# SCaLAR@Social Support Detection at IberLEF 2026 (SSD-2026): A Sequential DeBERTa-Based Pipeline with Probability Chaining and Lexical Augmentation for Hierarchical Text Classification

Shaurya Rohiet[1,*,†], Prathamesh Pradeep Khunte[1,*,†] and Anand Kumar M[1,*,†]

[1]*Department of Information Technology, National Institute of Technology Karnataka, Surathkal, India*

**Abstract**

Hierarchical text classification on social media presents significant challenges due to noisy user-generated content, severe class imbalance, and structurally dependent label hierarchies. This paper proposes a sequential transformer-based pipeline that decomposes the three-level annotation structure into three causally ordered stages: supportive intent detection (Task 1), individual vs. group classification (Task 2), and target group identification (Task 3). The system is built on DeBERTa-v3-base and replaces conventional CLS-based pooling with a Global Attention Pooling mechanism, enabling a learned weighted aggregation of token representations for improved contextual encoding. To address class imbalance, the framework integrates Focal Loss ($\gamma = 2$), a WeightedRandomSampler for balanced training, and data-level augmentation through Easy Data Augmentation (EDA). A key component of the architecture is probability chaining, where predicted probabilities from upstream tasks are incorporated as auxiliary features into downstream classifiers, modeling inter-task dependencies in the hierarchical structure. Additionally, for the most under-represented target groups in Task 3, the generic augmentation is replaced by a semantic-enrichment and WordNet-based lexical augmentation strategy that appends label-aware semantic cues to each post before synonym substitution, enhancing minority class representation while preserving label semantics. The proposed system achieves macro-F1 scores of 0.7937, 0.8384, and 0.6806 for Tasks 1, 2, and 3 respectively, with an average validation macro-F1 of 0.7709, indicating competitive performance on the higher-level tasks while also highlighting the increased difficulty of fine-grained classification in highly imbalanced settings. On the official SSD-2026 shared-task test set, the system attains per-task leaderboard rankings of 17, 18, and 17 for Tasks 1, 2, and 3 respectively, with corresponding macro-F1 scores of 0.8027, 0.7645, and 0.4765.

**Keywords**

Hierarchical Text Classification, DeBERTa, Social Support Detection, Probability Chaining, Semantic Enrichment, Focal Loss

## 1. Introduction

The proliferation of user-generated content on social media platforms has led to increased research interest in automated text understanding, particularly in domains involving nuanced socio-linguistic phenomena. Hierarchical text classification, where labels exhibit structured dependencies rather than forming a flat taxonomy, represents a particularly challenging formulation of this problem. In the target task [8, 9, 10], the annotation schema consists of three causally linked sub-tasks: (1) binary detection of supportive intent, (2) conditional classification of scope (individual vs. group), and (3) multi-class prediction of the target demographic group. This sequential dependency implies that errors at any stage directly affect downstream predictions.

Four key challenges characterize this setting. First, hierarchical dependency: predictions must respect the causal structure of the label space, making independent or parallel modeling suboptimal. Second, severe class imbalance: minority classes, particularly the Supportive category and specific demographic

*IberLEF 2026, León, Spain*

*Corresponding author.

†These authors contributed equally.

shauryarohiet.231it066@nitk.edu.in (S. Rohiet); prathameshk.231it051@nitk.edu.in (P. P. Khunte); m_anandkumar@nitk.edu.in (A. K. M)

0009-0007-8614-5065 (S. Rohiet); 0009-0003-5596-0463 (P. P. Khunte); 0000-0003-0310-4510 (A. K. M)

groups such as Women, Religion, Black Community, and LGBTQ, are underrepresented, leading to biased model behavior without corrective measures. Third, lexical noise: social media text frequently includes informal expressions, emojis, hashtags, and irregular syntax, which complicate representation learning. Fourth, error propagation: the sequential nature of the pipeline amplifies the impact of upstream misclassifications, resulting in compounded errors across tasks.

To address these challenges, this paper proposes a unified sequential classification framework built on DeBERTa-v3-base. The core architectural contribution is probability chaining, wherein the predicted probabilities from upstream tasks are incorporated as auxiliary inputs to downstream classifiers, explicitly modeling inter-task dependencies. To mitigate class imbalance, the system integrates Focal Loss, a WeightedRandomSampler, and Easy Data Augmentation (EDA) techniques. Additionally, for the most underrepresented Task 3 target groups, the generic augmentation is replaced by a semantic-enrichment step followed by WordNet-based lexical synonym augmentation, which improves minority class representation while preserving label semantics. Finally, task-specific threshold tuning is performed for binary classification stages to optimize decision boundaries.

## 2. Related Work

### 2.1. Transformer Models for Text Classification

The introduction of BERT [1] demonstrated that pre-training deep bidirectional transformers on large unlabeled corpora followed by task-specific fine-tuning achieves state-of-the-art performance across a wide range of natural language processing tasks. Subsequent models have refined this paradigm. RoBERTa improved training strategies through dynamic masking and larger batch sizes, while ALBERT introduced parameter sharing and embedding factorization to reduce computational cost. More recently, DeBERTa [2] proposed disentangled attention, separating content and positional information during attention computation. This modification has been shown to improve performance on classification tasks, particularly for short and noisy text such as social media data. In this work, DeBERTa-v3-base is selected due to its favorable trade-off between performance and computational efficiency compared to larger alternatives such as RoBERTa-large.

### 2.2. Class Imbalance in NLP

Class imbalance is a well-known challenge in text classification, often leading to biased models that favor majority classes. Focal Loss [3] addresses this issue by down-weighting easy examples and focusing training on harder, misclassified samples through a modulating factor. Sampling-based approaches, such as WeightedRandomSampler, rebalance class distributions during training by assigning higher sampling probability to minority-class instances. Data augmentation techniques further alleviate imbalance. EDA [4] introduces simple yet effective transformations including synonym replacement, random insertion, swapping, and deletion, which have shown consistent improvements in low-resource and imbalanced settings. Our approach combines these complementary strategies, loss reweighting, sampling, and augmentation, to improve robustness against skewed class distributions.

### 2.3. Hierarchical Text Classification

Hierarchical classification arises in applications where label structures exhibit dependencies across multiple levels [5]. Traditional approaches either flatten the hierarchy, thereby ignoring inter-label relationships, or train independent classifiers at each level, which limits information sharing across tasks. More advanced methods include local classifier-per-node strategies and global architectures that encode hierarchical constraints directly within the model [6]. Cascade-based approaches explicitly model inter-level dependencies by passing predictions from one level to the next. Our work follows this paradigm but extends it through a probability chaining mechanism, where the full softmax probability

distribution from upstream tasks is propagated as a continuous feature vector rather than using only hard labels. This preserves uncertainty information and enables more informed downstream predictions.

### 2.4. Data Augmentation for Imbalanced Text Classification

Large language models have recently been explored as a source of synthetic training data for NLP tasks [7]. While such models can generate diverse and contextually rich samples, unconstrained generation often introduces issues such as label inconsistency, hallucination, and distributional drift. Verification stages such as self-consistency checks and discriminator-based filtering have been proposed in prior work to mitigate these issues. Rather than relying on LLM-generated samples, in this work we adopt the semantic-enrichment and lexical-augmentation approach of Chi et al. [14] for the most underrepresented Task 3 classes. Class labels are first used to append a representative semantic cue to each training sentence, after which WordNet-based synonym replacement increases lexical diversity. This lightweight, label-aware strategy improves minority class representation while maintaining semantic and dataset integrity [11, 12, 13].

## 3. Methodology

### 3.1. Problem Formulation

Let $x$ denote an input social media post represented as a sequence of tokens. The hierarchical classification objective is to learn a structured prediction function $f : \mathcal{X} \to \mathcal{Y}_1 \times \mathcal{Y}_2 \times \mathcal{Y}_3$, where:

- $Y_1 \in \{0, 1\}$ (Non-Supportive, Supportive)
- $Y_2 \in \{0, 1\}$ (Individual, Group) [valid iff $Y_1 = 1$]
- $Y_3 \in \{0, 1, 2, 3, 4, 5\}$ (Nation, Other, LGBTQ, Black Community, Religion, Women) [valid iff $Y_2 = 1$]

The hierarchical validity constraint, downstream labels are meaningful only under specific upstream conditions, is encoded in training data filtering (Task 2 trains only on Supportive rows; Task 3 trains only on Supportive, Group rows) and in inference (non-supportive predictions collapse downstream tasks to null).

### 3.2. Text Preprocessing Pipeline

Raw social media text is subjected to a noise-reduction pipeline prior to tokenization. Contractions are expanded (e.g., "don't" → "do not") using rule-based normalization. Emoji are demojized into descriptive tokens (e.g., the smiling face emoji → `:smiling_face:`), preserving affective information in the text surface. URLs and @-mentions are stripped, as they carry no generalizable semantic content. Hashtags are segmented into their constituent words (`#BlackLivesMatter` → `BlackLivesMatter`). HTML entities and non-ASCII characters are normalized. Repeated characters are collapsed to at most two occurrences (`loooove` → `loove`). Numeric tokens are removed, and the resulting string is lowercased and whitespace-normalized. Rows yielding fewer than four tokens after cleaning are discarded, as are exact duplicates on the (text, task1) key.

### 3.3. DeBERTa-v3-base with Global Attention Pooling

The base encoder for all three tasks is `microsoft/deberta-v3-base`, a 12-layer transformer with hidden dimension $H = 768$. DeBERTa employs disentangled attention in which the attention weight between tokens $i$ and $j$ is decomposed as:

$$A(i, j) = A_c(i, j) + A_p(i, j) + A_{cp}(i, j) \tag{1}$$

where $A_c(i, j)$ is the content-to-content attention score, $A_p(i, j)$ is the relative-position-to-content score, and $A_{cp}(i, j)$ is the content-to-position score. This decomposition enables DeBERTa to separately

model what a token means and where it appears, which is particularly valuable for short, positionally expressive social media posts.

Rather than using the [CLS] token embedding as the sequence representation, which may not adequately capture distributional information for social media posts of varying length, we implement a Global Attention Pooling (GAP) layer. The GAP layer computes a learned weighted sum over all token representations in the final encoder layer:

$$h_{pool} = \sum_i \alpha_i \cdot h_i, \qquad \alpha_i = \text{softmax}(\text{MLP}(h_i)) \tag{2}$$

where $h_i \in \mathbb{R}^H$ is the hidden state at position $i$, and MLP is a two-layer feedforward network ($H \to H/2 \to 1$) with $\tanh$ activation. Attention weights are masked to zero for padding positions before the softmax normalization, ensuring that padding tokens do not contribute to the pooled representation. A dropout layer ($p = 0.2$) is applied to the pooled vector prior to the classification head.

### 3.4. Focal Loss

Standard cross-entropy loss treats each training sample equally, which causes gradient updates to be dominated by the numerous majority-class examples, suppressing learning of minority classes. Focal Loss [3] addresses this by introducing a modulating factor:

$$\text{FL}(p_t) = -(1 - p_t)^\gamma \cdot \log(p_t) \tag{3}$$

where $p_t$ is the model's estimated probability for the ground-truth class, and $\gamma$ is the focusing parameter. At $\gamma = 0$, FL reduces to standard cross-entropy. As $\gamma$ increases, the contribution of correctly classified easy examples is progressively down-weighted. We use $\gamma = 2$ throughout, consistent with [3]. Additionally, per-class weights $w_k$ inversely proportional to class frequency are incorporated into the loss: $w_k = 1/\text{freq}(k)$, normalized so that $\sum w_k/K = 1$. These weights are passed to the cross-entropy component and thus scale the focal modulation. The combination of frequency-based weighting and focal modulation provides complementary imbalance correction mechanisms.

### 3.5. WeightedRandomSampler

Beyond the loss-level correction of Focal Loss, we apply a data-level correction via WeightedRandomSampler. Each training sample $i$ is assigned a sampling weight:

$$s_i = 1/\text{freq}(\text{class}(i)) \tag{4}$$

The sampler draws a fixed number of samples per epoch with replacement according to the normalized distribution over $\{s_i\}$. This ensures that each mini-batch contains a roughly uniform class distribution irrespective of the original corpus imbalance. The dual application of Focal Loss (loss-level) and WeightedRandomSampler (data-level) creates a mutually reinforcing correction signal, the sampler ensures minority examples are frequently presented, while Focal Loss focuses gradient on the most informative among those presented.

### 3.6. Easy Data Augmentation (EDA)

We apply EDA [4] to the minority class in each task to further alleviate imbalance. For each minority-class training example, up to three augmented variants are generated, each produced by randomly selecting one of three operations: (a) Synonym Replacement, for each content word not in the stopword list, with probability proportional to its selection, a synonym from WordNet is substituted; (b) Random Swap, two tokens at random positions are exchanged; (c) Random Deletion, one token at a randomly selected position is removed. The augmentation ratio is controlled by a `target_ratio` parameter: minority-class samples are augmented until their count reaches `target_ratio` $\times$ `majority_count` (85% for

Task 1, 90% for Task 2). For Task 3, this generic EDA strategy is replaced by the semantic-enrichment and lexical-augmentation methodology of Chi et al. [14]. First, semantic enrichment is applied: the Task 3 target label is cleaned and a representative label word is appended to each training sentence, injecting label-aware semantic cues into the surface form (e.g., "hate these immigrants" becomes "hate these immigrants immigrant"). The enriched minority-class sentences are then augmented through WordNet-based lexical synonym replacement, in which eligible content words are substituted with their WordNet synonyms to increase lexical diversity while preserving semantics. Semantic enrichment is always applied before synonym augmentation, and the remainder of the Task 3 training pipeline is unchanged.

### 3.7. LLM-Based Data Generation and Verification

To further improve the representation of minority classes, we employ a two-stage LLM-assisted augmentation pipeline. In the generation stage, an instruction-tuned large language model is prompted to produce realistic Twitter/X-style posts following the dataset schema (`id`, `text`, `task1`, `task2`, `task3`). The implementation supports either Groq Llama-3-8B-8192 [19] or OpenAI GPT-4o-mini [18], depending on API availability, while using identical prompts and processing for both. Generation targets the four underrepresented Task-3 categories (Women, Religion, Black Community, and LGBTQ), with 20 samples generated per category (80 samples in total). A temperature of 0.85 is used during generation to encourage linguistic diversity while maintaining label consistency.

In the second stage, every synthetic sample is independently verified by the same model using a separate verification prompt that evaluates (i) consistency between the text and its assigned labels and (ii) grammatical coherence. Verification is performed deterministically (temperature $= 0.0$), and only samples receiving positive verification are retained. The resulting verified synthetic dataset is then merged with the original training data and used during model training. This two-stage generation-and-verification design is intended to improve the quality of synthetic data while minimizing label noise.

**Table 1**
Configuration of the LLM-based generation and verification pipeline.

| Parameter | Value |
|---|---|
| Generation model | OpenAI GPT-4o-mini or Groq Llama-3-8B-8192 |
| Verification model | Same model as generation |
| Generation temperature | 0.85 |
| Verification temperature | 0.0 |
| Samples per minority class | 20 |
| Target classes | Women, Religion, Black Community, LGBTQ |
| Output format | CSV (`id`, `text`, `task1`, `task2`, `task3`) |

For full reproducibility, we report the exact prompts and configuration used by the LLM-assisted generation and verification pipeline. The generation stage uses the following system and user prompts:

```
[System]
You are assisting with social support detection dataset augmentation. Generate
realistic Twitter/X-style posts that match the requested target
labels. Requirements:
- Produce natural, diverse, human-like social media text.
- Do not repeat existing examples.
- Keep each post concise (approximately 10-40 words).
- Preserve the assigned labels exactly.
- Avoid emojis unless they fit naturally.
- Avoid hashtags unless appropriate.
- Do not number the outputs.
- Return the output only as CSV.
CSV format: id,text,task1,task2,task3

[User]
Generate {N} new Twitter/X posts belonging to the following category.
Task 1: Supportive
Task 2: Group
Task 3: {TARGET_GROUP}
The generated examples should: be grammatically correct, resemble
genuine social media posts, contain varied wording, avoid duplicates,
and maintain the assigned labels. Return only CSV rows.
```

Listing 1: Generation prompt (system and user).

Here $N = 20$ samples are requested per call and $\text{TARGET\_GROUP} \in \{\text{Women}, \text{Religion}, \text{Black Community}, \text{LGBTQ}\}$, with Task 1 fixed to *Supportive* and Task 2 fixed to *Group*. Each generated sample is then independently verified using the following prompt:

```
[System]
You are an expert dataset annotator. Your task is to verify whether a
synthetic social media post matches its assigned labels and whether the
text is grammatically coherent. Check:
1. Does the text agree with Task1, Task2 and Task3 labels?
2. Is the text natural and grammatically correct?
Respond only with YES or NO followed by a very brief explanation.

[User]
Text: "{TEXT}"
Assigned labels: Task1: {TASK1}  Task2: {TASK2}  Task3: {TASK3}
Should this example be accepted?
```

Listing 2: Verification prompt (system and user).

The generation and verification models are OpenAI GPT-4o-mini or Groq Llama-3-8B-8192, with the same model used for both stages. Generation uses a temperature of $0.85$ to encourage linguistic diversity, while verification is performed deterministically at temperature $0.0$; only samples receiving a YES verdict are retained and merged with the original training data.

### 3.8. Probability Chaining

A central design choice in our pipeline is probability chaining, the use of upstream classifier probability outputs as explicit auxiliary features for downstream classifiers. After Task 1 training, we compute:

$$p_1 = P(Y_1 = 1 \mid x) \in [0, 1] \tag{5}$$

This scalar is appended to the GAP-pooled representation when training and running inference for Task 2:

$$h_2 = [h_{pool}; p_1] \in \mathbb{R}^{H+1} \tag{6}$$

Similarly, Task 3 receives both upstream probabilities:

$$p_2 = P(Y_2 = 1 \mid x, p_1) \in [0, 1], \qquad h_3 = [h_{pool}; p_1; p_2] \in \mathbb{R}^{H+2} \tag{7}$$

Although probability chaining is theoretically attractive for preserving upstream prediction uncertainty—a $p_1 = 0.52$ carries different information from $p_1 = 0.98$, even though both yield label $\hat{Y}_1 = 1$—our ablation suggests that in the current setting its practical benefit is limited, with validation performance remaining comparable to, and in some configurations below, that of the naive label-chaining baseline.

### 3.9. Binary Decision Threshold Tuning

Standard classification practice uses 0.5 as the decision boundary for binary tasks. However, under class imbalance, the optimal decision threshold for macro-F1 typically deviates from 0.5. We tune the threshold for Tasks 1 and 2 independently on the validation set by an exhaustive grid search over the interval $[0.20, 0.80]$ in 121 uniformly spaced steps. The threshold $\tau^*$ that maximizes validation macro-F1 is selected for inference:

$$\tau^* = \arg\max_{\tau \in [0.20, 0.80]} F1_{\text{macro}}(\hat{y} = \mathbf{1}[p \geq \tau], y_{\text{val}}) \tag{8}$$

This threshold is recorded alongside the per-task validation F1 scores to ensure full reproducibility.

## 4. Pipeline Architecture

The full end-to-end pipeline is organized as follows. During training, three SeqClassifier models are trained sequentially. Task 1 receives the cleaned text as input. Task 2 is trained exclusively on samples where $Y_1 = 1$ (Supportive), and takes as input the cleaned text along with the Task 1 probability $p_1$ as an auxiliary feature. Task 3 is trained exclusively on samples where $Y_1 = 1 \wedge Y_2 = 1$ (Supportive and Group), and receives the cleaned text along with both $p_1$ and $p_2$. Each model is saved to disk immediately after training, allowing kernel restarts between tasks without loss of progress.

During inference, hierarchical consistency is strictly enforced. If $\hat{Y}_1 = 0$ (Non-Supportive), then $\hat{Y}_2$ and $\hat{Y}_3$ are set to null irrespective of their raw predictions. If $\hat{Y}_1 = 1$ but $\hat{Y}_2 = 0$ (Individual), then $\hat{Y}_3$ is set to null. This ensures that downstream predictions are only produced when semantically valid. The Task 3 training data is first prepared through semantic enrichment and WordNet-based synonym augmentation, after which the three task-specific models are trained sequentially to produce the final submission outputs.

**Table 2**
Pipeline architecture overview.

| Stage | Input Features | Output | synthetic training data **Training Subset** |
|---|---|---|---|
| Task 1 | $x_{\text{clean}}$ | $p_1 \in [0, 1]$ | All samples |
| Task 2 | $x_{\text{clean}} + p_1$ | $p_2 \in [0, 1]$ | $Y_1 = 1$ |
| Task 3 | $x_{\text{clean}} + p_1 + p_2$ | $\hat{Y}_3 \in \{0, \ldots, 5\}$ | $Y_1 = 1,\ Y_2 = 1$ |

## 5. Experimental Setup

All experiments use DeBERTa-v3-base (`microsoft/deberta-v3-base`) as the transformer backbone. The maximum token sequence length is set to 160, which covers approximately the 95th percentile of cleaned post lengths. Training is performed using the AdamW optimizer with weight decay of 0.01. Task-specific learning rates are used: $2 \times 10^{-5}$ for Task 1 and Task 2, and $1.8 \times 10^{-5}$ for Task 3. Gradient accumulation is applied over 2 steps, resulting in effective batch sizes of 16 for Task 1 and Task 2, and 12 for Task 3. A linear warmup schedule is used over the first 10% of training steps, followed by linear decay to zero. Gradient norms are clipped at 1.0, and Automatic Mixed Precision (AMP) is enabled for efficient training on CUDA devices. All training and inference were carried out on a single

NVIDIA GeForce RTX 4070 GPU (8 GB VRAM) with 16 GB of system RAM, with total training time of approximately 6 hours and 17 minutes across the three subtasks.

Focal Loss with $\gamma = 2$ is used across all tasks to address class imbalance, in conjunction with weighted sampling. Task 1 and Task 2 are trained for up to 5 epochs, while Task 3 is trained for up to 7 epochs. Early stopping with a patience of 2 is applied based on validation macro-F1. The dataset is split into 85% training and 15% validation sets, stratified on Task 1 labels to preserve class distribution. All experiments are conducted with a fixed random seed of 42 for reproducibility. For Task 3, the batch size is reduced to 6 to accommodate the multi-class classification head within GPU memory constraints.

**Table 3**
Experimental hyperparameters across the three tasks.

| Hyperparameter | Task 1 | Task 2 | Task 3 |
|---|---|---|---|
| Learning Rate | $2 \times 10^{-5}$ | $2 \times 10^{-5}$ | $1.8 \times 10^{-5}$ |
| Batch Size | 8 | 8 | 6 |
| Gradient Accumulation | 2 | 2 | 2 |
| Effective Batch Size | 16 | 16 | 12 |
| Max Sequence Length | 160 | 160 | 160 |
| Max Epochs | 5 | 5 | 7 |
| Early Stopping Patience | 2 | 2 | 2 |
| Warmup Fraction | 10% | 10% | 10% |
| Weight Decay | 0.01 | 0.01 | 0.01 |
| Gradient Clipping | 1.0 | 1.0 | 1.0 |
| Loss Function | Focal ($\gamma = 2$) | Focal ($\gamma = 2$) | Focal ($\gamma = 2$) |

## 5.1. Ablation Study

To quantify the contribution of each major component in the proposed pipeline, we perform a series of controlled ablation experiments. Starting from the full model, individual modules are removed while keeping the remaining architecture, training procedure, and hyperparameters unchanged. We report validation macro-F1 for all ablation experiments. The evaluated components include Easy Data Augmentation (EDA), probability chaining, and the choice of loss function (Focal Loss versus standard Cross Entropy). The full model consists of DeBERTa-v3-base with Global Attention Pooling, Focal Loss, WeightedRandomSampler, EDA-based augmentation, and probability chaining.

Since all experiments use identical random seeds (seed $= 42$), data splits (stratified on Task 1 labels), model architecture, optimization settings, and checkpoint selection criteria, the observed performance differences can be attributed directly to the ablated component. We note that the macro-F1 values reported in this section are obtained from a controlled validation run conducted specifically for the ablation study using the earlier EDA-based augmentation configuration, and therefore differ from the final submission run reported in Section 6.

Table 4 summarizes the validation macro-F1 obtained by each configuration. The ablation experiments were conducted on an earlier controlled development configuration of the pipeline prior to the final Task-3 semantic-enrichment and WordNet-based augmentation refinements described in Section 3.6. The purpose of these experiments is to isolate the relative contribution of individual components under identical training conditions rather than to reproduce the final submission scores.

### 5.1.1. Effect of Easy Data Augmentation

Removing EDA causes the average macro-F1 to decrease from 0.7733 to 0.7094, corresponding to a drop of approximately 6.4 percentage points. The largest degradation occurs on Task 3, where performance falls from 0.6677 to 0.5218. This result is expected because Task 3 contains the most severe class imbalance and the smallest effective training set. The augmentation configuration used during the ablation experiments improves minority-class representation and particularly benefits Task 3, where

**Table 4**
Ablation study on validation macro-F1. All configurations are evaluated under identical settings except for the ablated component.

| Configuration | Task 1 | Task 2 | Task 3 | Average |
|---|---|---|---|---|
| Full Model | 0.7934 | 0.8588 | 0.6677 | 0.7733 |
| Without EDA | 0.7822 | 0.8244 | 0.5218 | 0.7094 |
| Without Probability Chaining | 0.8000 | 0.8504 | 0.7043 | 0.7849 |
| Cross Entropy (replacing Focal Loss) | 0.7777 | 0.8369 | 0.8356 | 0.8167 |

severe class imbalance is present, by increasing lexical diversity and exposing the model to a wider range of linguistic variations. Without augmentation, the classifier tends to overfit dominant classes and struggles to generalize to underrepresented demographic categories. The relatively smaller reductions observed on Tasks 1 and 2 suggest that these tasks already possess sufficient training data and are less dependent on augmented lexical diversity.

#### 5.1.2. Effect of Probability Chaining

Removing probability chaining increases the average macro-F1 from 0.7733 to 0.7849, with the most noticeable improvement occurring on Task 3 (0.6677 to 0.7043). Although probability chaining was designed to exploit hierarchical dependencies, this result suggests that the propagated probability features may introduce noise rather than useful information. Because Task 1 and Task 2 predictions are imperfect, downstream classifiers receive uncertain or incorrect confidence signals. These noisy features can amplify upstream errors and encourage over-reliance on probability values instead of the textual representations extracted by DeBERTa. A second contributing factor is probability calibration: if the upstream confidence scores are not well calibrated, they may not accurately represent prediction uncertainty, leading the downstream classifier to learn misleading correlations that reduce generalization. The results indicate that the contextual representation learned by DeBERTa already captures much of the information needed for downstream prediction, limiting the practical benefit of explicit probability propagation.

#### 5.1.3. Effect of the Loss Function

Replacing Focal Loss with standard Cross Entropy improves the average macro-F1 from 0.7733 to 0.8167, with a particularly pronounced gain on Task 3, where macro-F1 increases from 0.6677 to 0.8356. While Focal Loss is generally beneficial for imbalanced datasets, this result suggests that the combination of WeightedRandomSampler and EDA augmentation may already provide sufficient class balancing. Under these conditions, Focal Loss may overemphasize difficult or noisy examples, devoting excessive model capacity to outliers and borderline cases. Cross Entropy, in contrast, optimizes all samples uniformly and may produce more stable and better-calibrated decision boundaries once class imbalance has already been partially corrected through sampling and augmentation. The large gain observed on Task 3 further suggests that, in this specific setting, Focal Loss may suppress useful gradient information for minority categories whose representation has already been strengthened upstream.

#### 5.1.4. Discussion of Ablation Results

The ablation study reveals that data-centric techniques contribute more strongly to performance than hierarchical feature propagation. EDA demonstrates a clear and consistent benefit, particularly for the highly imbalanced Task 3 classification problem. Conversely, probability chaining does not improve performance in the current controlled setting and may instead introduce error propagation from upstream tasks. Most notably, standard Cross Entropy outperforms Focal Loss despite the presence of class imbalance, indicating that the balancing mechanisms already present in the pipeline (weighted

sampling and EDA) may sufficiently address minority-class scarcity and reduce the need for aggressive loss reweighting. Overall, these results indicate that robust data balancing and augmentation strategies contribute more to performance than explicit hierarchical probability propagation, while also highlighting the sensitivity of hierarchical classification systems to error propagation across task boundaries.

## 6. Results and Analysis

Table 5 presents the evaluation results on the validation set. Task 1 achieves a macro-F1 score of 0.7937, indicating strong performance on the supportive intent detection task despite class imbalance. Task 2 obtains a macro-F1 of 0.8384, reflecting moderate performance on the conditional classification of individual versus group targets. Task 3 achieves a macro-F1 of 0.6806, highlighting the increased difficulty of fine-grained multi-class classification under hierarchical constraints. The overall pipeline average macro-F1 is 0.7709.

**Table 5**
Comparison of macro-F1 scores per subtask between the proposed sequential DeBERTa pipeline and baseline models. Higher is better.

| Model | Subtask 1 | Subtask 2 | Subtask 3 |
|---|---|---|---|
| BERT-base-uncased (untrained) | 0.3091 | 0.1459 | 0.0075 |
| RoBERTa-base (untrained) | 0.4353 | 0.4553 | 0.1479 |
| Qwen2.5-1.5B-Instruct (zero-shot) | 0.6031 | 0.6208 | 0.4894 |
| **Proposed (model_a/b/c)** | **0.7937** | **0.8384** | **0.6806** |

**Baseline Comparison.** Table 5 compares our proposed pipeline against three baselines: an untrained (randomly initialized classification head on top of pre-trained) BERT-base-uncased [1], an untrained RoBERTa-base [15], and a zero-shot Qwen2.5-1.5B-Instruct [17] prompted with the task definition and label set. The untrained transformer baselines perform poorly, particularly on Subtask 3, where the BERT-base baseline barely exceeds chance level ($\approx 0.01$), confirming that fine-tuning with class-imbalance corrections is essential for this dataset. The zero-shot Qwen2.5-1.5B-Instruct baseline performs noticeably better than untrained encoders on all three subtasks (0.6031 / 0.6208 / 0.4894), indicating that instruction-tuned LLMs capture meaningful supportive-intent signals even without task-specific supervision. However, our task-specialised DeBERTa-v3 pipeline with Global Attention Pooling, Focal Loss, weighted sampling, EDA and probability chaining outperforms the strongest baseline by absolute margins of $+0.191$, $+0.218$, and $+0.191$ macro-F1 on Subtasks 1, 2, and 3 respectively. The pipeline yields substantial and consistent gains across all three subtasks, indicating that the semantic-enrichment and WordNet-based augmentation of the underrepresented target groups, together with the supervised fine-tuning of the task-specific classifiers, is beneficial even for the most challenging fine-grained target-group classification. Throughout this paper, the labels *model_a*, *model_b*, and *model_c* refer to the three task-specific DeBERTa-v3 classifiers in our sequential pipeline, corresponding to the Subtask 1 (supportive intent), Subtask 2 (individual vs. group), and Subtask 3 (target group) models, respectively.

Task 1 achieves the highest performance among the three tasks, demonstrating that the model effectively captures supportive intent even in noisy social media text. However, the presence of implicit sentiment and ambiguous phrasing still introduces classification challenges. Task 2 exhibits a slight drop in performance, which can be attributed to its dependence on correct upstream predictions as well as the inherent ambiguity between individual and group references.

Task 3 is the most challenging component of the pipeline, as it involves six-class classification under compounded hierarchical dependencies. Errors from both Task 1 and Task 2 propagate to this stage, reducing overall accuracy. Additionally, severe class imbalance across demographic groups further impacts performance. Despite the use of probability chaining and targeted augmentation strategies,

distinguishing between fine-grained categories such as Religion, LGBTQ, and Black Community remains difficult due to overlapping linguistic patterns and limited training samples. Overall, the results highlight the effectiveness of the proposed sequential framework for higher-level tasks while also revealing the limitations of hierarchical pipelines in handling fine-grained classification under noisy and imbalanced conditions.

## 6.1. Validation Results: Per-Class Analysis

To complement the aggregate macro-F1 scores and provide a clearer picture of behavior under class imbalance, Table 6 reports accuracy together with macro precision, recall, and F1 for each subtask on the validation set, and Table 7 reports per-class precision, recall, and F1 for the fine-grained Task 3 target-group classification. We emphasize that all scores reported in this paper are computed on our held-out validation split and are *not* official shared-task test/leaderboard results; the official SSD-2026 test-set scores obtained by our system are reported separately at the end of this section.

**Table 6**
Validation-set performance by subtask (DeBERTa-v3-base pipeline), reporting accuracy and macro-averaged precision, recall, and F1.

| Task | Accuracy | Macro P | Macro R | Macro F1 |
|---|---|---|---|---|
| Task 1 – Support Detection | 86.07% | 0.8094 | 0.7823 | 0.7937 |
| Task 2 – Individual vs. Group | 91.42% | 0.8769 | 0.8094 | 0.8384 |
| Task 3 – Target Group[†] | 76.34% | 0.7009 | 0.6697 | 0.6806 |
| **Overall (average)** | **84.61%** | **0.7957** | **0.7538** | **0.7709** |

[†]Task 3 metrics are computed using `sklearn.metrics.f1_score` with macro averaging over the union of labels present in the ground-truth and predicted outputs. During validation, the Religion class has no ground-truth instances (support $= 0$), but the model predicts this class for two samples. Consequently, `scikit-learn` includes the Religion class in the macro-average computation, yielding the reported Task 3 macro-F1 of 0.6806. Tables 7 and 8 present the corresponding per-class statistics and confusion matrix.

**Table 7**
Per-class precision, recall, and F1 for Task 3 (target-group identification) on the validation set. Support denotes the number of validation instances per class.

| Class | Precision | Recall | F1 | Support |
|---|---|---|---|---|
| Nation | 0.80 | 0.85 | 0.83 | 116 |
| Other | 0.62 | 0.58 | 0.60 | 66 |
| LGBTQ | 1.00 | 0.71 | 0.83 | 21 |
| Black Community | 0.78 | 0.88 | 0.82 | 16 |
| Religion | 0.00 | 0.00 | 0.00 | 0 |
| Women | 1.00 | 1.00 | 1.00 | 5 |

**Official SSD-2026 test-set results.** On the official SSD-2026 shared-task test set, our system obtained macro-F1 scores of $0.8027$ (Task 1), $0.7645$ (Task 2), and $0.4765$ (Task 3), corresponding to leaderboard ranks of 17, 18, and 17 respectively. While the Task 1 test score is comparable to its validation counterpart ($0.7937$), the Task 2 and, most notably, Task 3 test scores fall below their validation values ($0.8384$ and $0.6806$); this drop, largest for the fine-grained Task 3, likely reflects a degree of overfitting to the validation split during model and threshold selection.

Table 8 presents the corresponding Task 3 confusion matrix. The dominant confusions occur between the *Nation* and *Other* categories, which account for most of the residual error, while the minority *Women*, *Black Community*, and *LGBTQ* classes are recovered with high precision once augmentation

has strengthened their representation. The *Religion* class has no validation instances but nonetheless receives two predictions, and therefore does not contribute reliable per-class statistics.

**Table 8**
Task 3 confusion matrix on the validation set (rows = true class, columns = predicted class).

| True \ Pred | Nation | Other | LGBTQ | Black | Religion | Women |
|---|---|---|---|---|---|---|
| Nation | 99 | 15 | 0 | 0 | 2 | 0 |
| Other | 24 | 38 | 0 | 4 | 0 | 0 |
| LGBTQ | 0 | 6 | 15 | 0 | 0 | 0 |
| Black | 0 | 2 | 0 | 14 | 0 | 0 |
| Religion | 0 | 0 | 0 | 0 | 0 | 0 |
| Women | 0 | 0 | 0 | 0 | 0 | 5 |

## 7. Error Analysis

We perform a qualitative analysis of validation set errors to identify systematic failure modes across all three tasks. For Task 1 (Supportive detection), the predominant error type is false negatives on the Supportive class. Texts that express support in an implicit or context-dependent manner (e.g., "These people deserve to be heard" without explicit group reference) are frequently misclassified as Non-Supportive. A secondary source of error arises from false positives, where generally positive or empathetic language is incorrectly interpreted as supportive intent toward a specific group or individual.

For Task 2 (Individual vs. Group), the primary source of error occurs in sentences that simultaneously reference both an individual and a broader group identity (e.g., "Support [Name] and all the women who stand with her"). In such cases, the model tends to favor the Individual label, likely due to the strong signal from named entities. While probability chaining partially mitigates this issue by incorporating upstream contextual confidence, ambiguity in individual–group co-reference remains a challenging aspect of the task.

For Task 3 (Target Group classification), errors are concentrated among semantically similar classes. The Religion and Nation categories exhibit significant confusion, as expressions of support for religious communities often overlap with national identity references. Similarly, confusion between LGBTQ and Women categories is observed, reflecting overlapping linguistic patterns in discourse. The semantic-enrichment and WordNet-based augmentation improves representation for heavily augmented classes such as Women, while its impact on Religion cannot be reliably evaluated because the validation split contains no Religion instances; however, it has limited impact on resolving ambiguity between closely related categories such as LGBTQ and Women, suggesting the need for more targeted augmentation strategies.

Additionally, error propagation from upstream tasks contributes significantly to downstream misclassifications. Error propagation from earlier subtasks contributes to a noticeable fraction of the remaining Task 3 errors, reflecting the inherent limitation of sequential pipelines. This observation suggests that future work could explore joint or end-to-end training approaches to reduce compounded error effects.

## 8. Discussion

The empirical results provide mixed evidence regarding the effectiveness of individual design choices in the proposed pipeline. Although the proposed pipeline incorporates explicit modeling of hierarchical dependencies, the controlled ablation study indicates that probability chaining does not consistently improve validation macro-F1 and may introduce noise through imperfect upstream predictions. In contrast, the semantic-enrichment and WordNet-based augmentation strategy for underrepresented Task 3 groups was incorporated to strengthen minority-class representation for fine-grained target-

group classification, reflecting the value of lightweight, label-aware augmentation that preserves semantics over generic perturbation strategies.

Despite these improvements, the relatively lower performance on Task 3 highlights the inherent difficulty of fine-grained classification under hierarchical constraints. The dependence on upstream predictions introduces compounded error effects, and the six-class structure further increases classification complexity. While the combined representation $[h_{pool}; p_1; p_2]$ provides useful contextual information, it is not always sufficient to fully resolve ambiguity between closely related demographic categories.

From a practical perspective, the pipeline introduces moderate computational overhead. Each of the three DeBERTa models is trained independently and stored, allowing incremental experimentation and recovery without retraining the entire system. The semantic-enrichment and WordNet-based augmentation for Task 3 is lightweight and fully local, requiring no external API access, and adds negligible preprocessing overhead.

An additional advantage of the sequential design is improved interpretability. Intermediate predictions $p_1$ and $p_2$ are directly observable and can be logged for analysis, enabling practitioners to trace prediction errors across the hierarchy. This facilitates debugging and provides insights into how errors propagate through the system, which is more difficult in fully joint or black-box multi-task models.

## 9. Limitations

Several limitations of the current work warrant acknowledgment. First, the sequential pipeline propagates upstream decisions during inference. Although probability chaining introduces soft signals, the Task 3 training subset is still defined by hard upstream labels ($Y_1 = 1, Y_2 = 1$), meaning that Task 3 is not exposed to upstream prediction errors during training. This limits its ability to recover from misclassifications in earlier stages.

Second, the semantic-enrichment and WordNet-based augmentation used for Task 3 is restricted to English, since WordNet synonym resources are English-centric. Extending this approach to other languages, such as Spanish, would require multilingual lexical resources, which are not explored in this work.

Finally, while the Global Attention Pooling mechanism provides token-level weighting, the interpretability of the underlying DeBERTa model remains limited. The attention weights offer only a coarse indication of token importance and do not constitute a complete or rigorous explanation of model decisions.

A further limitation concerns the use of LLM-generated synthetic data for minority classes. Although every synthetic sample is filtered through an automated label-consistency and grammar verification stage, this process does not fully eliminate risk. Residual label noise may persist when the verification model and the generation model share the same biases, and synthetic posts may exhibit distributional drift relative to genuine social-media comments—for example, differing in register, slang, code-mixing, or topical coverage. Because only 80 synthetic samples are added across four classes, this risk is bounded in scope, but over-reliance on synthetic augmentation could in principle cause the classifier to fit artifacts of the generator rather than authentic minority-class signal. We therefore treat synthetic augmentation as a complement to, rather than a replacement for, real annotated data, and we regard further validation of the synthetic set against genuine social-media data as an important direction for ensuring its quality.

## 10. Conclusion and Future Work

This paper presents a sequential transformer-based framework for hierarchical text classification built on DeBERTa-v3-base with Global Attention Pooling. The proposed system addresses key challenges in the task, including hierarchical label dependency, class imbalance, and noisy social media text, through a combination of probability chaining, Focal Loss, weighted sampling, EDA-based augmentation, and a semantic-enrichment and WordNet-based lexical augmentation strategy for the most underrepresented target groups. Experimental results show that the approach achieves validation macro-F1 scores of

0.7937, 0.8384, and 0.6806 on Tasks 1, 2, and 3 respectively, highlighting competitive performance on higher-level tasks and the increased difficulty of fine-grained classification under hierarchical constraints. The analysis further reveals the impact of error propagation and class overlap on downstream performance. Overall, our experiments suggest that targeted data augmentation contributes more strongly to performance than explicit modeling of inter-task dependencies, whose benefit appears limited in the current setting. The insights from this work provide a foundation for future improvements, including joint multi-task learning and more robust data generation techniques.

Several promising directions emerge for future research. First, joint multi-task learning with hierarchical loss weighting could enable end-to-end training across all tasks, allowing Task 3 to benefit from gradients propagated through upstream layers rather than relying on filtered subsets. Additionally, graph-based modeling of label dependencies, where the hierarchy is represented as a directed acyclic graph, could provide a more principled alternative to scalar probability chaining.

Second, extending the framework to a multilingual setting using models such as mDeBERTa-v3 could improve performance on low-resource language variants by leveraging shared representations across languages. Third, uncertainty-aware methods such as conformal prediction could be incorporated to provide calibrated confidence estimates for each task, enabling safer deployment in real-world applications. Finally, extending the lexical augmentation strategy with richer, contextualized synonym resources and more demographically varied training examples, may help reduce confusion between closely related classes, particularly those identified in the error analysis.

## Acknowledgments

We thank the organisers of the Social Support Detection (SSD-2026) shared task at IberLEF 2026 for providing the dataset and the evaluation framework that made this work possible.

## Declaration on Generative AI

During the preparation of this work, the authors used generative AI in two distinct capacities. First, as an explicit and documented component of the methodology, instruction-tuned large language models (OpenAI GPT-4o-mini or Groq Llama-3-8B-8192) were used to generate and verify synthetic training samples for the most underrepresented Task-3 classes, as described in Section 3.7; all generated samples passed an automated label-consistency and grammar verification stage, and the resulting data were used only to augment the training set. Second, for manuscript preparation, generative AI tools were used in a limited capacity for grammar and spelling correction and for paraphrasing and language refinement of author-written draft text. After using these tools, the authors reviewed and edited all content as needed and take full responsibility for the publication's content.

## References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[2] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," in *Proc. ICLR*, 2021.

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. ICCV*, 2017, pp. 2980–2988.

[4] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 6382–6388.

[5] C. N. Silla Jr. and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, pp. 31–72, 2011.

[6] J. Zhou, C. Ma, D. Long, G. Xu, N. Ding, H. Zhang, P. Xie, and G. Liu, "Hierarchy-aware global model for hierarchical text classification," in *Proc. ACL*, 2020, pp. 1106–1117.

[7] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-Instruct: Aligning language models with self-generated instructions," in *Proc. ACL*, 2023.

[8] M. Shahiki Tash, L. Ramos, A. Zahra, A. Y. Barrera Animas, F. Balouchzahi, O. Kolesnikova, G. Sidorov, A. Gelbukh, R. Monroy, and 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.

[9] A. Bonet-Jover, J. Á. González-Barba, and 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.

[10] Z. Ahani, M. S. Tash, F. Balouchzahi, L. Ramos, G. Sidorov, and A. Gelbukh, "Social support detection from social media texts," *PLOS ONE*, 2026. DOI: https://doi.org/10.1371/journal.pone.0337476.

[11] O. Kolesnikova, M. S. Tash, Z. Ahani, A. Agrawal, R. Monroy, and G. Sidorov, "Advanced machine learning techniques for social support detection on social media," *Heliyon*, vol. 11, no. 10, 2025. DOI: https://doi.org/10.1016/j.heliyon.2025.e43437.

[12] M. S. Tash, L. Ramos, Z. Ahani, R. Monroy, O. Kolesnikova, H. Calvo, and G. Sidorov, "Online social support detection in Spanish social media texts," *arXiv preprint arXiv:2502.09640*, 2025.

[13] M. Shahiki-Tash, Z. Ahani, L. Ramos, et al., "Linguistic analysis in different types of support using LIWC for Spanish and English online communities," 2025. DOI: https://doi.org/10.21203/rs.3.rs-9557287/v1.

[14] W. W. Chi, T. Y. Tang, N. M. Salleh, M. Mukred, H. AlSalman, and M. Zohaib, "Data Augmentation With Semantic Enrichment for Deep Learning Invoice Text Classification," *IEEE Access*, 2024.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[16] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. ICLR*, 2020.

[17] Qwen Team, "Qwen2.5 Technical Report," *arXiv preprint arXiv:2412.15115*, 2025.

[18] OpenAI, "GPT-4o System Card," *arXiv preprint arXiv:2410.21276*, 2024.

[19] A. Grattafiori et al. (Llama Team, AI@Meta), "The Llama 3 Herd of Models," *arXiv preprint arXiv:2407.21783*, 2024.