**Algorithm 1: Generate Responses with Multi-Model Few-Shot Reasoning**

**Input :**

- Record set $\mathcal{R} = \{\rho^{(1)}, \rho^{(2)}, \ldots, \rho^{(N)}\}$, where each record $\rho^{(i)}$ has:
  - Drug name $d^{(i)}$
  - URL $u^{(i)}$
- Prompt set $\mathcal{Q} = \{q^{(1)}, q^{(2)}, \ldots, q^{(L)}\}$
- Few-Shot Examples $\mathcal{E} = \{(q^{(j)}, a^{(j)})\}$ (at least 2 examples)
- AI Model Endpoints $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$

**Output:** Structured answers $\hat{y}(j,i)_k$ and justifications $\hat{J}(j,i)_k$ for each $q^{(j)}$ by each model $M_k$, associated with $d^{(i)}$ and $u^{(i)}$.

**Initialization:**

1. Setup logging; load $\mathcal{Q}$.
2. Set $U_{processed} \leftarrow \emptyset$.

**Process Records:**

**foreach** $\rho^{(i)} \in \mathcal{R}$ **do**

Extract $\{d^{(i)}, u^{(i)}\}$.

**if** $u^{(i)} \in U_{processed}$ **then**

**continue**

Retrieve $T_{report}$ from $u^{(i)}$

Retrieve $T_{abstract}$ from $u^{(i)}$ Form context:

$$C(i) \leftarrow \mathcal{E} \parallel T_{report} \parallel T_{abstract}$$

**Query Models:**

**foreach** $q^{(j)} \in \mathcal{Q}$ **do**

**foreach** $M_k \in \mathcal{M}$ **do**

Submit $(C(i), q^{(j)})$ to $M_k$; let $\tilde{a}(j,i)_k$ be the raw response.

Parse $\tilde{a}(j,i)_k$ to extract:

$$\hat{y}(j,i)_k \quad \textit{and} \quad \hat{J}(j,i)_k$$

**Store & Update:**

Save $\hat{y}(j,i)_k$ and $\hat{J}(j,i)_k$ with $\{d^{(i)}, u^{(i)}\}$.

Update $U_{processed} \leftarrow U_{processed} \cup \{u^{(i)}\}$.

**Post-Processing:**

1. Validate consistency across:

$$\hat{y}(j,i)_1, \hat{y}(j,i)_2, \ldots, \hat{y}(j,i)_k$$

2. Flag discrepancies if any.

**Finalize:**

Finalize logs, close files, and output the dataset.

**Algorithm 2: Decision Tree Classification**

**Input :** Directory $D$ of CSV files. Each file $F \in D$ contains columns $\{\texttt{NSCLC\_Diagnosed}, \texttt{NSCLC\_Treatment}, \texttt{Drug\_Discontinued}, \texttt{Outcome\_Favourable}\}$ (optionally `Link`).
[1mm] Expected pattern: $P = (\texttt{yes}, \texttt{yes}, \texttt{no}, \texttt{yes})$.

**Output:** For each $F \in D$, a filtered CSV file $F_{rel}$ containing rows $r$ such that $\Big(\ell\big(r[\texttt{NSCLC\_Diagnosed}]\big), \ell\big(r[\texttt{NSCLC\_Treatment}]\big), \ell\big(r[\texttt{Drug\_Discontinued}]\big), \ell\big(r[\texttt{Outcome\_Favourable}]\big)\Big) = P$ with $\ell(x)$ denoting the lowercase, trimmed form of $x$.

**Algorithm:**

**1. Initialization:**

$\forall F \in D, \quad R_F \leftarrow \emptyset.$

**2. For each CSV file $F \in D$ with extension `.csv`:**

Set the file path: $path \leftarrow join(D, F)$.

Load the data: $df \leftarrow read\_csv(path, header = 0)$.

**3. For each row $r \in df$:**

Define

$$\vec{v}(r)\Big(\ell\big(r[\texttt{NSCLC\_Diagnosed}]\big), \ell\big(r[\texttt{NSCLC\_Treatment}]\big), \ell\big(r[\texttt{Drug\_Discontinued}]\big), \ell\big(r[\texttt{Outcome\_Favourable}]\big)\Big).$$

**if** $\vec{v}(r) = P$ **then**

$R_F \leftarrow R_F \cup \{r\}$.

**4. Output:**

**if** $R_F \neq \emptyset$ **then**

Write $R_F$ to a new CSV file $F_{rel}$ (e.g., $F_{relevant\_links.csv}$).

**else**

Log that no relevant rows were found for $F$.

**5. Termination:**

Process all $F \in D$ and terminate.

**Algorithm 3: Individual Model Classifier**

**Input :**

- **Link List:** $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_N\}$
- **LLM Responses:** For each $\ell \in \mathcal{L}$, preliminary response $r^{(1)}_{LLM}(\ell)$ (NSCLC mention) and additional responses $R(\ell)$.
- **LLM Endpoint Credentials:** Access credentials or API keys.
- **Prompts/Context:** $\mathcal{P}$ (original question set or context).

**Output:** Relevance Classification for each $\ell \in \mathcal{L}$ where

$$Label(\ell) \in \{Relevant, NotRelevant\}$$

along with $R(\ell)$ and final validation response $f(\ell)$.

**Initialization:**

1. Initialize data structure $\mathcal{R} : \ell \mapsto R(\ell)$; set up logging.

**Collect Responses:**
**foreach** $\ell \in \mathcal{L}$ **do**
  Retrieve $r^{(1)}_{LLM}(\ell)$ and store all responses $R(\ell)$.

**Initial Check:**
**foreach** $\ell \in \mathcal{L}$ **do**
  **if** $r^{(1)}_{LLM}(\ell)$ *does not contain "yes"* **then**

$$Label(\ell) \leftarrow NotRelevant$$

    **continue**

**Concatenate:**
**foreach** $\ell \in \mathcal{L}$ *that passed* **do**
  Form prompt:

$$T(\ell) \leftarrow \big(R(\ell) \,\|\, \mathcal{P}\big)$$

**Final Validation:**
**foreach** $\ell \in \mathcal{L}$ *with* $T(\ell)$ **do**
  Submit $T(\ell)$ to LLM; obtain $f(\ell)$.

**Classification:**
**foreach** $\ell \in \mathcal{L}$ **do**

$$Label(\ell) \leftarrow f(\ell)$$

**Output:**
Record $Label(\ell)$, $R(\ell)$, and $f(\ell)$ in structured format.

**Algorithm 4: Majority Vote Ensemble**

**Input :**

- **Base Folder** $B$ containing subfolders $\mathcal{F} = \{F_1, F_2, \ldots, F_L\}$ with CSV files (each with a **Link** column).
- **Model Prefixes** $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ for locating CSV files.
- **Minimum Occurrences** $m \in N$ (default: $m = 2$).
- Pre-processed CSVs via LLM-based Classification (only relevant links present).

**Output:** For each $p \in \mathcal{P}$, a CSV file listing links $\ell$ that satisfy

$$\mathcal{O}(\ell) \geq m,$$

with:

- $\mathcal{M}(\ell)$: set of models flagging $\ell$,
- $\mathcal{O}(\ell)$: total occurrence count.

**Initialization:**

1. Setup logging; define subfolders $\mathcal{F}$ and map each $F \in \mathcal{F}$ to its model.

**foreach** $p \in \mathcal{P}$ **do**

**Initialize:** For each link $\ell$, set $\mathcal{O}(\ell) = 0$ and $\mathcal{M}(\ell) = \emptyset$.

**foreach** $F \in \mathcal{F}$ **do**

Locate CSV files in $F$ with filename prefix $p$.

**foreach** *CSV file* **do**

Read file; for each row extract link $\ell$ from the **Link** column.

**if** $\ell \notin \mathcal{O}$ **then**

Set $\mathcal{O}(\ell) \leftarrow 1$ and $\mathcal{M}(\ell) \leftarrow \{modelcorrespondingtoF\}$.

**else**

Update $\mathcal{O}(\ell) \leftarrow \mathcal{O}(\ell) + 1$ and add the model to $\mathcal{M}(\ell)$.

**Aggregate:**

Define the set of qualifying links:

$$\mathcal{L} = \{\ell \mid \mathcal{O}(\ell) \geq m\}.$$

**Output:**

Create a CSV file with columns: **Link**, **Models** ($\mathcal{M}(\ell)$), and **Count** ($\mathcal{O}(\ell)$); log the number of aggregated links for prefix $p$.

**Finalize:**

1. Repeat for each $p \in \mathcal{P}$; finalize logs and close files.