabling them to interpret subtle cues (BERTective; Fornaciari et al., 2021) such as indirectness, hedging, and topic shifts that simpler lexical models often fail to detect. These findings led us to assume that encoder embeddings are capable of capturing the contextual information required for this task, and we conducted our experiments primarily using encoder-based models.

Taken together, these prior approaches and our results reinforces both the usefulness and limitations of encoder-based models that have been validated in previous studies.

## 3 Task Description

Subtask 1 of CLARITY SemEval 2026 (Task 6) is a 3-label classification task that focuses on response clarity. The objective is to determine how clearly an interviewee answers a question based on three categories: clear reply, ambivalent reply, and clear non-reply. Furthermore, the taxonomy for this Subtask is designed to capture the clarity of the information provided, independent of the speaker's intent or truthfulness.

Subtask 2 of CLARITY SemEval 2026 (Task 6) is a 9-label classification that focuses on response evasion strategies. The evasion label consisted of nine evasion techniques, including explicit, implicit, dodging, deflection, partial/half-answer, general, declining to answer, claims ignorance, and clarification. These categories represent specific discourse strategies in which interviewees respond unclearly, incompletely, or evasively.

Subtask 1 and Subtask 2 are both evaluated using macro F1 scores to ensure balanced performance, as the dataset is heavily imbalanced, with several minority classes accounting for less than 5% of the dataset, leading to a long tailed distribution.

Overall, Thomas et al. (2024) introduces and defines these categories and provide annotated data created by "three human annotators alongside an expert with a background in political science and political discourse analysis who acts as a validator".

## 4 Methods

For both Subtask 1 and 2, we performed a comparison of architectures ranging from our baseline (Logistic Regression with tf-idf) model to encoder models (RoBERTa, DeBERTa, XLNet), few-shot learning (SetFit), and instruction-tuned LLM (LLaMA-8B with LoRA) using the appropriate training techniques for each model class, although evaluating the models based on Macro F1.

Furthermore, as suggested by Thomas et al. (2024) we start with fine grained Subtask 2 evasion level classification to solve the coarse grain clarity level classification Subtask 1. This relationship between the two classes is shown in Figure 1, which illustrates the taxonomy used for clarity and evasion mapping. The reason for this is that Subtask 1 classifies answers into three broad categories (Clear Reply, Ambivalent Reply, Clear Non-Reply). Instead, training these models on Subtask 2 first enables them to learn specific behaviors (e.g., Dodging, Deflection, Ignorance, Clarification), which provides a solid baseline for what models are performing best on both Subtasks. Following this logic, we first conducted extensive experimentation on Subtask 2, identifying the architectures that performed best on the nine way classification Subtask 2, and then adapted those models for Subtask 1 classification.

### 4.1 Subtask 2 System Overview

In this section, we discuss our entire pipeline, including data preprocessing, and the experiments conducted to identify the optimal fine tuning methods for the best models. Our core approach to completing both subtasks was to prioritize subtask 2 (Evasion Detection), as it is a more fine grained classification problem than subtask 1 (Clarity Classification), according to the logic outlined in Thomas et al. (2024). Providing clearer insights into which model architecture generalizes well to the questions and answers present in the dataset.

Figure 1 shows the taxonomy introduced by Thomas et al. (2024), which relates evasion categories to clarity categories as subclasses. While we do not remap evasion labels to clarity labels in our system, this taxonomy indicates the relationship between the two subtasks. It informs our strategy of evaluating models on subtask 2 first. Since Subtask 2 is a fine grained classification problem, we first identified the strongest models on that task and then adapted those architectures for Task 1.

### 4.2 Subtask 2 Preprocessing

The preprocessing framework we used follows the approach proposed in Thomas et al. (2024), which

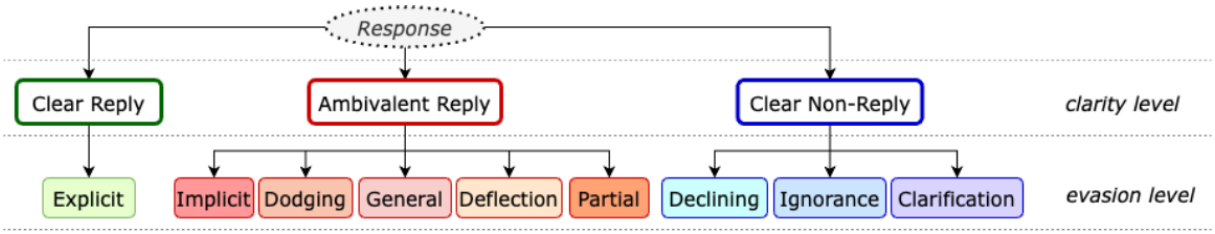


Figure 1: Illustration of the taxonomy used for clarity and evasion mapping, reproduced from Thomas et al. (2024).

includes question decomposition and contextual adjuncts (or interpretive grounding). Each instance in the data set consists of the raw text of an interview, with the question, the interviewer's answer, and an evasion label provided by an annotator. While we do not remap Subtask 2 predictions into Subtask 1 labels, this hierarchical relationship influences our training order as Subtask 2 serves as a more finegrained and diagnostic benchmark for Subtask 1. Therefore, we first run experiments on Subtask 2 to identify the best performing model, then apply those architectures without label remapping to a three way classification task. Moreover, sequence formatting, truncation, and padding are described in the tokenization section below.

#### 4.2.1 Tokenization

Most models we used had a maximum token length of 512, which fits most question-and-answer pairs while still complying with our computational and model-specific constraints. Since the default HuggingFace tokenizer is used for each model, when encountering sequences shorter than the allowed token length, the tokens are padded to this length using the current model's padding technique. Upon encountering sequences exceeding the allowed token length, the sequence is truncated from the end. This method preserves the beginning of the sequence, which contains the question and at least the start of the answer, both of which are most important for classification, and prevents distortion in classifier behavior. In typical bidirectional encoder models, which limit input length to 512 tokens, around 29% of inputs exceed that limit and are truncated. The tokenizers used were the default HuggingFace tokenizers for each model; all tokenizers follow the same methodology of truncation and padding.

#### 4.2.2 Data Splitting Subtask 2

In Subtask 2, since the test set is held out for the competition, we split the training set into 80% training, 10% development, and 10% testing selection and fine tuning for the models ran on Subtask 2. This technique was used only for fine tuning to identify which model and fine tuning method yielded the best Macro F1 score. In the final submission, we took the model with the best Macro F1 score, which was DeBERTa-v3-large, and ran it on all training data instead of doing a traditional split to produce the submission file for the best possible Macro F1 score.

### 4.3 Model Architectures

For Subtask 1, the same models and configurations as Subtask 2 unless otherwise noted, except that we replace the 9-class evasion head with a 3-class clarity classification head. Therefore, model architectures and configurations for Subtask 1 and 2 are described below.

#### 4.3.1 Logistic Regression With TF-IDF

Logistic regression is our baseline model to provide a lower bound on performance and to assess whether our larger models are necessary for Subtask 1 and 2 and perform as expected. We use TF-IDF vectorization with a maximum of 40,000 features and an n-gram range of (1,2), capturing both unigrams and bigrams. Additionally, the model removes English stop words during vectorization. The vectors are then fed into a logistic regression classifier with balanced class weights to address the imbalanced dataset.

#### 4.3.2 LLaMA-3-8B with LoRA

Motivated by the previous work in the Thomas et al. (2024), which explored LLaMA variations and reported the highest Macro F1 score for Subtask 2 with the 70-billion-parameter LLaMA model, we decided to investigate the smaller LLaMA-3-8B model for both Subtasks. Due to computational constraints, we were unable to run the LLaMA-3-70B model. For our run, we used LLaMA-3-8B with Low-rank adaptation (LoRA) and instruction tuning. For LoRA, we experimented with configurations ranging from conservative to more aggres-

sive and trained with learning rates from 1e-4 to 2e-4, using an effective batch size of 4. Due to GPU memory constraints, we were limited to running 3-4 epochs.

›

#### 4.3.3 SetFit (Paraphrase-Mpnet-Base-V2)

We also evaluated the SetFit model to test whether a sample efficient method could achieve better generalization on the imbalanced, small dataset, especially given the imbalance seen in the Subtask 2 clarity labels. Our implementation used the paraphrase-mpnet-base-v2 sentence transformer as the base encoder, incorporating clarity labels as contextual information into the input text format. To address class imbalance, we equalized the sample counts across all classes to 64 examples per label, balancing the training data and ensuring each evasion category received equal representation during training.

**Transformer-Based Models**

#### 4.3.4 XLNet-large

Based on Thomas et al. (2024), which implemented the XLNet model due to the model's ability to handle long context dependencies. We ran the model for both Subtasks, utilizing the same preprocessing pipeline and a similar fine tuning strategy as RoBERTa and DeBERTa, allowing us to evaluate if permutation based pretraining was well suited for Subtask 2 under our methodology.

#### 4.3.5 RoBERTa-large

For Subtask 1 and 2 we directly compared the RoBERTa-large (355M parameters) model with the DeBERTa-v3-large (435M parameters) model. During this comparison, we used a learning rate of 1e-5 and a batch size of 4 per device, with 4 gradient accumulation steps, resulting in an effective batch size of 16. In addition, the models were trained for 15 epochs with a weight decay of 0.05, a warmup ratio of 0.1, and a cosine learning rate scheduler. We also tried various hyperparameter fine-tuning runs, including an effective batch size of 32 and a weight decay of 0.01. We kept the random seed fixed to 42 for all runs.

#### 4.3.6 DeBERTa-v3-large

For Subtask 1 and 2 we implemented the DeBERTa-v3-large (435M parameters) model, which utilizes the disentangled attention mechanism and an enhanced masked decoder. The specifics of what we did to fine tune the model for Subtask 1 and 2 are as follows: To ensure the model's pretrained representations adapt to the 9-label classification Subtask 2, we randomly initialize a 9-label classification head on the final [CLS] representation. Additionally, for Subtask 1 we take the same approach only adjusting the [CLS] token for a 3-label classification head, as this is the default behavior of the model. All encoder weights are jointly fine tuned with the respective Subtask head. During training, a learning rate of 8e-6, 0.05 weight decay, 10% warmup, cosine decay, and a batch size of 4 with gradient accumulation of 4 (effective batch size of 16) were used. Mixed precision (FP16) is enabled, and to address class imbalance, we used a class-weighted cross-entropy loss function with label smoothing of 0.1. Furthermore, frequency-based class weighting was used to increase the impact of rare labels. Finally, we explored the effects of learning rate and batch size on performance and optimization by conducting hyperparameter fine-tuning, as discussed further in the results section. For Subtask 1, we additionally evaluated the performance of DeBERTa-v3-large compared to DeBERTa-v3-base, which is a smaller model from the same BERT family. Our goal was to evaluate the effect model capacity and performance have on on a simpler 3-label task.

### 4.4 Subtask 1 System Overview

For Subtask 1 (clarity level classification), we treat the problem as a supervised-learning 3-label natural language classification problem, where the classification labels include Clear Reply, Clear Non-Reply, Ambivalent. According to Thomas et al. (2024), Subtask 1 should be viewed as collapsing the 9 categories from the Subtask 2 into 3. This effectively means that we should treat Subtask 1 as an extension of Subtask 2. So, following the setup for Subtask 2, we used the same tokenization method for Subtask 1 as well as the same set of models that we train and run inference on. However, we trained those models directly on the 3 clarity labels instead and evaluated them using macro F1 scores over the 3 classes.

### 4.5 Subtask 1 Preprocessing

For Subtask 1, we use the same QEvasion dataset that is used for Subtask 2. Each row of this dataset contains raw interview text, an interviewer's question, a response from an interviewee, and a clarity label. For the purpose of clarity classification, we keep only the question, answer,and clarity label.

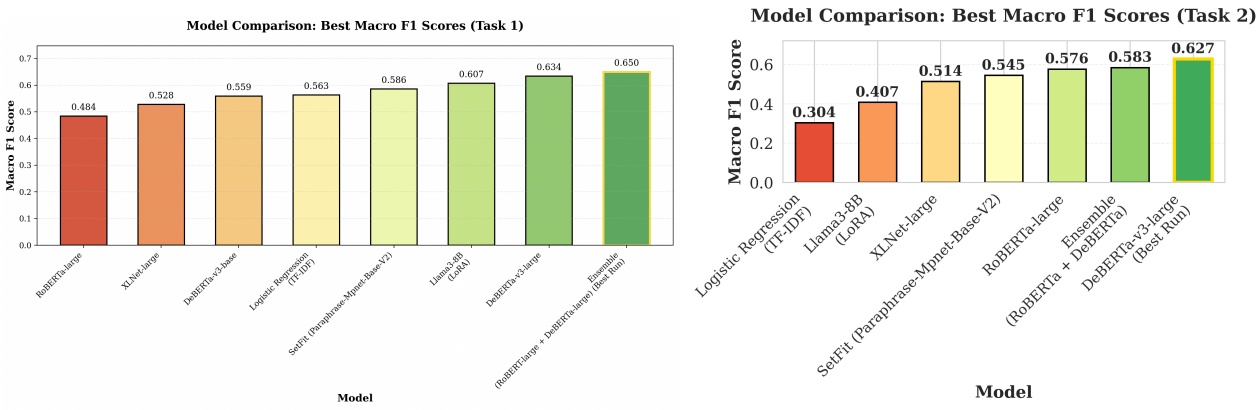


Figure 2: The left side is for the Macro F1 scores from Subtask 1 and the Right side of Figure 2 summarizes the Macro F1 scores for Subtask 2.

We concatenate the question and the response to one string, separating them with a space, and use that as the input to all models. This means making sure that both the question and the response necessary for training are provided, while ensuring that the clarity label is excluded. The tokenization method, sequence length, truncation, and padding all use the same techniques discussed in the Subtask 2 sections.

#### 4.5.1 Data Splitting Subtask 1

In Subtask 1, we do not need a separate test split to compare models, since we are no longer looking for the correct model architecture. Consequently, we split the training data into 90% training and 10% development, then test on the given test set. This approach allows us to allocate as much data as possible to training while still maintaining a sufficiently large development dataset for F1 scores, enabling early stopping and avoiding overfitting.

## 5 Results

### 5.1 Subtask 2

For our baseline model, we used standard logistic regression, which achieved a Macro F1-score of 0.3040 on the held-out test set. Based on these results, this simple model does not perform well because it cannot capture the contextual and semantic relationships required to discriminate among the nine evasion categories.

The SetFit with the Paraphrase-Mpnet-Base-V2 model achieved a Macro F1 score of 0.5449, which outperforms the logistic regression baseline. However, despite our fine tuning efforts SetFit with the Paraphrase-Mpnet-Base-V2 model's Macro F1 did not reach the best performing bidirectional encoder based models.

The LLaMA-3-8B LoRA model achieved a Macro F1 score of 0.4073. Even with parameter fine-tuning, the model remained unstable and consistently produced a lower Macro F1 score than transformer bidirectional encoder architectures. This result suggests that LoRA performed poorly, likely due to the scaling factor LLMs have: the more parameters a model has, the better its performance, which is likely why it does not come close to LLaMA-3-70 B's performance, as seen in Thomas et al. (2024). Additionally, the decoder-only structure of large language models, which prevents them from attending to future tokens, led to decreased performance when used with the bidirectional encoders we tested (RoBERTa, DeBERTa, SetFit with Paraphrase-Mpnet-Base-V2, and XLNet).

The first bidirectional encoder model we tried was XLNet-large, which achieved a Macro F1 of 0.5138, outperforming the logistic regression baseline and the LLaMA-3-8B LoRA model but underperforming RoBERTa and DeBERTa. This result is expected, as permutation-based retraining may be less suitable for detailed contextual understanding across question-and-answer pairs.

Since our first runs of RoBERTa-large performed similarly to DeBERTa-v3-large, we set up a comparison between the two models. During this comparison, the RoBERTa-large model achieved a Macro F1-score of 0.5698, while DeBERTa-v3-large achieved a Macro F1-score of 0.6055, with a difference of 0.0357. After seeing this output, we realized that RoBERTa-large may need additional fine tuning. Our best run used an effective batch size of 32 and a weight decay of 0.01, achiev-

ing an F1 score of 0.5757. Given the significant performance gap between the two models, we are confident that the non-deterministic variance would not alter the ranking of the models. Based on this, we conclude that DeBERTa-v3-large performs better on Subtask 2 and should be fine tuned further.

The best run of hyperparameter tuning for DeBERTa-v3-large gave a best overall macro-F1 of 0.6268. The learning rate made the most difference, as larger learning rates were causing instability in training, leading to divergence, while smaller ones converged slower without improving performance. However, the 8e-6 learning rate provided stability and adaptation without getting stuck in any local minima or failing to find a better maxima. Further, the effective batch size selected was 16 because smaller batches generated noisier gradients and larger batches got stuck in their local minima and generalized poorly to new data, hence lowering the Macro F1. Other hyperparameters used were weight decay of 0.05 for regularization, a warm-up ratio of 0.1 with cosine annealing as a schedule for the learning rate, 20 training epochs in total, label smoothing of 0.1 to reduce over-confidence, balanced class weights to handle the issue of class imbalance, mixed precision training enabled (FP16) for efficiency, and 777 as the seed.

Since RoBERTa-large and DeBERTa-v3-large achieved the highest Macro F1 scores out of all the models we attempted an ensemble of the two together which achieved a Macro F1 score of 0.5827, improving over RoBERTa-large alone but does not surpass the F1 score of the best DeBERTa-v3-large model. Additionally, this ensemble trial further proves that DeBERTa-v3-large is the better model as RoBERTa-large weighed down the Macro F1 of the ensembling with a score of 0.5698 while DeBERTa-v3-large achieved a score of 0.6055 during this ensemble run.

### 5.2 Subtask 1

For Subtask 1, we start with the same logistic regression that uses TF-IDF features. It produces the Macro F1 Score of 0.563, which we use as the baseline. Compared to what it yields for Subtask 2 (F1 score of 0.304), this can be considered a relatively strong result, which tells us that a 3-class classification problem might be simple enough for a simple linear regression to capture its main features.

Another model that we used is SetFit (Pannerselvam et al., 2024) that runs on paraphrase-mpnet-base-v2 encoder. This experiment further improved the Macro F1 Score to 0.586, slightly outperforming logistic regression. However, the improvement is insignificant compared to Subtask 2. Achieving this results with SetFit in comparison the baseline means that since the problem was almost linearly separable, sp using sentence-level semantic embeddings did not lead to a significant improvement.

The LLaMa-3-8B model fine-tuned with LoRA achieved a noticeably better result for Subtask 1 compared to Subtask 2. It yielded a Macro F1 score of 0.607, which is also significantly higher than both of the previous models. These results suggest that a decoder-only model can effectively capture the difference between 3 labels even though they do not have full bidirectional attention.

In contrast, the bidirectional encoder-only models that we used performed surprisingly poorly on Subtask 1. XLNet-large (Yang et al., 2019) obtained a Macro F1 score of 0.528, while RoBERTa-large (Liu et al., 2020) got only 0.484. Both of these results are below the baseline of 0.563. DeBERTa-v3-base (He et al., 2021) performed very similarly to the baseline with a Macro F1 Score of 0.559. This low performance can be attributed to the 3-class classification task being too simple to benefit from deep contextual modeling.

To address this issue, we fine-tuned a DeBERTa-v3-large (He et al., 2021) model for Subtask 1 using the same preprocessing pipeline, but a different training schedule - we set the number of epochs to 16 instead of like we did previously leaving early stopping. This configuration gave the Macro F1 score of 0.634, which outperformed the previous leader, which is LLaMa-3-8B. These results were consistent the findings from the Subtask 2: bidirectional encoder-only models become the best performing architecture once again.

In an attempt to improve the results even further, we constructed an ensemble of RoBERTa-large (Liu et al., 2020) and DeBERTa-v3-large (He et al., 2021) by averaging their logits at inference time. This ensemble yielded a Macro F1 Score of 0.650. Therefore, we can conclude that the ensemble of RoBERTa-large (Liu et al., 2020) and DeBERTa-v3-large (He et al., 2021) as the final system for the SemEval submission.

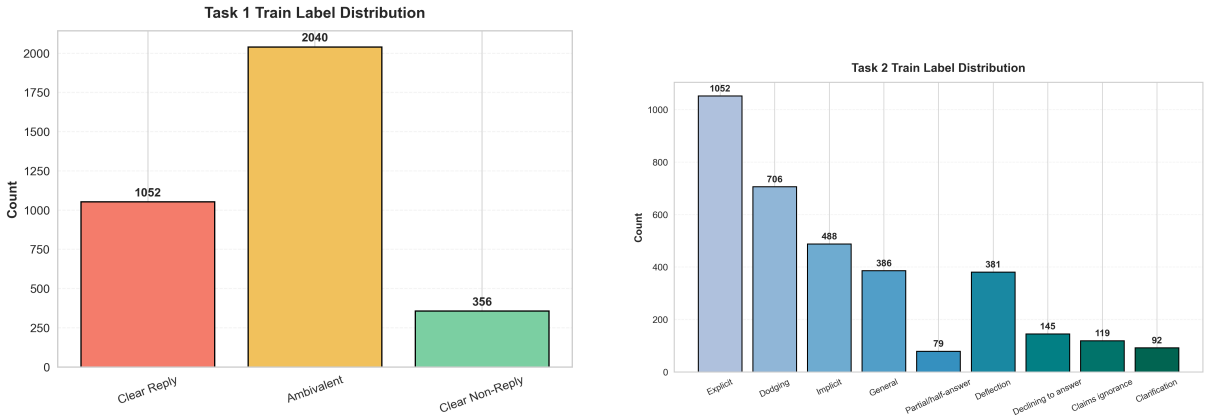


Figure 3: The left side shows the class distribution for Subtask 1 and the Right side of Figure 2 shows the class distribution for Subtask 2.

## 6 Error Analysis

This section discusses the inaccuracies observed in the system we presented due to uncontrollable factors which lead to variance in the F1 score outputs. We underline the effects of class imbalance, seed sensitivity, common misclassification patterns, and architectural limitations that shaped model performance.

### 6.1 Class Imbalance Issues

The QEvasion dataset displays severe class imbalance, specifically for Subtask 2 the "Explicit" label being around 28% of the data while the minority classes such as "Partial/Half-Answer" and "Clarification" represent only 2–3%. Additionally, for Subtask 1 the "Clarity" labels are imbalanced as well, with Ambivalent making up 59.2% of the data while clear-reply and clear Non-Reply make up 30.5% and 10.3% respectively. These imbalance distributions can be seen in Figure 3 for each subtasks labels. This imbalance introduces challenges during training, including bias toward majority classes learning to overfitting for those classes and underfitting for the minority classes. In reviewing potential errors we observed repeated confusions from the model between categories such as "Dodging" and "Deflection" or "Implicit" and "General" in subtask 2. These patterns suggest that the fine grained distinctions between certain categories are difficult for the model to learn consistently, especially when the minority classes have significantly fewer examples to exhibit their defining features.

### 6.2 Seed Sensitivity

Several extensive experiments were conducted to evaluate the sensitivity of the models performance to random seed initialization. We observed statistically significant differences in performance across random seeds with macro F1 scores fluctuating by up to 0.0380 despite having identical hyperparameters. Although this non-deterministic factor affects the reliability of the results, we concluded that the variation was not significant enough to have any model overcome the best performing model in either subtask.

## 7 Conclusion

We have conducted a comprehensive research into the effectiveness of a wide range of approaches-from traditional statistical machine learning algorithms to current LLM-based models-for distinguishing clarity and evasion in answers from political interviews. Our results show that encoder-based models, including DeBERTa variants, maintain strong performance across both Subtask 1 (clarity) and Subtask 2 (evasion). Especially, this performance gap becomes even more evident when the classification objective shifts from clarity labels in Subtask 1 to the evasion categories in Subtask 2.

These findings reproved the strengths of encoder-based models observed in prior works (Alvarez and Morrier, 2025; Nuaimi et al., 2025; Fornaciari et al., 2021), particularly their ability to capture contextual and pragmatic aspects of meaning. At the same time, the results indicate that such models can move beyond similarity-based clarity assessment (yes/no) and begin to infer the speaker's underlying communicative intentions.

## References


R. Michael Alvarez and Jacob Morrier. 2025. Measuring the quality of answers in political q&as with large language models. *Political Analysis*, page 1–18.

Peter Bull. 2003. *The microanalysis of political communication: Claptrap and ambiguity*. Routledge.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Tommaso Fornaciari, Federico Bianchi, Massimo Poesio, and Dirk Hovy. 2021. BERTective: Language models and contextual information for deception detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2699–2708, Online. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 181 others. 2024. The Llama 3 Herd of Models. *arXiv e-prints*, arXiv:2407.21783.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Khaled Al Nuaimi, Gautier Marti, Alexis Marchal, and Andreas Henschel. 2025. Detecting evasive answers in financial Q&A: A psychological discourse taxonomy and lightweight baselines. In *Proceedings of The 10th Workshop on Financial Technology and Natural Language Processing*, pages 191–196, Suzhou, China. Association for Computational Linguistics.

Kathiravan Pannerselvam, Saranya Rajiakodi, Sajeetha Thavareesan, Sathiyaraj Thangasamy, and Kishore Ponnusamy. 2024. SetFit: A robust approach for offensive content detection in Tamil-English code-mixed conversations using sentence transfer fine-tuning. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 35–42, St. Julian's, Malta. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Konstantinos Thomas, Giorgos Filandrianos, Maria Lymperaiou, Chrysoula Zerva, and Giorgos Stamou. 2024. "i never said that": A dataset, taxonomy and baselines on response clarity classification. *Preprint*, arXiv:2409.13879.

Zijie Wang, Md Mosharaf Hossain, Shivam Mathur, Terry Cruz Melo, Kadir Bulut Ozler, Keun Hee Park, Jacob Quintero, MohammadHossein Rezaei, Shreya Nupur Shakya, Md Nayem Uddin, and Eduardo Blanco. 2023. Interpreting indirect answers to yes-no questions in multiple languages. *Preprint*, arXiv:2310.13290.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zulipiye Yusupujiang, Alafate Abulimiti, and Jonathan Ginzburg. 2022. Classifying the response space of questions: A machine learning approach. In *Proceedings of SemDial 2022*, pages 59–69, Dublin, Ireland.