# Hierarchical Social Support Detection with Byte-Level Transformers

Mario Mendieta[1,*,†]

Universidad Panamericana, Facultad de Ingeniería, Mexico City, Mexico

**Abstract**

Detecting expressions of social support in social media is important for understanding online communities and identifying positive interactions in digital discourse. The task is inherently hierarchical: a comment must first be identified as supportive, the support target must then be determined, and finally the specific group receiving the support may be categorized. We propose a hierarchical multi-task learning framework based on the byte-level Transformer ByT5, which processes raw UTF-8 text without subword tokenization. The model jointly optimizes the three classification tasks using conditional loss masking and task-weighted optimization. Experiments on annotated YouTube comments demonstrate stable performance despite strong class imbalance and sparse categories.

**Keywords**

Social Support Detection, Hierarchical Classification, Multi-task Learning, Byte-level Transformer, Social Media

## 1. Introduction

Online social media platforms have become an important space where individuals seek and offer emotional, informational, and community support. Messages expressing encouragement, empathy, or solidarity can help foster resilience, strengthen social ties, and improve users' well-being. Automatically identifying such expressions of social support can therefore provide valuable insights into online communities and enable the development of tools that help researchers and practitioners monitor positive interactions in digital environments.

Detecting social support in online discourse, however, presents several challenges. Expressions of support are often subtle, informal, and context-dependent, particularly in social media texts where spelling variations, emojis, and unconventional punctuation are common. In addition, the task is naturally hierarchical: a comment must first be classified as supportive or non-supportive, then the target of the support must be identified (e.g., an individual or a group), and finally the specific social group receiving the support may need to be categorized. This hierarchical structure introduces dependencies between tasks and exacerbates the effects of class imbalance and data sparsity in deeper levels of the classification pipeline.

This work is developed as part of our participation in the Social Support Detection (SSD) shared task at IberLEF 2026 [1], organized within the Iberian Languages Evaluation Forum [2]. The shared task provides a common annotation framework and evaluation setting for detecting expressions of social support in social media, encompassing the hierarchical structure of the task. Framing our experiments within this shared task allows our results to be compared against a standardized benchmark and connects our work to the broader effort of advancing social support detection in Spanish and other Iberian languages.

In this work, we address these challenges using a hierarchical multi-task learning framework based on the ByT5 architecture. ByT5 is a byte-level Transformer that processes raw UTF-8 text without relying on subword tokenization, making it well suited for noisy social media language. The proposed model jointly optimizes the three classification tasks corresponding to the levels of the annotation hierarchy using conditional loss masking and task-weighted optimization. This design allows the model

*IberLEF 2026, September 2026,* León *Spain*

0287502@up.edu.mx (M. Mendieta)

to learn progressively specialized representations while maintaining stable training behavior despite the strong class imbalance present in the dataset.

## 2. Related Works

### 2.1. Social Support Detection in Social Media

The automatic detection of social support in online discourse has recently emerged as an important research direction in Natural Language Processing. Social support expressed in social media interactions can play a crucial role in fostering community resilience, strengthening social ties, and improving users' well-being. Detecting such supportive behaviors computationally allows researchers to better understand online communities and to develop tools that promote healthier digital environments.

Ahani et al. [3] formally introduce the task of *Social Support Detection (SSD)* from social media texts. Their work frames SSD as a multi-level classification problem that identifies whether a comment expresses support and further determines the target of that support. Using a dataset of YouTube comments, the authors explore a range of traditional machine learning and neural models, incorporating linguistic, psycholinguistic, emotional, and sentiment features to capture supportive language patterns. Their results demonstrate that combining lexical and psycholinguistic features can effectively distinguish between supportive and non-supportive comments and identify whether the support is directed toward an individual or a group.

Complementing these detection efforts, Shahiki-Tash et al. [4] conduct a fine-grained linguistic analysis of different types of social support using LIWC, comparing supportive language across Spanish and English online communities. Their findings characterize the psycholinguistic markers associated with distinct forms of support and reinforce the value of psycholinguistic features for modeling supportive discourse.

Building on this task formulation, Tash et al. [5] introduce a dataset specifically designed for detecting social support in Spanish social media texts. Their work proposes one of the first annotated corpora for this task and explores both balanced and unbalanced versions of the dataset. The authors evaluate a variety of approaches, including traditional machine learning methods and transformer-based models, highlighting the challenges posed by data imbalance and the importance of high-quality annotated data for social support detection. Their study demonstrates that balanced datasets can significantly improve performance in the downstream classification tasks.

More recently, Kolesnikova et al. [6] investigate advanced machine learning approaches for social support detection. Their work evaluates a range of modern NLP techniques, including transformer-based architectures and large language models, and analyzes the hierarchical nature of the task. In particular, they consider a multi-level classification structure in which the first level identifies supportive comments, the second determines whether the support is directed toward an individual or a group, and the third categorizes the specific social group receiving the support. Their experiments demonstrate that transformer-based models provide strong performance for this hierarchical classification problem and highlight the importance of addressing class imbalance in the dataset.

These studies collectively establish the task of social support detection and provide the annotated datasets and baseline models that motivate the present work. In contrast to prior approaches that rely primarily on traditional machine learning models or general-purpose transformer architectures, our work investigates a hierarchical multi-task framework built on the ByT5 architecture to model the structured nature of the annotation scheme.

### 2.2. Advanced Machine Learning Methods for Social Support Detection

Kolesnikova et al. [6] present a comprehensive study of machine learning approaches for the automatic detection of social support in social media texts. Their work builds on the dataset introduced by earlier studies in this research line and evaluates several modeling paradigms across the three hierarchical classification tasks: (i) distinguishing supportive from non-supportive comments, (ii) identifying whether

the support targets an individual or a group, and (iii) categorizing the specific group receiving the support.

The dataset used in their experiments consists of YouTube comments annotated according to this three-level hierarchy. From a larger collection of more than 60,000 comments, the authors selected approximately 10,000 annotated samples for machine learning experiments. The annotation scheme mirrors the hierarchical structure used in subsequent studies, including the classification of group-level support into categories such as *Nation*, *LGBTQ*, *Black Community*, *Women*, *Religion*, and *Other*. [5]

To address the classification problem, Kolesnikova et al. evaluate a wide range of modeling techniques. Traditional machine learning approaches rely on feature-based representations such as TF–IDF vectors, unigram features, and psycholinguistic indicators derived from lexical resources such as LIWC. These representations are used to train classifiers including Logistic Regression, Support Vector Machines, and Random Forest models. In addition to these feature-based approaches, the authors experiment with neural architectures including CNN and BiLSTM models using pretrained word embeddings such as GloVe and FastText.

More recent transformer-based encoders are also explored. In particular, pretrained language models such as RoBERTa are fine-tuned for the classification tasks, while large language models including GPT-3, GPT-4, and GPT-4o are evaluated using zero-shot prompting strategies. Across these experiments, transformer-based encoders consistently outperform traditional machine learning methods, demonstrating the importance of contextual representations for capturing nuanced expressions of supportive discourse in social media. [6]

The best results reported in their study are obtained with the *RoBERTa-base* encoder. Using macro F1 as the evaluation metric, their model achieves scores of approximately **0.78 for Task 1**, **0.84 for Task 2**, and **0.80 for Task 3**. These results represent improvements over earlier approaches based on CNN models with GloVe embeddings and classical feature-based classifiers. [6]

Despite these strong results, the experiments primarily treat the classification tasks as separate problems or apply them sequentially in a pipeline. As a consequence, the hierarchical dependencies between the tasks are not explicitly modeled during training. Errors produced at earlier stages may therefore propagate to deeper levels of the classification hierarchy, particularly in the more fine-grained group categorization task.

In contrast, the approach proposed in this work explicitly incorporates the hierarchical structure of the annotation scheme into the learning objective. Rather than training independent classifiers, we adopt a hierarchical multi-task learning framework in which all tasks share a common encoder and are optimized jointly. Task-specific losses are conditionally activated according to the hierarchical labels, and task weights are used to balance the influence of deeper levels of the hierarchy during training. Additionally, while prior work primarily relies on token-level transformer encoders such as RoBERTa, our model explores the use of the byte-level *ByT5* architecture, which processes raw UTF-8 text directly and is particularly robust to the spelling variation and informal language commonly found in social media comments.

## 3. Data set

### 3.1. Dataset Description

The experiments in this work are conducted using a dataset of YouTube comments annotated for expressions of social support in online discussions [5, 3, 6]. The corpus was developed to facilitate research on computational detection of supportive behavior in social media conversations and consists of comments annotated according to a hierarchical labeling scheme.

The annotations were created following guidelines designed to capture different levels of supportive discourse, ranging from the presence of support to the identification of the social group receiving that support.

The full corpus contains both English and Spanish comments. In this work, however, we focus exclusively on the **English subset**. Restricting the experiments to a single language avoids cross-lingual

variability and allows the model to focus on learning linguistic patterns associated with supportive discourse in English comments.

The English subset contains **7,998 comments**. Each comment is annotated using a three-level hierarchical structure capturing progressively more specific aspects of supportive discourse:

- **Task 1: Support detection.** Each comment is labeled as either *Supportive* or *Non-Supportive*.
- **Task 2: Support target identification.** For comments labeled as supportive, the target of support is identified as *Individual*, *Group*, or *No* target.
- **Task 3: Group category classification.** When the support target corresponds to a social group, the group type is categorized into classes such as *Nation*, *LGBTQ*, *Black Community*, *Women*, *Religion*, and *Other*.

This hierarchical annotation scheme reflects the semantic progression from identifying supportive discourse to determining the specific target and community associated with that support. The hierarchical nature of the labels also motivates the modeling approach adopted in this work, where predictions are performed sequentially across the three tasks.

## 3.2. Label Distribution

The dataset exhibits substantial class imbalance across the hierarchical tasks.

For Task 1, the majority of comments are labeled as *Non-Supportive* (6,203 instances), while 1,795 comments are labeled as *Supportive*. The distribution of Task 1 labels is shown in Figure 1.

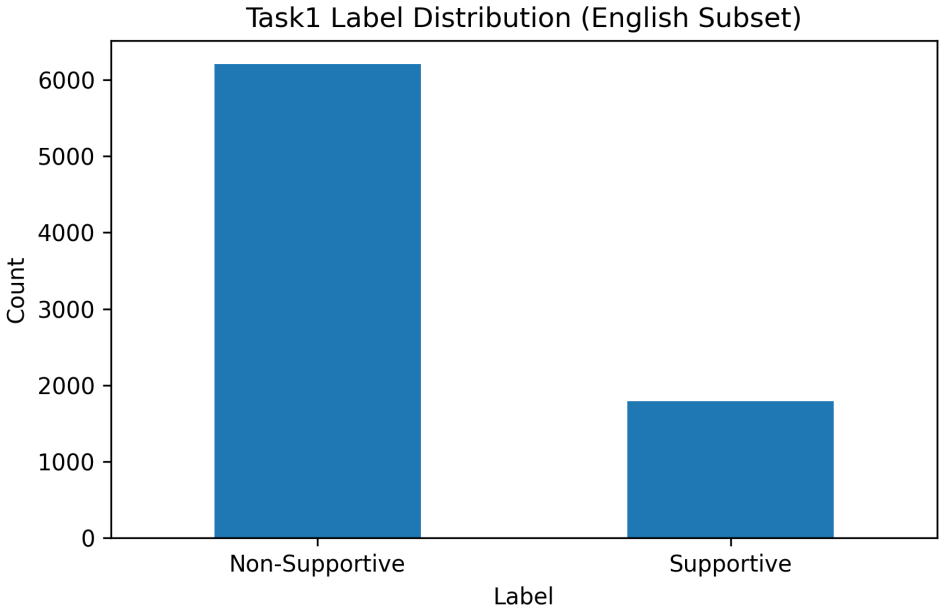


**Figure 1:** Distribution of Task1 labels in the English subset.

For Task 2, most comments contain no identifiable support target (6,203 instances). Among supportive comments, 1,450 target a group and 345 target an individual. The label distribution is shown in Figure 2.

Task 3 exhibits stronger imbalance because it is defined only for comments supporting groups. The most frequent category is *Nation* (786 instances), followed by *Other* (416), *LGBTQ* (123), and *Black Community* (91). Some classes are relatively rare, including *Women* (19) and *Religion* (15). The distribution of Task 3 categories is shown in Figure 3.

The presence of rare classes motivates the use of class-weighted loss functions and weighted sampling during model training.

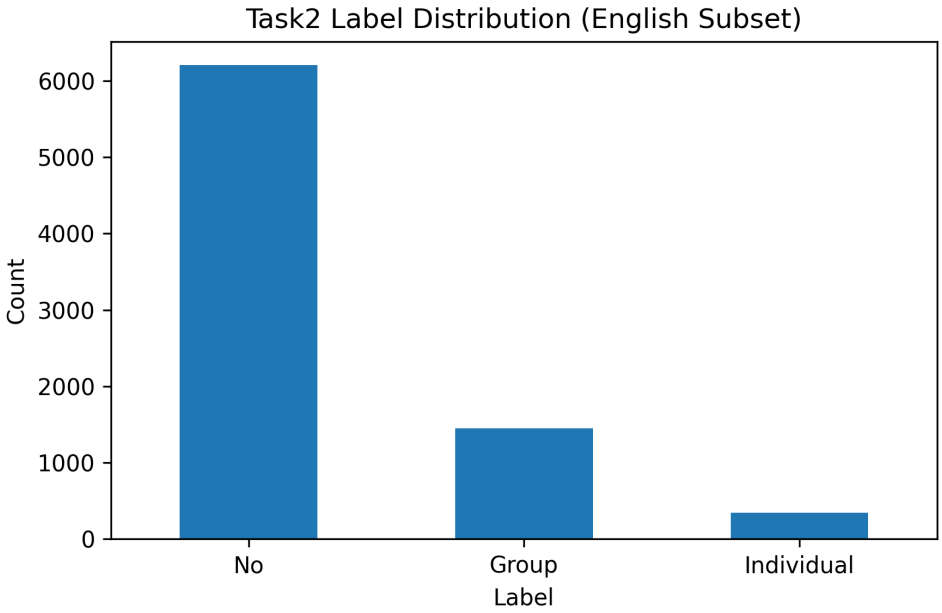


**Figure 2:** Distribution of Task2 labels in the English subset.

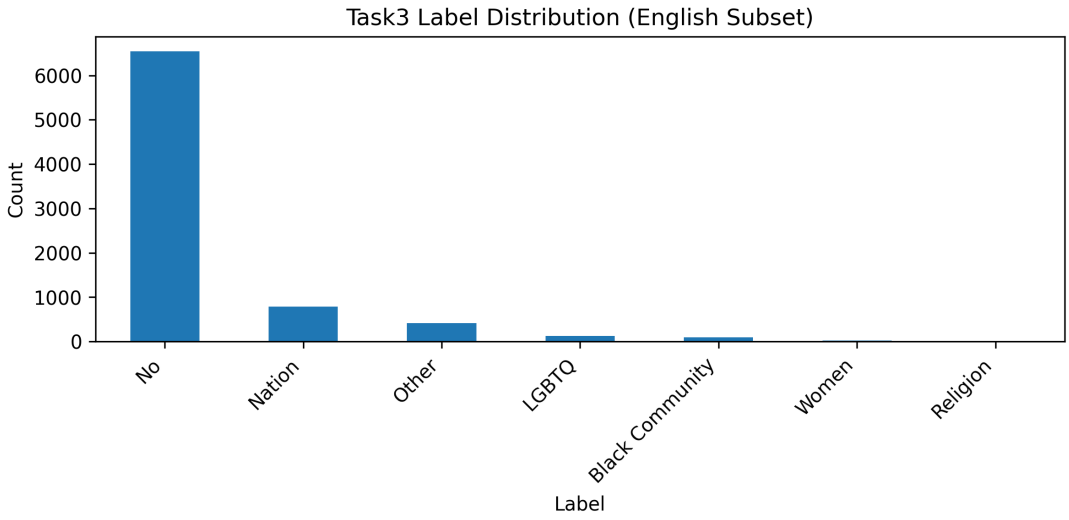


**Figure 3:** Distribution of Task3 group categories in the English subset. The dataset exhibits substantial class imbalance across target groups.

### 3.3. Comment Length Analysis

We also analyzed the distribution of comment lengths in the English subset. The average comment length is **125.6 characters**, with a median length of **87 characters**. However, the distribution exhibits a long tail, with some comments reaching up to **942 characters**.

Using the ByT5 tokenizer, the mean tokenized length is **154.7 tokens** and the median tokenized length is **116 tokens**. The longest tokenized comment contains **971 tokens**. The distribution of tokenized comment lengths is shown in Figure 4.

### 3.4. Sequence Length Considerations

To determine an appropriate maximum input length, we examined the proportion of comments truncated under different sequence limits. With a maximum input length of 384 tokens, approximately **13.09%** of

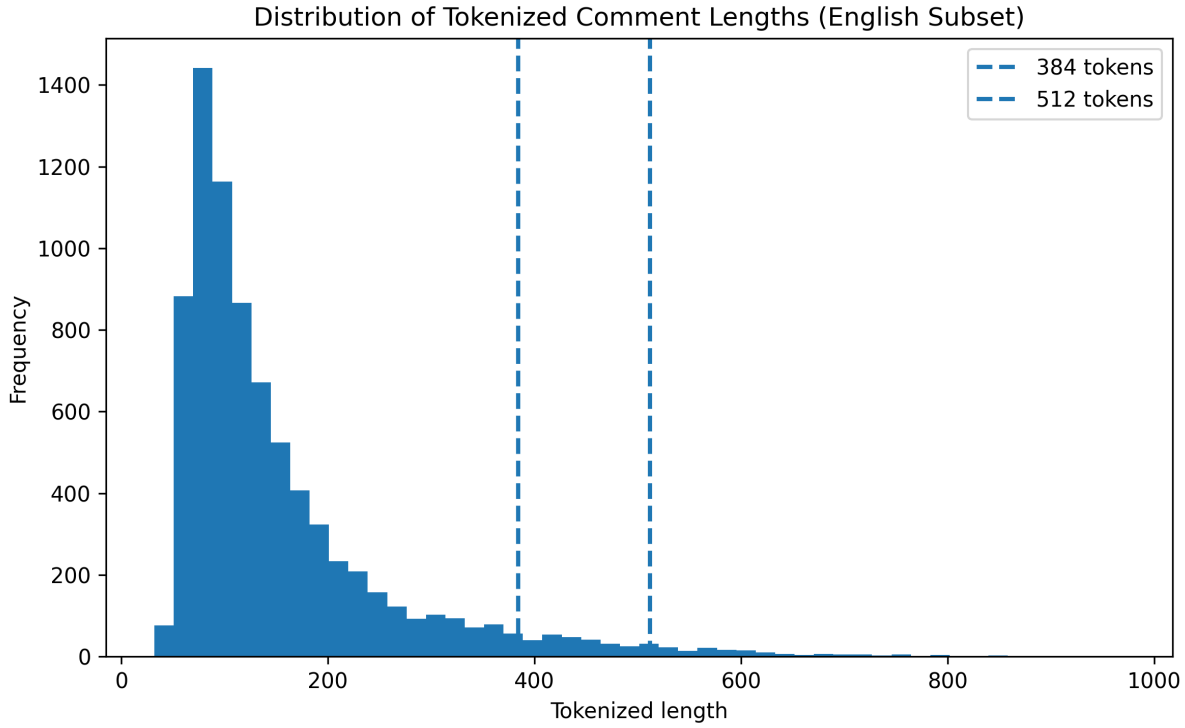


**Figure 4:** Distribution of tokenized comment lengths in the English subset using the ByT5 tokenizer. The dashed vertical lines indicate candidate maximum sequence lengths of 384 and 512 tokens.

comments (1,047 instances) would be truncated. Increasing the limit to 512 tokens reduces truncation to **5.38%** (430 comments).

Based on this analysis, we adopt a maximum sequence length that balances computational efficiency with minimizing potential information loss due to truncation.

# 4. Methodology

## 4.1. Task Formulation

The objective of this work is to detect and classify social support in social media messages. We formulate the problem as a hierarchical multi-task classification problem composed of three related tasks.

- **Task 1:** Binary classification that determines whether a message is *Supportive* or *Non-Supportive*.
- **Task 2:** For messages classified as supportive, the model predicts the type of support: *Individual*, *Group*, or *No*.
- **Task 3:** For messages where the support type corresponds to *Group*, the model identifies the specific target community (e.g., *Black Community*, *LGBTQ*, *Women*, etc.).

This hierarchical structure reflects the annotation scheme of the dataset and allows the model to capture dependencies between related classification tasks.

## 4.2. Text Representation

We employ the **ByT5 model**, a byte-level Transformer architecture that processes raw UTF-8 text directly without relying on subword tokenization [7]. Instead of operating on word pieces or subword units, ByT5 represents text as sequences of bytes, allowing the model to process arbitrary character

sequences without a fixed vocabulary. This representation is particularly well suited for social media text, which often contains informal language, spelling variations, emojis, and special characters.

Before tokenization, each message is optionally prefixed with a short instruction-style prompt:

$$\texttt{"Classify task1/task2/task3: "} + x$$

The prompt was primarily introduced to improve training stability. During preliminary experiments, we observed that including a short instruction-like prefix helped the model converge more consistently when optimizing the hierarchical multi-task objective. The prompt provides a contextual signal indicating that the input text should be interpreted as part of a classification task involving multiple prediction stages. Instruction-style prompts are commonly used in text-to-text Transformer architectures to condition the model on the intended task and improve alignment between encoder representations and downstream objectives [7, 8].

The resulting sequence is tokenized using the ByT5 tokenizer and truncated or padded to a maximum sequence length of 512 tokens during training.

### 4.3. Model Architecture

The proposed model consists of a shared encoder based on the ByT5 architecture followed by three task-specific classification heads.

Given an input message $x$, the ByT5 encoder produces contextualized hidden representations:

$$H = Encoder(x) \in \mathbb{R}^{T \times d}$$

where $T$ is the sequence length and $d$ is the hidden dimension of the encoder.

To obtain a fixed-length representation suitable for classification, we apply mean pooling over the encoder outputs using the attention mask:

$$z = \frac{\sum_{t=1}^{T} m_t h_t}{\sum_{t=1}^{T} m_t}$$

where $h_t$ denotes the encoder representation at position $t$ and $m_t$ is the attention mask indicating valid tokens. This masked mean pooling operation produces a fixed-length representation of the input sequence, $z$ represents the pooled embedding of the message.

This shared representation is then passed to three independent linear classifiers:

$$\begin{aligned} \hat{y}_1 &= W_1 z, \\ \hat{y}_2 &= W_2 z, \\ \hat{y}_3 &= W_3 z, \end{aligned} \tag{1}$$

where $W_1$, $W_2$, and $W_3$ are task-specific weight matrices corresponding to Task 1, Task 2, and Task 3 respectively.

The architecture therefore learns a shared semantic representation of the input text while allowing each task to specialize through its own classifier.

### 4.4. Hierarchical Multi-task Objective

The three tasks are trained jointly using a hierarchical multi-task learning objective. Each task corresponds to a classification problem optimized using cross-entropy loss.

Let $\hat{y}_k$ denote the predicted logits for task $k$ and $y_k$ the corresponding ground-truth label. The per-task losses are defined as

$$\begin{aligned} L_1 &= CE(\hat{y}_1, y_1), \\ L_2 &= CE(\hat{y}_2, y_2), \\ L_3 &= CE(\hat{y}_3, y_3). \end{aligned} \tag{2}$$

Because the annotation scheme is hierarchical, lower-level tasks are only meaningful when higher-level conditions are satisfied. In particular,

$$\nVdash_{sup} = \mathbf{1}(y_1 = \text{Supportive}), \qquad \nVdash_{grp} = \mathbf{1}(y_1 = \text{Supportive} \wedge y_2 = \text{Group})$$

indicate whether the corresponding tasks are applicable.

The overall training objective combines the three losses as

$$L = L_1 + \lambda_2 \nVdash_{sup} L_2 + \lambda_3 \nVdash_{grp} L_3$$

where $\lambda_2$ and $\lambda_3$ control the relative importance of deeper tasks.

We treated these coefficients as hyperparameters and explored several configurations in order to balance the influence of the deeper tasks in the hierarchy. In particular, we evaluated the combinations $(\lambda_2, \lambda_3) \in \{(0.3, 0.3), (0.3, 0.5), (0.4, 0.7), (0.3, 0.7)\}$.

The final configuration

$$\lambda_2 = 0.3, \qquad \lambda_3 = 0.7$$

was selected because it provided the most stable optimization behavior while improving performance on the deeper hierarchical tasks, particularly Task 3.

In practice, this objective is implemented using masked cross-entropy losses. Samples that do not satisfy the hierarchical conditions are excluded from the corresponding loss computation during training.

### 4.5. Handling Class Imbalance

The dataset exhibits substantial class imbalance, particularly for deeper levels of the hierarchy. To mitigate this issue we combine two complementary strategies: class-weighted loss functions and weighted sampling.

**Class-weighted loss** For each task, class weights are computed using inverse-frequency weighting

$$w_c = \frac{N}{K \cdot n_c}$$

where $n_c$ denotes the number of samples belonging to class $c$, $K$ is the number of classes, and $N$ is the total number of training samples. These weights are applied within the cross-entropy loss in order to increase the contribution of underrepresented classes during optimization.

**Weighted sampling** In addition to class-weighted losses, we modify the probability with which samples are drawn during stochastic optimization. Instead of uniform sampling, training examples are sampled according to probabilities

$$q_i = \frac{w_i}{\sum_{j=1}^{N} w_j}$$

where $w_i$ denotes a sample-specific weight.

**Task3 boosting** Samples contributing to Task 3 (i.e., YouTube comment labeled as *Supportive* and *Group*) are assigned an additional multiplicative weight

$$w_i \leftarrow w_i \cdot \gamma$$

where $\gamma$ is a boosting factor that increases the frequency with which Task3 examples appear in mini-batches.

**Rare-class adjustment** To further emphasize rare Task 3 classes we apply inverse-frequency weighting

$$w_i \leftarrow w_i \cdot \frac{N_c}{K \cdot n_c}$$

where $n_c$ denotes the number of samples in class $c$ and $N_c$ the total number of Task3 samples. To prevent instability caused by extremely rare classes the resulting weights are clipped

$$w_i \leftarrow \min(\max(w_i, 1), \tau)$$

where $\tau$ corresponds to a maximum weighting factor.

## 4.6. Training Procedure

The model is fine-tuned using the HuggingFace Transformers framework with a custom hierarchical training loop.

Optimization is performed using the AdamW optimizer with gradient clipping in order to stabilize training. Gradient accumulation is used to simulate a larger effective batch size while maintaining a small per-device memory footprint.

To improve optimization stability, a linear learning-rate warmup schedule is used during the first $6\%$ of training steps.

| Parameter | Value |
|---|---|
| Learning rate | $1 \times 10^{-4}$ |
| Per-device batch size | 4 |
| Gradient accumulation steps | 4 |
| Epochs | 4 |
| Maximum sequence length | 512 |
| Weight decay | 0.01 |
| Warmup ratio | 0.06 |
| Max gradient norm | 1.0 |

**Table 1**
Training hyperparameters

Training is performed on a GPU when available. Random seeds are fixed across Python, NumPy, and PyTorch to ensure reproducibility.

## 4.7. Evaluation

Model performance is evaluated using standard classification metrics including precision, recall, and F1-score. Evaluation is performed separately for each task.

Task 1 is evaluated on all messages in the dataset. Task 2 is evaluated only on messages labeled as supportive. Task 3 is evaluated only on messages where the support type corresponds to a group.

Confusion matrices are also reported to analyze classification errors across classes and to better understand the model's behavior across the hierarchical label structure.

## 4.8. Decision Threshold Calibration

The first stage of the hierarchical classifier predicts whether a YouTube comment is *Supportive* or *Non-Supportive*. Instead of using the default decision rule based on the maximum softmax probability (equivalent to a threshold of $0.5$), we introduce a calibrated threshold $\tau$ for the probability of the *Supportive* class.

Let $p_i$ denote the predicted probability that sample $i$ belongs to the *Supportive* class:

$$p_i = P(y_i = \text{Supportive} \mid x_i)$$

The classification decision is then defined as

$$\hat{y}_i = \begin{cases} \text{Supportive}, & \text{if } p_i \geq \tau \\ \text{Non-Supportive}, & \text{otherwise} \end{cases}$$

The threshold $\tau$ directly controls the routing of samples to downstream tasks in the hierarchical architecture. Only YouTube comments predicted as *Supportive* are passed to Task2, and only those predicted as both *Supportive* and *Group* are evaluated by Task3.

To determine an appropriate threshold, we evaluated several candidate values of $\tau \in \{0.60, 0.65, 0.70, 0.75, 0.80, 0.85\}$ on the validation set. The selected threshold balances precision and recall while maintaining a routing distribution consistent with the observed class frequencies in the dataset.

This calibration step is particularly important in hierarchical classification, as overly permissive thresholds can propagate large numbers of misclassified samples to downstream tasks, degrading overall performance. This post-training calibration step improves macro-F1 performance without altering model parameters.

Figure 5 illustrates the hierarchical routing strategy used during inference. The threshold $\tau$ controls the trade-off between recall and precision at the first stage, thereby determining how many samples are propagated to downstream tasks.

# 5. Results

## 5.1. Training Dynamics

To analyze the convergence behavior of the model, we monitored the training loss during fine-tuning. Figure 6 shows the evolution of the training loss across optimization steps.

The loss decreases rapidly during the initial training phase and stabilizes after approximately 1000 training steps, suggesting that the model successfully learns the hierarchical structure of the tasks without signs of training instability.

## 5.2. Official Evaluation Overview

The submitted system was evaluated on the official SSD-2026 English leaderboard. It achieved macro-F1 scores of 0.7412 for Subtask 1, 0.6706 for Subtask 2, and 0.4715 for Subtask 3, with an average macro-F1 of 0.6278 across the three English evaluations. The diagnostic internal/oracle tables below are included to analyze the model behavior and should be interpreted as complementary to the official leaderboard scores.

## 5.3. Task 1: Support Detection

Task 1 corresponds to binary classification between *Supportive* and *Non-Supportive* comments. Table 2 summarizes the final results on the full test set.

The confusion matrix for Task 1 is shown in Table 3, where rows correspond to observed labels and columns correspond to predicted labels.

[1]The term *oracle* is commonly used in machine learning and natural language processing to denote an evaluation setting in which part of the ground-truth information is assumed to be known. Historically, the term originates from theoretical computer science and optimization, where an *oracle* refers to an abstract entity that can provide correct answers to specific queries during an algorithm's execution. In hierarchical classification pipelines, oracle evaluation replaces the predictions of earlier stages with their true labels when evaluating downstream components. This allows researchers to measure the intrinsic performance of later classifiers independently of error propagation from upstream tasks.

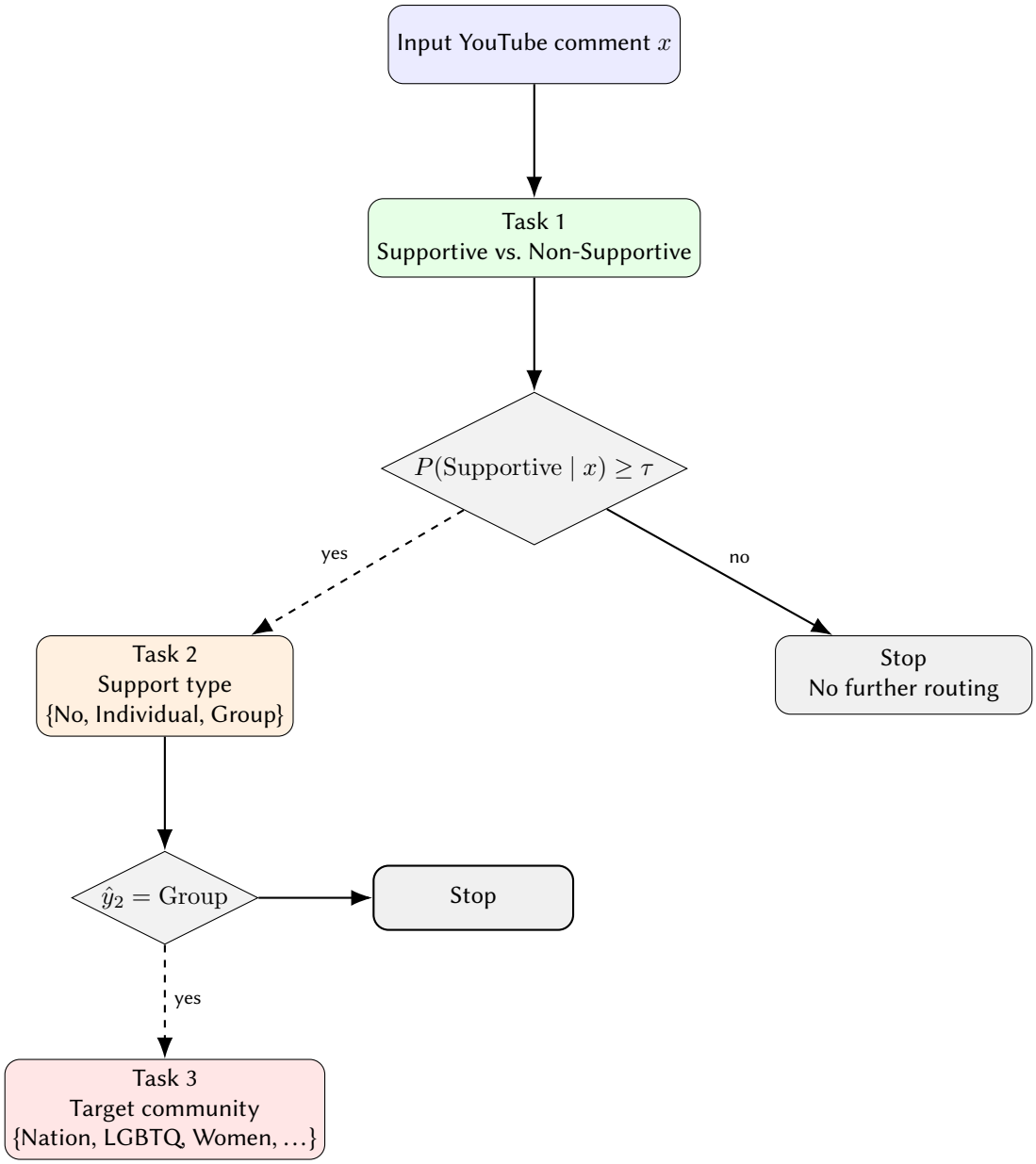


**Figure 5:** Hierarchical classification pipeline with threshold-based routing. Task 1 predicts whether a YouTube comment is supportive. Only comments whose predicted supportive probability exceeds threshold $\tau$ are routed to Task 2. Comments predicted as belonging to the *Group* category are subsequently routed to Task 3.

The macro-average F1-score for Task 1 is 0.73, which reflects the imbalance between the two classes. While the classifier performs strongly on the majority *Non-Supportive* class, the lower recall for the *Supportive* category reduces the macro-average performance. This behavior is expected given the relatively smaller number of supportive comments in the dataset.

## 5.4. Task 2: Support Target Identification

Task 2 predicts whether supportive comments target an *Individual*, a *Group*, or neither. Under oracle evaluation, Task 2 is applied only to comments whose gold Task 1 label is *Supportive*. The results are summarized in Table 4.

The confusion matrix for Task 2 is shown in Table 5.

The macro-average F1-score for Task 2 is 0.42, which is substantially lower than the weighted F1-score (0.78). This difference reflects the strong class imbalance between *Group* and *Individual* support.

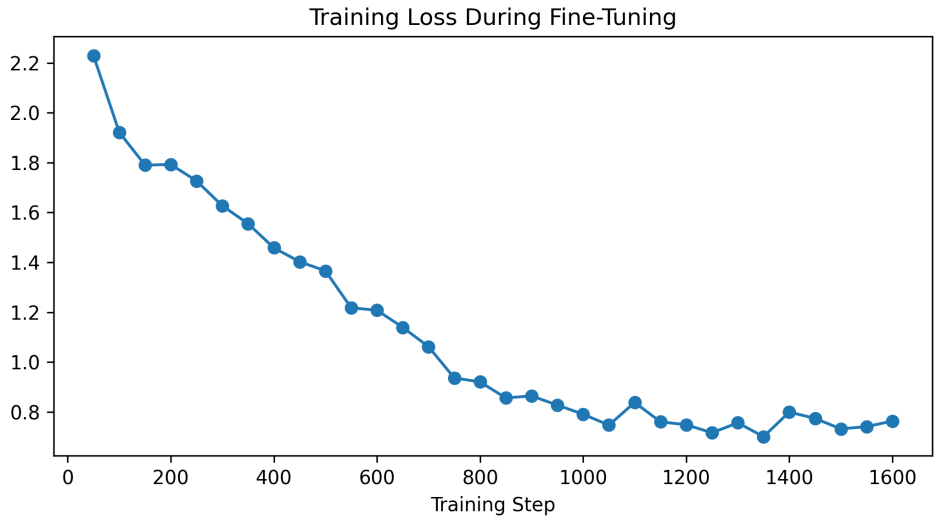


**Figure 6:** Training loss during model fine-tuning. The loss decreases steadily during optimization, indicating stable convergence of the hierarchical multi-task objective.

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Non-Supportive | 0.87 | 0.91 | 0.89 | 1250 |
| Supportive | 0.63 | 0.53 | 0.57 | 350 |
| Accuracy | | | 0.83 | |
| Macro avg | 0.75 | 0.72 | 0.73 | 1600 |
| Weighted avg | 0.82 | 0.83 | 0.82 | 1600 |

**Table 2**
Task 1 results on the full test set with decision threshold $\tau = 0.95$.

| | Predicted Non-Supportive | Predicted Supportive |
|---|---|---|
| Observed Non-Supportive | 1141 | 109 |
| Observed Supportive | 165 | 185 |

**Table 3**
Confusion matrix for Task 1 (support detection).

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| No | 0.00 | 0.00 | 0.00 | 0 |
| Individual | 0.68 | 0.25 | 0.37 | 76 |
| Group | 0.82 | 0.97 | 0.89 | 274 |
| Accuracy | | | 0.81 | |
| Macro avg | 0.50 | 0.41 | 0.42 | 350 |
| Weighted avg | 0.79 | 0.81 | 0.78 | 350 |

**Table 4**
Task 2 oracle results.

Although the model achieves high performance for the dominant *Group* category, the lower recall for the *Individual* class reduces the macro-average score.

| | Pred No | Pred Individual | Pred Group |
|---|---|---|---|
| Obs No | 0 | 0 | 0 |
| Obs Individual | 0 | 19 | 57 |
| Obs Group | 0 | 9 | 265 |

**Table 5**
Confusion matrix for Task 2 (oracle evaluation).

### 5.5. Task 3: Group Category Classification

Task 3 predicts the specific community category for comments expressing support toward a group. Under oracle evaluation, Task 3 is applied only to comments whose gold Task 1 label is *Supportive* and whose gold Task 2 label is *Group*. The results are shown in Table 6.

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Black Community | 0.58 | 1.00 | 0.73 | 11 |
| LGBTQ | 0.94 | 1.00 | 0.97 | 17 |
| Nation | 0.87 | 0.95 | 0.91 | 162 |
| Other | 0.93 | 0.67 | 0.78 | 78 |
| Religion | 0.00 | 0.00 | 0.00 | 3 |
| Women | 1.00 | 0.67 | 0.80 | 3 |
| Accuracy | | 0.86 | | |
| Macro avg | 0.62 | 0.61 | 0.60 | 274 |
| Weighted avg | 0.87 | 0.86 | 0.86 | 274 |

**Table 6**
Task 3 oracle results.

Task 3 yields the strongest downstream oracle performance, with overall accuracy of 0.86 and weighted F1-score of 0.86. The classifier performs particularly well on the *Nation* and *LGBTQ* categories, achieving F1-scores of 0.91 and 0.97 respectively. Performance on *Black Community* and *Women* is also encouraging despite their limited sample sizes. In contrast, the *Religion* category remains challenging, likely due to its extremely small number of training examples.

## 6. Discussion

### 6.1. Byte-level vs Token-level Representations

An important aspect of our approach is the use of the *ByT5* architecture, which differs substantially from the token-level transformer encoders commonly used in prior work such as BERT or RoBERTa. Traditional models rely on subword tokenization schemes (e.g., WordPiece or Byte-Pair Encoding), where the input text is first segmented into tokens drawn from a fixed vocabulary. While this representation is efficient and widely used, it can be sensitive to spelling variations, informal expressions, and non-standard characters that frequently occur in social media discourse.

By contrast, ByT5 operates directly on raw UTF-8 byte sequences, eliminating the need for subword tokenization. This design allows the model to process text in a language-agnostic manner and to capture patterns that might otherwise be fragmented by tokenization. In the context of social media comments—where spelling variations, emojis, and unconventional punctuation are common—byte-level representations may therefore provide a more robust encoding of the input text.

However, byte-level modeling also introduces challenges. Because the model operates on longer sequences of bytes rather than compressed token sequences, the effective sequence length increases and the model must learn semantic structures from a lower-level representation. As a result, training

may require stronger regularization and careful hyperparameter tuning in order to achieve stable convergence.

### 6.2. Performance of the Deepest Classification Task

One notable result of our experiments is the relatively strong performance observed for Task 3, the group category classification problem. Under oracle evaluation, this task achieves the highest overall F1-score among the downstream tasks.

At first glance this may appear counterintuitive, since Task 3 represents the most fine-grained classification problem in the hierarchy. However, this outcome can be explained by two factors. First, Task 3 operates on a much smaller and more homogeneous subset of the data—only comments that are both supportive and directed toward a group. This filtering effect significantly reduces the diversity of the input distribution, making the classification task more focused.

Second, the semantic distinctions between several of the group categories are relatively well defined. Categories such as *Nation*, *LGBTQ*, or *Black Community* often appear in comments with explicit lexical markers, which can facilitate discrimination by the model once the correct subset of comments has been identified.

Nevertheless, performance varies considerably across classes. Categories with very few training examples, such as *Religion* or *Women*, remain difficult to learn reliably.

### 6.3. Limits of Rebalancing for Extremely Small Classes

Class imbalance is a central challenge in the dataset used for this task. While techniques such as class weighting, oversampling, or loss rebalancing are commonly used to mitigate imbalance, these approaches have limited effectiveness when the minority classes contain extremely few examples.

In such cases, the problem is not only one of imbalance but also one of insufficient information. When only a handful of labeled instances are available, the model cannot reliably estimate the underlying distribution of the class regardless of the weighting scheme applied during training. As a consequence, the model may either fail to learn meaningful patterns for these classes or may overfit to the few available examples.

This phenomenon is visible in our results for the smallest Task 3 categories, where even substantial weighting adjustments did not lead to stable improvements in F1-score. These findings suggest that addressing the long-tail distribution of social support categories may require additional strategies, such as targeted data augmentation or the collection of additional annotated examples.

### 6.4. Training Stability Under Class Imbalance

Training deep models under severe class imbalance can lead to unstable optimization behavior, particularly in multi-task settings where multiple losses interact during learning. In our experiments, several steps were taken to stabilize training.

First, the hierarchical loss formulation ensures that deeper tasks contribute to the optimization only when the corresponding higher-level conditions are satisfied. This prevents the model from attempting to learn downstream classifications for samples where those labels are not applicable.

Second, task-specific weighting coefficients were introduced to balance the influence of each level of the hierarchy during optimization. By assigning lower weight to intermediate tasks and higher weight to the most specific classification task, the model is encouraged to learn progressively specialized representations without overwhelming the shared encoder.

Finally, careful threshold calibration was applied during evaluation in order to control the precision–recall trade-off of the first classification stage. Because routing decisions depend heavily on Task 1 predictions, adjusting the decision threshold proved essential for stabilizing the behavior of the overall hierarchical pipeline.

Together, these strategies helped mitigate some of the instability introduced by the strongly imbalanced class distributions present in the dataset.

### 6.5. Comparison with Previous Work

It is also informative to compare our findings with the results reported by Kolesnikova et al. [6], who evaluate several machine learning and transformer-based approaches for the same general social support detection problem. In their experiments, the best-performing model—based on a RoBERTa encoder—achieves macro F1 scores of approximately 0.78 for support detection (Task 1), 0.84 for support target identification (Task 2), and 0.80 for group category classification (Task 3).

Direct comparison between these results and those reported in this study should be interpreted with caution. Although both works rely on the same annotation framework and originate from the same dataset family, the experimental subsets are not identical. In particular, the present study focuses exclusively on the English subset of the dataset, whereas the experiments in [6] are conducted on a broader collection of annotated comments. Differences in sampling, preprocessing, and class distributions may therefore influence the reported performance metrics.

Nevertheless, several observations can be made. First, the general pattern of performance across the hierarchy is consistent with prior work: the initial support detection task is relatively stable, while deeper levels of the hierarchy are more sensitive to data sparsity and class imbalance. Second, the results confirm the effectiveness of transformer-based encoders for modeling supportive discourse in social media texts.

The main difference between the two approaches lies in the modeling strategy. While the models evaluated by Kolesnikova et al. typically treat the tasks independently or apply them sequentially in a pipeline, our method explicitly incorporates the hierarchical structure of the annotation scheme into the training objective. By optimizing all tasks jointly using a shared encoder and conditional loss masking, the proposed approach encourages the model to learn representations that respect the dependency structure between support detection, support target identification, and group category classification.

In addition, our work explores the use of a byte-level transformer encoder (ByT5), which processes raw UTF-8 text directly rather than relying on subword tokenization. This design choice is particularly relevant for social media data, where spelling variation, informal language, and unconventional punctuation can challenge token-based models. The results presented in this study therefore provide complementary evidence that byte-level architectures can be competitive for hierarchical social support detection tasks.

## 7. Conclusion and Future Work

This work explored hierarchical social support detection using a multi-task learning framework based on the ByT5 architecture. The proposed approach models the annotation hierarchy explicitly by jointly optimizing three classification tasks: support detection, support target identification, and group category classification. By introducing conditional loss masking and task-weighted optimization, the model learns progressively more specialized representations while respecting the hierarchical dependencies between tasks.

Experimental results show that the model achieves stable performance across the three levels of the hierarchy despite the strong class imbalance present in the dataset. The results highlight the importance of carefully designing the training objective to prevent deeper tasks from introducing instability during optimization. The use of a byte-level encoder also proved advantageous for handling the noisy and informal language characteristic of social media comments, where spelling variations, emojis, and unconventional punctuation frequently occur.

Our analysis further suggests that while transformer encoders can effectively capture supportive discourse patterns, performance remains sensitive to class imbalance and data sparsity in the deeper levels of the hierarchy. Extremely small classes remain difficult to learn reliably even when class rebalancing or loss weighting techniques are applied, indicating that in such cases the main limitation is often the scarcity of informative examples rather than imbalance alone.

Several directions for future work emerge from this study. First, it would be valuable to investigate whether combining a token-based encoder such as BERT or RoBERTa with the same hierarchical

multi-task learning framework could yield improved performance. Token-level encoders may capture higher-level semantic structures more efficiently, and a systematic comparison with the byte-level ByT5 architecture could provide further insight into the trade-offs between token-level and byte-level representations for social media text classification.

Second, an interesting extension would be to train the model jointly on both English and Spanish comments. Because ByT5 operates directly on raw UTF-8 byte sequences and does not rely on language-specific tokenization, it is naturally suited for multilingual settings. Mixing the English and Spanish subsets of the dataset could increase the effective amount of training data while enabling the model to learn cross-lingual patterns of supportive discourse.

Another promising direction concerns the routing mechanism used in the hierarchical classification pipeline. In this work, routing decisions are controlled through calibrated decision thresholds applied to the predictions of the first classification stage. Future research could explore more sophisticated strategies for optimal routing, including probabilistic routing schemes, learned routing policies, or architectures that jointly optimize routing decisions and downstream classifiers. Such approaches could potentially improve the stability and efficiency of hierarchical classification pipelines.

An additional avenue for future work is to explore tighter integration between routing decisions and the hierarchical training objective. In the current implementation, routing is controlled through calibrated thresholds while the hierarchical loss uses task weights $\lambda_2$ and $\lambda_3$ to regulate the influence of deeper tasks during training. A more unified framework could jointly optimize routing policies and task-specific loss weights, allowing the model to learn how uncertainty in earlier predictions should propagate through the hierarchy. Such approaches may further improve both training stability and overall performance in hierarchical classification problems.

Overall, the combination of hierarchical multi-task learning, calibrated routing strategies, and byte-level text representations offers a promising direction for building robust models of supportive discourse in social media environments.

## 8. Limitations

This study is subject to several limitations. First, the dataset contains extremely small classes in the deeper levels of the hierarchy. In these cases, the number of available examples is often too limited for the model to learn reliable patterns, and techniques such as class weighting or re-balancing cannot fully compensate for the lack of data.

Second, the experiments were conducted only on the English subset of the dataset. While this simplifies the experimental setting, it prevents us from evaluating the potential benefits of multilingual training.

Finally, routing decisions in the hierarchical pipeline are currently controlled through calibrated thresholds rather than learned routing policies, which may limit the optimality of the overall decision process.

## Declaration on Generative AI

During the preparation of this work, the author used Claude (Anthropic) and ChatGPT (OpenAI) in order to: assist with editing, language refinement, and LaTeX formatting; clarify technical concepts; and provide programming assistance in implementing the author's own model design in Python. The methodological decisions, including the choice of the ByT5 architecture, the hierarchical classification structure, and the loss formulation, were made by the author. After using these tools, the author reviewed and edited all content and code and takes full responsibility for the publication's content.

**Acknowledgments**

The author thanks the SSD-2026 and IberLEF 2026 organizers for providing the shared-task dataset, task definition, and evaluation framework.

## References


[1] M. Shahiki Tash, L. Ramos, A. Zahra, A. Y. Barrera Animas, F. Balouchzahi, O. Kolesnikova, G. Sidorov, A. Gelbukh, R. Monroy, F. Vargas, Overview of SSD-2026 at IberLEF 2026: Social Support Detection, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2026), co-located with the 42nd Conference of the Spanish Society for Natural Language Processing (SEPLN 2026), CEUR-WS.org, 2026.

[2] A. Bonet-Jover, J. A. González-Barba, L. Chiruzzo, Overview of IberLEF 2026: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2026), co-located with the 42nd Conference of the Spanish Society for Natural Language Processing (SEPLN 2026), CEUR-WS.org, 2026.

[3] Z. Ahani, M. S. Tash, F. Balouchzahi, L. Ramos, G. Sidorov, A. Gelbukh, Social support detection from social media texts, PLOS ONE (2026). doi:10.1371/journal.pone.0337476.

[4] M. Shahiki-Tash, Z. Ahani, L. Ramos, et al., Linguistic analysis in different types of support using LIWC for Spanish and English online communities, Research Square preprint (2026). doi:10.21203/rs.3.rs-9557287/v1.

[5] M. S. Tash, L. Ramos, Z. Ahani, R. Monroy, H. Calvo, G. Sidorov, Online social support detection in spanish social media texts, arXiv preprint arXiv:2502.09640 (2025).

[6] O. Kolesnikova, M. S. Tash, Z. Ahani, A. Agrawal, R. Monroy, G. Sidorov, Advanced machine learning techniques for social support detection on social media, Heliyon 11 (2025).

[7] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, C. Raffel, Byt5: Towards a token-free future with pre-trained byte-to-byte models, Transactions of the Association for Computational Linguistics 10 (2022) 291–306.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67.