CLARITY - Unmasking Political Question Evasions

1. Problem Description

1.1 Background

In the era of **mass information dissemination**, **question evasion** and **response ambiguity** are widespread phenomena in political interviews and debates. Detecting these phenomena is an important aspect of political discourse studies.

Research by Bull (2003) presents a meta-analysis of five studies on political interview Q&As, concluding that:

- Politicians gave clear responses to only **39–46%** of questions during televised interviews
- Non-politicians had a significantly higher **70–89%** reply rate

1.2 Task Definition

Response Clarity Classification is an NLP task that evaluates how clearly a given response addresses its corresponding question.

Formal definition:

- **Input:** A question-answer pair (Q, A) from a political interview
- **Output:** A clarity label from a predefined taxonomy indicating how unambiguously the response addresses the question

Task objective: Given a question Q and its corresponding answer A, classify the response into one of three clarity categories (Clear Reply, Ambivalent Reply, Clear Non-Reply) and optionally into one of nine fine-grained evasion sub-categories.

1.3 Proposed Taxonomy

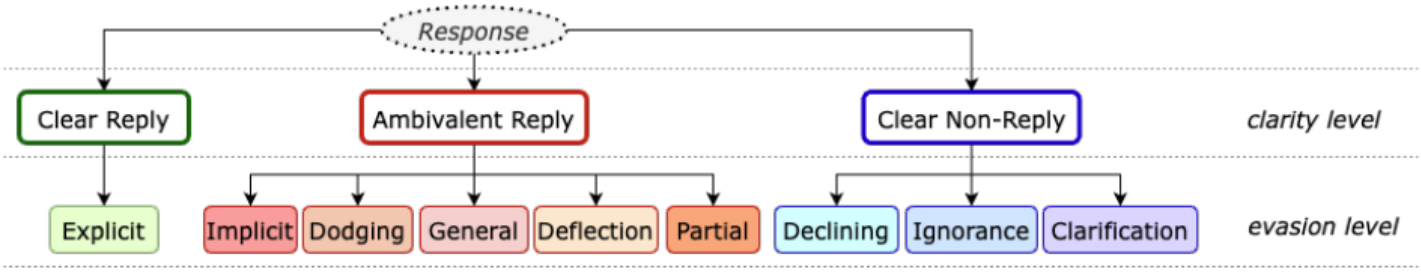
For this problem it is proposed a **two-level hierarchical taxonomy** for response clarity classification:

High-level (3 categories):

1. **Clear Reply** - replies that admit only one interpretation
2. **Clear Non-Reply** - responses where the answerer openly refuses to share information
3. **Ambivalent Reply** - a response is given in the form of a valid answer but allows for multiple interpretations

Low-level (9 evasion sub-categories) - explains in more detail the categorisation of responses:

- Explicit (under Clear Reply)
- Implicit, Dodging, General, Deflection, Partial/half-answer (under Ambivalent Reply)
- Declining to answer, Claims ignorance, Clarification (under Clear Non-Reply)



2. Dataset Details

2.1 Data Source

- **Source**: Official White House website presidential interviews
- **Time span**: 2006–2023
- **Total interviews**: 287 unique interviews
- **Total QA pairs**: 3445 question-answer pairs

2.2 Presidents Covered

In this dataset, are present interviews from the four most recent U.S. Each president exhibits distinct patterns in response clarity, with notable differences in their use of evasion techniques.

President	Service Period
George W. Bush	2001–2009
Barack Obama	2009–2017
Donald J. Trump	2017–2021
Joseph R. Biden	2021–2023

2.3 Label Distribution

Clear Reply

- Explicit: 1051
- **Total: 1051**

Ambivalent Reply

- Dodging: 704
- Implicit: 488
- General: 386
- Deflection: 381
- Partial/half-answer: 79
- **Total: 2038**

Clear Non-Reply

- Declining to answer: 145
- Claims ignorance: 119
- Clarification: 92
- **Total: 356**

Distribution by clarity level:

- **Clear Reply**: 1051 responses (30.5%) - politicians provide direct, unambiguous answers
- **Ambivalent Reply**: 2038 responses (59.2%) - the majority of responses allow for multiple interpretations
- **Clear Non-Reply**: 356 responses (10.3%) - explicit refusals to answer or requests for clarification

2.4 Response Clarity Statistics per President

Response Type	G.W. Bush	B. Obama	D.J. Trump	J.R. Biden
Clear Reply	34.31%	22.38%	32.60%	37.34%
Clear Non-Reply	8.68%	9.50%	11.77%	10.53%
Ambivalent	57.00%	68.12%	55.62%	52.13%

Key observations:

- Barack Obama has the highest rate of Ambivalent (evasive) replies at 68.12%
- Joseph Biden provides the most Clear Replies at 37.34%
- All presidents use Ambivalent replies more than half the time

2.5 Annotation Process

The annotation pipeline combines LLM assistance with human expertise:

1. **Question decomposition**: ChatGPT breaks multi-barrelled questions into singular QA pairs (sQAs)
2. **Human annotation**: 3 non-expert annotators label each sQA using the taxonomy
3. **Expert validation**: A political science expert validates annotations and resolves conflicts
4. **Quality control**: Weekly checks and counterfactual sQAs ensure annotator attentiveness

2.6 Dataset Availability

The dataset is publicly available on Hugging Face as **QEvasion**: ailsntua/QEvasion

Dataset splits:

- Train: 3448 rows
- Test: 308 rows

Key features in the dataset:

- `interview_question` - the full original question posed by the interviewer
- `interview_answer` - the full text of the interviewee's response
- `question` - extracted sub-question (for multi-part questions)
- `clarity_label` - high-level label: "Clear Reply", "Clear Non-Reply", or "Ambivalent Reply"
- `evasion_label` - fine-grained sub-category (Explicit, Implicit, Dodging, General, Deflection, Partial/half-answer, Declining to answer, Claims ignorance, Clarification)
- `president` - name of the president involved
- `date` - date of the interview
- `multiple_questions` - boolean flag for multi-part questions
- `gpt3.5_summary` - optional GPT-3.5 generated summary
- `gpt3.5_prediction` - optional GPT-3.5 predicted label

3. Approaches used and State of the Art

3.1. Existing Literature

While “politicians dodging questions” is an old problem, framing it as a hierarchical classification task (based on evasion and clarity) for language models is quite novel. As such, the authors of the CLARITY dataset (that also proposed the problem) have tested multiple models and approaches that could represent a good baseline for the underlying problem. Out of these, the best results will be considered as state of the art for now, as the contest is still ongoing. These approaches are presented in the next section.

Other than that, *“Did They Answer?" (Ferracane et al., 2021)* addresses the mirroring problem of **subjectivity in discourse analysis**, challenging the standard assumption that conversation acts have a single "ground truth" interpretation (just like in our problem). Focusing on adversarial Question - Answer pairs from U.S. congressional hearings, the authors argue that the interpretation of a response - specifically whether a witness is answering or evading - depends heavily on the observer's own biases. While this work may focus more on the subjective intent of a question rather than informational completeness of clarity, this can still be relevant because the exact same linguistic phenomenon is studied: **political evasion and non-cooperative answering.** To explore this, they curated a dataset of over 1,000 QA pairs where multiple annotators labeled both the conversation act (what the witness did, e.g., "answer" or "shift") and the communicative intent (why they did it, e.g., "honest" or "lying"), while also recording the annotator's sentiment toward the speakers. The results showed that genuine disagreement occurred in 53.5% of cases, confirming that perception of evasion is widely variable. In their experiments, a fine-tuned RoBERTa model outperformed traditional baselines in predicting these labels. In our problem, the interviewer and, most importantly, the labeling human are considered to be neutral, but this is not easy to achieve in practice, so this can lead to more ambiguous responses and more “confusion” for our model as it will struggle where the human struggles.

Our problem’s taxonomy (Explicit, Dodging, Clear, etc) is obviously built on discourse analysis research from social science. While these aren’t NLP papers, they define the rules that a model is trying to learn.

For example, *Bull & Mayer (1993): How Not to Answer Questions in Political Interviews* established the first comprehensive typology of political equivocation, identifying 30 distinct strategies politicians use to avoid answering questions, such as "attacking the question" or "making political points." This effectively provided the specific label set for the "Evasion" layer of the modern Clarity taxonomy. As such, our task is essentially a computational adaptation of Bull & Mayer’s descriptive work, converting their manual sociological categories into the fine-grained target labels that modern Large Language Models are trained to detect in the hierarchical classification strategy.

Rasiah (2010): A Framework for the Systematic Analysis of Evasion refined the study of evasion by introducing an "Intermediate" category for responses that are neither fully direct nor fully evasive, and distinguishing between "covert" (subtle) and "overt" (explicit) evasion. This framework provides the theoretical basis for the "Ambivalent" class in the Clarity task, which represents the grey area between clear answers and clear refusals. Her distinction between covert and overt evasion directly explains the primary computational

challenge of the Clarity task: while "overt" evasion corresponds to the easily detectable Clear Non-Reply, "covert" evasion corresponds to Ambivalent strategies like deflection, which require the deep semantic reasoning that current models struggle to perform accurately.

3.2 Practical Approaches

As mentioned, the dataset creators have provided a set of approaches using various models, which we will consider as SOTA for the moment as development and research is still ongoing.

The authors effectively implemented encoder architectures and modern generative Large Language Models (LLMs). The experimental framework was designed not just to test model architectures, but to also evaluate two distinct classification strategies: the Direct Clarity approach, where models predict one of the three high-level labels (Clear Reply, Ambivalent, Clear Non-Reply) immediately, and a hierarchical Evasion-Based approach, where models first identify specific evasion techniques (like "Dodging" or "Deflection") and then map them to the higher-level clarity classes.

The first category of approaches focused on Encoder Models, specifically DeBERTa, RoBERTa, and XLNet, which serve as standard baselines for text classification tasks. The authors fine-tuned these models using standard supervised learning. While these models are computationally efficient, they struggled significantly with the length of political interviews. Because models like RoBERTa and DeBERTa have a strict input limit of 512 tokens, a significant portion of the training data had to be truncated, leading to a loss of context. Consequently, while the "base" versions of these models achieved moderate success (with RoBERTa-base reaching an F1 score of roughly 0.58 on the direct task), the "large" versions failed to converge properly, often collapsing to predict a single label for every input.

Moving to Generative LLMs, the authors explored prompting strategies using models like ChatGPT (GPT-3.5), Llama-2, and Falcon. They tested Zero-Shot (ZS), Few-Shot (FS), and Chain-of-Thought (CoT) , prompting them to see if models could perform the task without weight updates. ChatGPT proved to be a great baseline in the zero-shot setting, significantly outperforming open-source models like Llama-7b and Falcon-7b, which frequently hallucinated labels outside the taxonomy. However, while prompting demonstrated that LLMs possess inherent world knowledge useful for the task, the lack of task-specific adaptation limited their ability to discern fine-grained evasion categories.

The most advanced approach implemented was Instruction-Tuning using LoRA (Low-Rank Adaptation). The authors fine-tuned Llama-2 (7b, 13b, 70b) and Falcon (7b, 40b) specifically on their dataset, teaching the models to follow the instruction "classify the type of answer." This method bridged the gap between raw capability and task knowledge. The results showed that fine-tuning significantly reduced

hallucinations and allowed even smaller models (like Llama-13b) to outperform larger, non-tuned models like Falcon-40b.

The study conclusively identifies the **Instruction-Tuned Llama-2-70b** using the **Evasion-Based Classification strategy** as the State of the Art. This specific configuration achieved a Macro-F1 score of **0.713**, surpassing all other baselines. The massive parameter count of the 70B model provided the necessary "prior knowledge" to recognize political entities and context, but, more interestingly, forcing the model to predict the fine-grained evasion type before inferring clarity-consistently yielded better results than asking for the clarity label directly.

4. Our Experiments

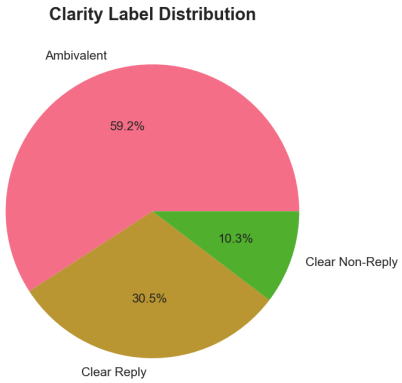
This section discusses our findings and approaches in tackling this problem, going from the initial data exploration and processing to the actual model training approaches and results.

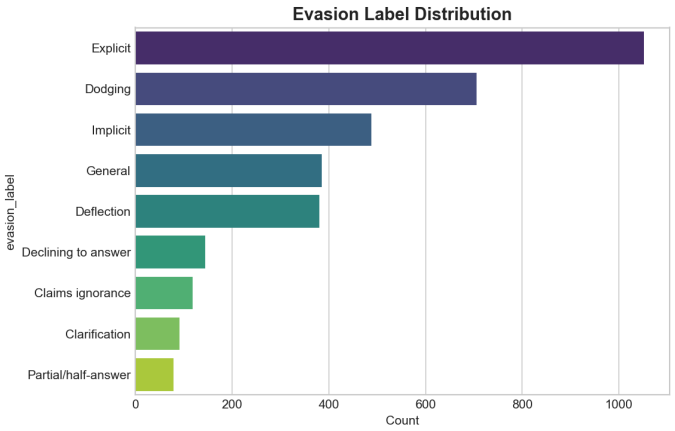
4.1 Exploratory Data Analysis (colab link)

Before diving into model selection, we performed an analysis of the dataset to better understand its structure and the linguistic patterns of political evasion.

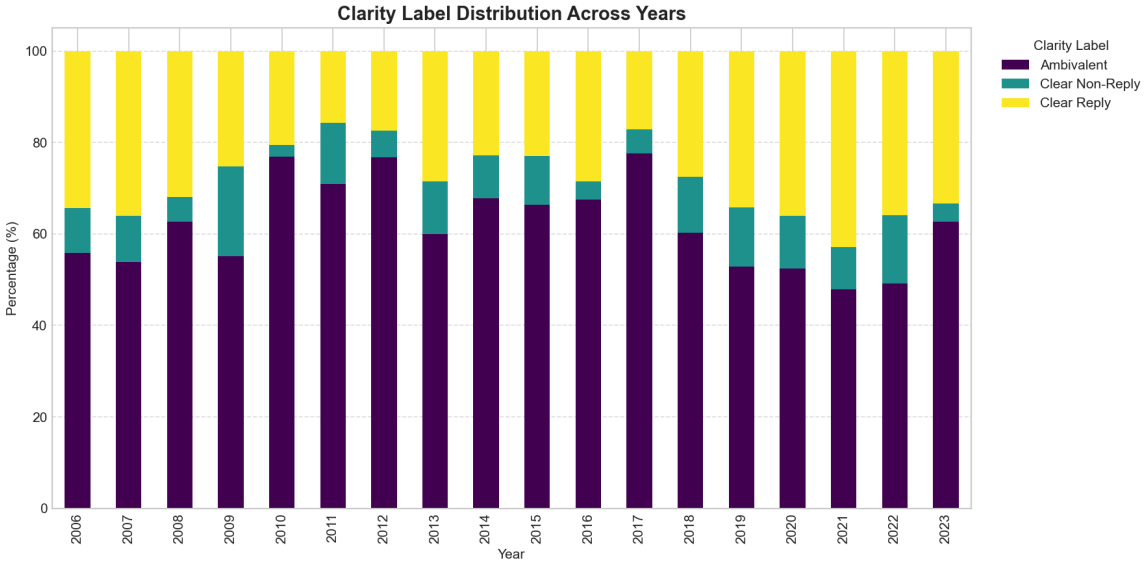
The most important features of the dataset are *interview_question*, *interview_answer*, *question*, *clarity_label*, *evasion_label* and *annotator1*, *annotator2*, *annotator3*. The *interview_question* column contains the whole set of questions asked in the interview in a singular piece of text and the *interview_answer* contains the whole response to all of these questions. Using another LLM, the authors of the dataset break the interviewer's speech into multiple atomic sub-questions, which are added in the *question* column (so multiple dataset entries may have the same *interview_question* and *interview_answer* but different questions). But, the clarity and evasion labels are referred only to the sub-responses of our sub-questions, which are not provided in the dataset. As such, a model will have to focus only on the important part of the whole answer in order to classify the clarity/evasion of each question.

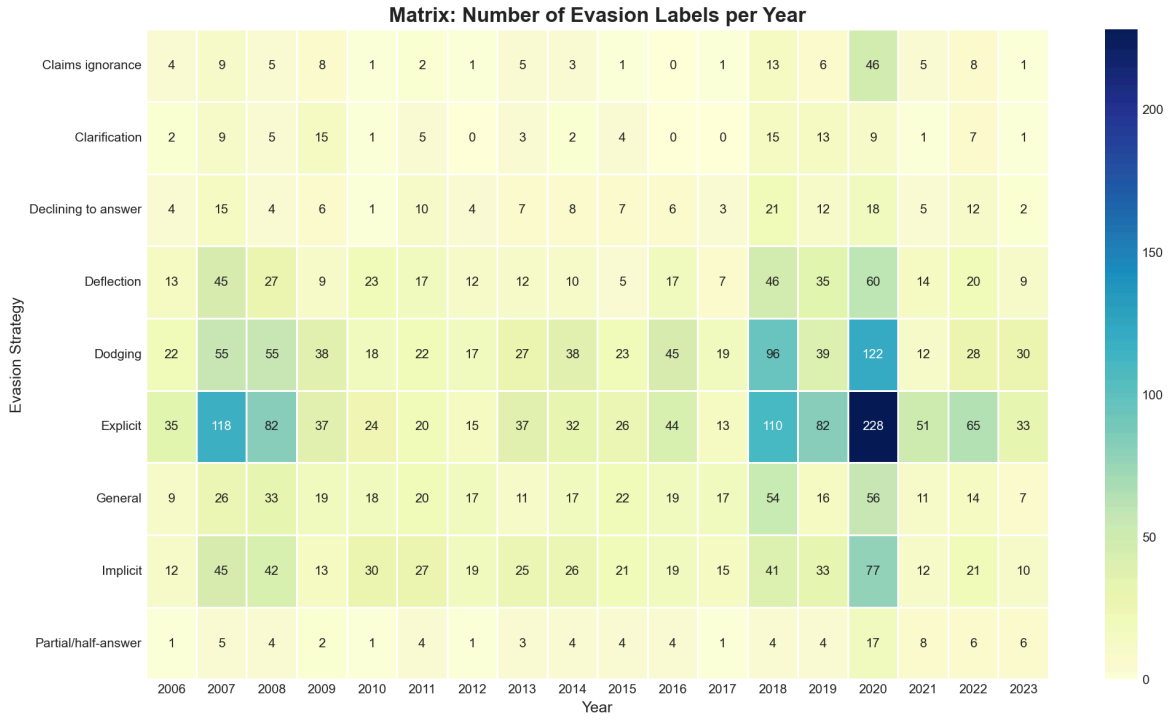
First of all, the thing that stood out in the dataset was the **class imbalance** among the clarity and evasion labels. The dataset is heavily skewed, with Ambivalent Reply accounting for roughly 59% of the training data, while Clear Non-Reply represents only about 10%. We attempted to combat this in different ways (using weight balancing, oversampling) in our approaches, which will be discussed in the next section.



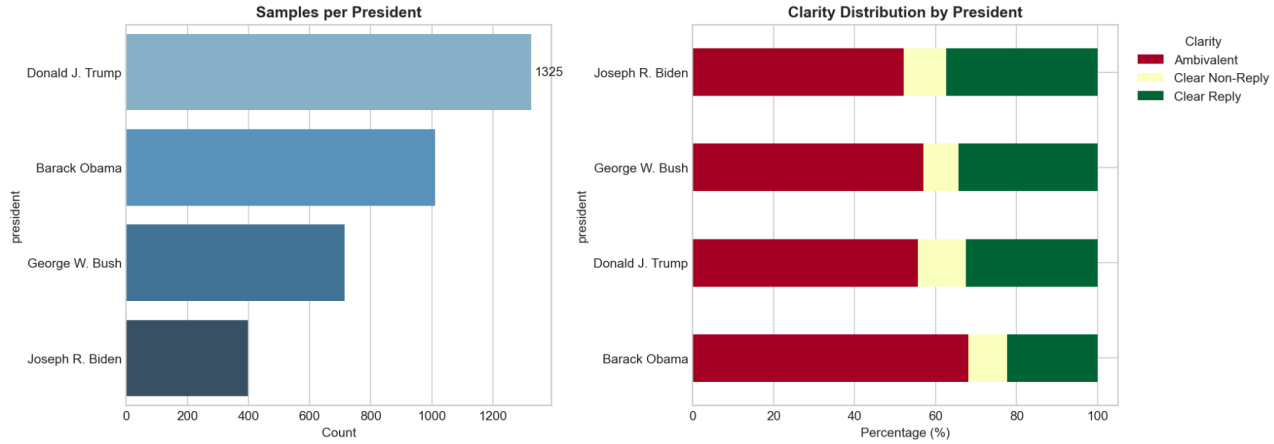


We then grouped the data by processing interview dates and extracting years to observe **temporal trends**. Interestingly, there is a spike of interviews during the 2007-2008 and 2018-2020 years and also a small percentual increase of explicit, clear-reply answers, which may be correlated to certain global events.

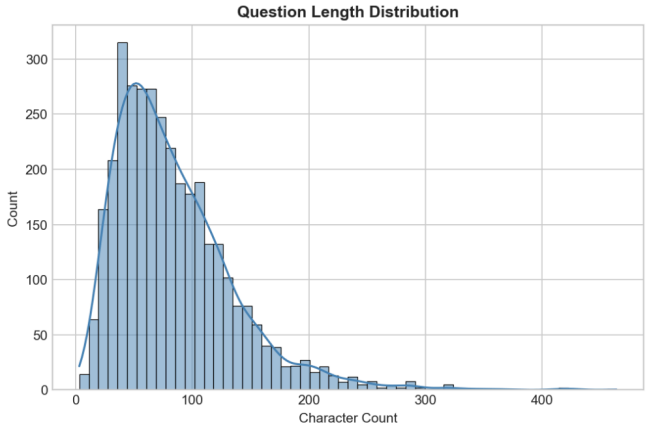


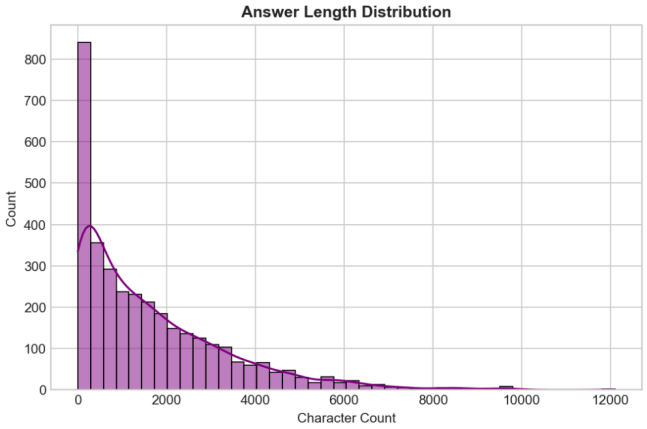


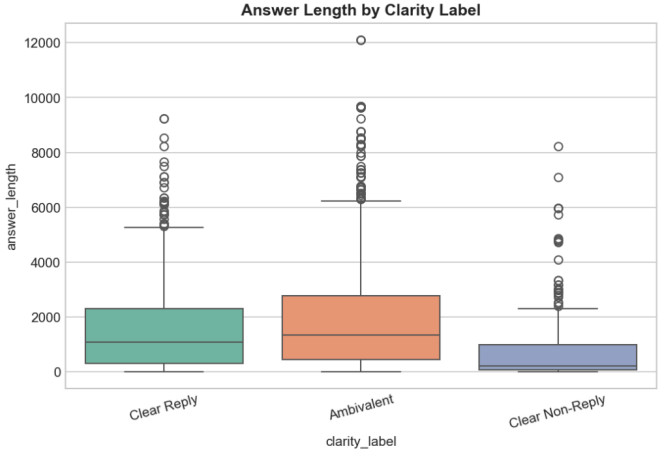
In the same fashion, we attempted to correlate the speakers (presidents) to their response types. The conclusion is the one discussed above (in the Dataset section).

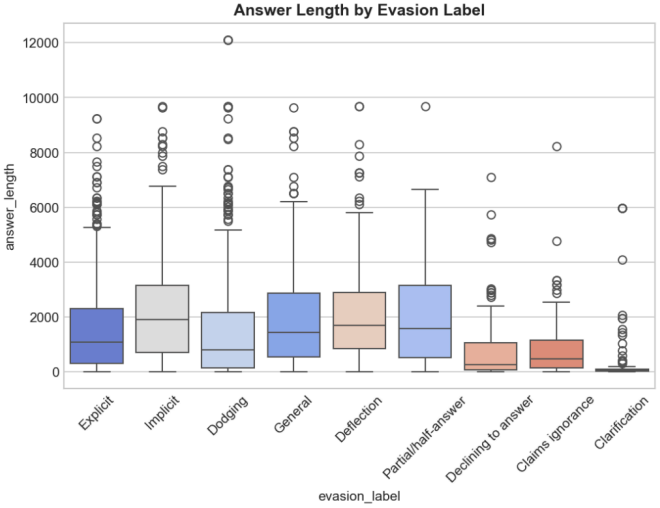


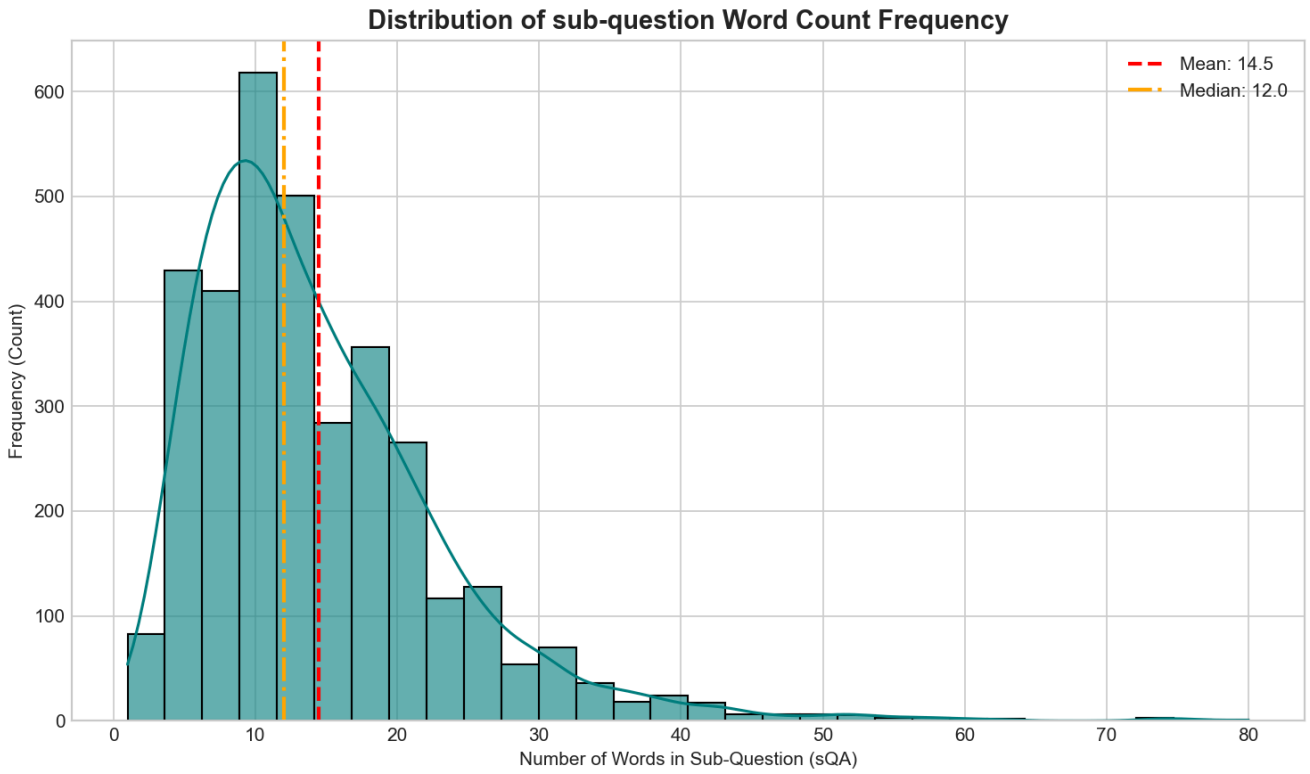
We also engineered text-based features, calculating lengths (in characters and words) of both questions and answers and correlating them to their specific label. We observed that Ambivalent replies tend to be longer than direct Clear Replies, a pattern that simpler models might use as a shortcut for classification.


Question Length Distribution
Count
Character Count
0
50
100
150
200
250
300
0
100
200
300
400


Answer Length Distribution
Count
Character Count
0
100
200
300
400
500
600
700
800
0
2000
4000
6000
8000
10000
12000


Answer Length by Clarity Label
answer_length
clarity_label
0
2000
4000
6000
8000
10000
12000
Clear Reply
Ambivalent
Clear Non-Reply


Answer Length by Evasion Label
answer_length
evasion_label
0
2000
4000
6000
8000
10000
12000
Explicit
Implicit
Dodging
General
Deflection
Partial/half-answer
Declining to answer
Claims ignorance
Clarification



Our analysis also consisted of searching for errors and inconsistencies in the dataset and filter them. As such, we found that approximately 1.3% of our data was marked as inaudible, meaning that the interview answer had potentially missing pieces (these were marked with the '[inaudible]' label inside the text), which may be misleading for a model that tries to infer the meaning of a sub-question from the whole interview answer. In conclusion, we decided to remove them.

Also, we observed that around 22% of our entries had questions (the atomic ones, not the whole interviewer's speech) that were not actually questions, but affirmative statements. This happened because each atomic question was automatically extracted from an LLM generated breakdown of the interviewer's questions, which didn't manage to keep an interrogative speech for all the examples.

This is an example of a typical "question": "ensuring Finland that the U.S. will remain a reliable NATO partner for decades to come."

As such, in order to maintain consistency in the dataset and for the models, we decided to convert these to questions. We observed that most of these affirmations are addressed in the tone of "respond to the following topic", so we found it fitting to just wrap each statement as a question in the following way: *"What about <statement>?"*. This is very quick and cheap compared to conversion using another LLM or other NLP methods.

In the end, we saved this cleaned dataset separately and used it parallel to the initial dataset in order to compare the differences in results.

4.2 Approaches

We experimented with three distinct modeling strategies, ranging from classical machine learning baselines to discriminative and generative Large Language Models. We also needed to find a resource efficient way to solve this problem as we were constrained in this aspect. For development, we used a Google Colab environment consisting of one T4 GPU and one A100 GPU.

4.2.1 Classic Machine Learning approach - XGBoost (colab link)

As a first step, we chose a classical machine learning approach to establish a solid baseline. The idea was to test if the tasks could be solved by simply transforming the input text into numerical vectors and applying a standard classification algorithm.

To implement this we used a **SentenceTransformer** (specifically the all-mpnet-base-v2 model). A SentenceTransformer is a neural network built on top of a transformer language model (like BERT or RoBERTa) that turns whole sentences into fixed-length numeric vectors. Internally, it processes the text using self-attention mechanisms to get contextual embeddings for each token, and then applies a pooling operation (e.g., mean of all vectors) to collapse the entire sequence into a single representation. The result is a high-dimensional vector (768 dimensions in our case) where semantically similar texts are located close together in the vector space.

We used this model to encode each question and answer into its own vector. We also computed the cosine similarity between them to measure how "close" the answer was to the question (e.g., similar words / topics). These features were then flattened and fed into an **XGBoost** classifier.

Following the observations from the contest's paper, we trained the model on evasion labels (9 classes) for both tasks. The final clarity predictions for Task 1 were derived by mapping the predicted evasion labels to their corresponding clarity labels.

Experimental Setup:

For feature extraction, we used the pre-trained all-mpnet-base-v2 model without further fine-tuning, generating static embeddings of size 768.

To optimize the XGBoost classifier, we performed a Grid Search over key hyperparameters to maximize the Macro-F1 score on the validation set. The search space included:

- **n_estimators**: The number of trees to build. We tested values such as **[200, 500, 1000]**. A higher number allows the model to learn more complex patterns but increases the risk of overfitting.
- **max_depth**: The maximum depth of each tree. We explored depths of **[4, 6, 8, 10]**. Deeper trees can capture more specific interactions but are also prone to overfitting.
- **learning_rate**: The scaling factor that controls how much each new tree's predictions are allowed to change the current ensemble prediction. We experimented with rates like **[0.05, 0.1, 0.2, 0.3]**.

The Grid Search identified the optimal configuration as **n_estimators=200**, **max_depth=10**, and **learning_rate=0.2**. The performance results for this baseline are presented in the Results section below.

4.2.2 Large Language Model - Discriminative Approach (colab link)

For our discriminative approach, we fine-tuned the **RoBERTa-base** model for sequence classification using input sequences formatted as '**Q: [Question] \n A: [Answer]**'.

To handle the class imbalance, we computed class weights and incorporated them into the training loss function. This ensured that the model did not simply learn to predict the majority class (**Ambivalent Reply**) but paid adequate attention to the minority **Clear Non-Reply** cases.

Similar to the XGBoost baseline, we trained on the 9 fine-grained evasion strategies (Task 2) and mapped predictions to the 3 clarity classes (Task 1), allowing the model to leverage granular supervision.

Unlike the paper's baselines which discarded sequences longer than 512 characters, we utilized truncation to retain all training samples.

Experimental Setup:

To optimize the RoBERTa-base model, we performed a Grid Search over key hyperparameters to maximize the Macro-F1 score on the validation set. The search space included:

- **Learning rate:** We tested values of **[5e-5, 1e-5, 5e-6, 1e-6]**.
- **Batch size:** We experimented with sizes of **[8, 16, 32]**.
- **Weight decay:** We evaluated regularization values of **[0.0, 0.01, 0.1]**.
- All experiments used the **Cross-Entropy** loss function, modified with computed class weights to address the imbalance.

The Grid Search identified the optimal configuration as **learning rate=1e-5**, **batch size=16**, and **weight decay=0.1**. The performance results for this baseline are presented in the Results section below.

4.2.3 Large Language Model - Generative Approach (colab link)

This method was backed by the findings from the paper *"Exploring the limits of transfer learning with a unified text-to-text transformer."* [5], which explicitly demonstrates that treating classification as a text generation problem works effectively. Instead of using a traditional classifier head like in the previous approach, this method is trained to literally generate the text string of the label, using single words or short phrases that the model already understands. The authors note that during fine-tuning, the model quickly learns to constrain its output, such that it doesn't generate text different from the target labels, which we also observed in our attempts.

As such, we deployed our most advanced approach: fine-tuning a Generative Large Language Model (LLM). The goal was to use the reasoning skills these models already have, treating the task as a text generation problem where the model acts as an expert analyst. Due to limited computing resources, we

needed an efficient way to fine-tune such a large model. We chose **Meta-Llama-3.1-8B** and fine-tuned it using the unsloth framework. Unsloth is optimized to significantly reduce memory usage and speed up training. It does so by using 4-bit quantization, which compresses the model weights to fit into memory without losing much accuracy.

Instead of updating all the model's parameters during backpropagation, we used **LoRA** (Low-Rank Adaptation). This method keeps the main model unchanged and only trains small "adapter" layers added to it. We applied this to all the main layers of the model, resulting in 41,943,040 trainable parameters out of 8.07 billion, meaning we only trained **0.52%** of the total number of parameters.

Experimental Setup:

First of all, we formatted the data using the alpaca format, which is then fed into the model for fine-tuning, using an initial instruction text that describes the clarity + evasion taxonomy and the required task. The input section represented the subquestion - full-answer pair, while the output represented the classified label.

```
INSTRUCTION = """You are an expert in political discourse analysis. Analyze the
following question-answer pair from a political interview and classify the evasion
strategy.

Context: The question is a specific sub-question, but the answer is the full
response. Focus only on the portion of the answer relevant to the sub-question.

The taxonomy of responses consists of 3 main clarity levels, which are further
divided into 9 specific evasion types:

1. Clear Reply (Unambiguous)
   - 'Explicit': The information requested is explicitly stated (in the requested
form)

2. Ambivalent Reply
   - 'Implicit': The information requested is given, but without being explicitly
stated (not in the expected form)
   - 'General': The information provided is too general/lacks the requested
specificity
   - 'Partial/half-answer': Offers only a specific component of the requested
information
   - 'Dodging': Ignoring the question altogether
```

```
   - 'Deflection': Starts on topic but shifts focus and makes a different point than
asked

3. Clear Non-Reply
   - 'Declining to answer': Acknowledge the question but directly or indirectly
refusing to answer at the moment
   - 'Claims ignorance': The answerer claims/admits not to know the answer themselves
   - 'Clarification': Does not provide the requested information and asks for
clarification

Task: Output ONLY the specific evasion type (e.g., 'Explicit', 'Deflection',
etc.)."""

alpaca_prompt = """Below is an instruction that describes a task, paired with an
input that provides further context. Write a response that appropriately completes
the request.

### Instruction:
{}

### Input:
{}

### Response:
{}"""
```

In an attempt to further improve the model's performance, we tried to apply a combination of multiple tweaks:

- **Enhancing the input with interviewer context:** In the initial prompt text we only provided the question and full answer as inputs for the model. The model then has to focus on the right part of the answer (in order to infer the sub-answer for the question) using only the question itself as context. As such, we also tried an enhanced version where the full interview question is given and the instruction text is modified to explain the structure.
- **Trained on clean data:** cleaned according to our data analysis.
- **Combatting class imbalance:** We attempted to train on an artificial dataset obtained using **oversampling**. We artificially multiplied the number of entries that contain few specific labels in order to

create a smoother label distribution in our dataset. From approximately 3400 entries in our dataset we reached approximately 8000 data entries.

- **Training on both evasion and clarity labels:** For the clarity task, we had two approaches: training directly on clarity labels, or training directly on evasion labels which are then converted to clarity.
- **Hyperparameter tweaking:** Batch size, gradient accumulation, epochs, learning rate, etc.

Based on these, here are some sampled results, where *macro_f1* is the F1 score for the test set:

```
{
    "config": {
      "model": "Llama-3.1-8B",
      "adapter": "LoRA",
      "r": 16,
      "epochs": 7,
      "learning_rate": 0.0002,
      "instruction": <clarity_only_enhanced_instruction_set>,
      "batch_size": 2,
      "grad_accum": 8
    },
    "train_data": "/content/drive/MyDrive/NLP-Clarity/data/train.csv",
    "macro_f1": 0.54 (clarity)
 },

{
    "config": {
      "model": "Llama-3.1-8B",
      "adapter": "LoRA",
      "r": 16,
      "epochs": 4,
      "learning_rate": 0.0002,
      "instruction": <evasion_enhanced_instruction_set>,
      "batch_size": 2,
      "grad_accum": 4
    },
    "train_data":
"/content/drive/MyDrive/NLP-Clarity/data/train_filtered_oversampled.csv",
    "macro_f1": 0.09372979961215254 (evasion)
  },

{
    "config": {
      "model": "Llama-3.1-8B",
      "adapter": "LoRA",
      "r": 16,
      "epochs": 5,
```

```
    "learning_rate": 0.0002,
    "instruction": <enhanced_instruction_set>,
    "batch_size": 1,
    "grad_accum": 1
  },
  "train_data": "/content/drive/MyDrive/NLP-Clarity/data/train_cleaned.csv",
  "macro_f1": 0.38048801201532756 (evasion) -> 0.56 (clarity)
},
```

Conclusions:

- Enhanced input with interviewer context slightly outperforms the non-enhanced versions
- Training on oversampled data leads to a very small loss after a few epochs, but very bad macro F1 -> overfitting
- Training on the clean dataset improves the score by a small margin
- Training on the evasion labels and then converting to clarity label provides better results than simple clarity training, as evasion labels are more expressive, providing more granular information

5. Best Results

To align with the competition standards, we used the official evaluation script provided by the organizers. The metric for model quality is **Macro-F1**, but its computation is adapted to handle the multi-annotator nature of the dataset. In this task, the "gold standard" for a sample is not always a single label but a set of valid labels provided by multiple annotators (e.g., Annotator 1 says "Dodging", Annotator 2 says "Implicit"). A prediction is considered a True Positive (TP) if it matches **any** of the valid labels in this set. The metric is computed as follows.

1. For each evasion class the following are computed: precision (proportion of times the model predicted a given class and was correct), recall and f1 score (which is the harmonic mean of precision and recall).
2. Finally, the Macro-F1 is derived by taking the unweighted average of the F1-scores across all classes.

This evaluation strategy ensures that the results realistically reflect the model's ability to handle the class imbalance.

Model	**Task 1: Clarity (Macro-F1)**	**Task 2: Evasion (Macro-F1)**
XGBoost (Baseline)	0.38	0.21

Llama 3.1 8B	0.56	0.38
RoBERTa-base	**0.63**	**0.49**

The fine-tuned RoBERTa-base model emerged as our best-performing approach, achieving the best score for both tasks. This discriminative approach outperformed our generative Llama-3.1-8B model, suggesting that efficient encoder-based architectures can still provide a strong, resource-friendly solution for this task, particularly when techniques like class weighting are used to combat the dataset's inherent class imbalance. In contrast, the Llama-3.1-8B model struggled to overcome the imbalance, as demonstrated by the experiments with oversampling, which resulted in the model overfitting the augmented data and leading to a very low Macro-F1 score on the test set. However, these results highlight the potential of generative models in classification scenarios, considering that we used a fairly low resource model with very imbalanced classes.

The best results from this table are backed up by actual submissions from the official contest page.
We submitted our results on the following pages:

1. **Task 1:** https://www.codabench.org/competitions/10879/
2. **Task 2:** https://www.codabench.org/competitions/11131/

6. Conclusions

This document detailed our experiments in tackling the novel Natural Language Processing task of **Response Clarity Classification**, aimed at automatically unmasking political question evasion using a two-level hierarchical taxonomy. The foundation of this work is the **QEvasion dataset**, comprising 3,445 question-answer pairs from U.S. presidential interviews, but posing a significant challenge of this domain: **class imbalance**, with "Ambivalent Reply" accounting for the majority of responses. Also, our pre-processing included converting ambiguous affirmative statements into interrogative sub-questions to maintain data consistency.
We evaluated three distinct modeling strategies:

1. **Classic Machine Learning (XGBoost):** Served as a strong baseline, achieving a Macro-F1 of 0.38 for the Clarity task, but was clearly weaker than a Deep Learning approach.
2. **Generative Large Language Model (Instruction-Tuned Llama-3.1-8B):** Achieved a Macro-F1 of 0.56. While showcasing the potential of text-to-text generative models for classification, it struggled with data augmentation techniques like oversampling, leading to overfitting.
3. **Discriminative Large Language Model (Fine-Tuned RoBERTa-base):** This approach emerged as our most effective and resource-efficient solution, achieving the best scores for both tasks with a **Macro-F1**

of 0.63 for Clarity (Task 1) and **0.49 for Evasion (Task 2)**. Its success is attributed to its encoder-based architecture combined with explicit class weighting to mitigate the dataset's inherent imbalance.

The results reinforce the finding from the existing State of the Art that the **hierarchical Evasion-Based Classification strategy** (training on the fine-grained evasion labels and then mapping to the high-level clarity labels) is superior to direct clarity prediction. Our best-performing RoBERTa-base model offers a robust and computationally practical solution for this analysis task.

References

1. Thomas, K., Filandrianos, G., Lymperaiou, M., Zerva, C., & Stamou, G. (2024). "I Never Said That": A dataset, taxonomy and baselines on response clarity classification. *arXiv preprint arXiv:2409.13879*. https://arxiv.org/abs/2409.13879
2. Bull, P., & Mayer, K. (1993). How not to answer questions in political interviews. *Political Psychology*, 651–666.
3. Ferracane, E., Durrett, G., Li, J. J., & Erk, K. (2021). Did they answer? Subjective acts and intents in conversational discourse. *arXiv preprint arXiv:2104.04470*. https://arxiv.org/abs/2104.04470
4. Rasiah, P. (2010). A framework for the systematic analysis of evasion in parliamentary discourse. *Journal of Pragmatics*, *42*(3), 664–680.
5. Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." Journal of machine learning research 21.140 (2020): 1-67.
6. Han, Daniel, and Michael Han. *Unsloth*. Unsloth Team, 2023, github.com/unslothai/unsloth.
7. Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." ICLR 1.2 (2022): 3.