SRH-ReliableAI at SimpleText 2026 Task 3: A Hierarchy-Constrained, Adversarially Trained Multi-Task Ensemble for Scientific Research Area Classification

Notebook for the SimpleText Lab at CLEF 2026

Pratik Priyanshu[1,*], Swati Chandna[1]

[1]SRH University Heidelberg, Heidelberg, Germany

Abstract

We describe the SRH-ReliableAI submission to the SimpleText 2026 Task 3 shared task on research area classification. Given a scientific paper's title and abstract, the task requires predicting both a coarse *field area* (5 classes) and a fine-grained *discipline area* (23 classes) from a two-level DFG-style taxonomy; the official evaluation reports Exact Match per gold field area together with the macro-average over the five field areas. We propose a hierarchy-constrained multi-task ensemble that combines a general-purpose encoder (DeBERTa-v3-large) and a domain-specific encoder (SciBERT). Each encoder is fine-tuned with stratified 5-fold cross-validation under a shared multi-task head, with Fast Gradient Method adversarial perturbations on the input embeddings and multi-sample dropout for regularization. At inference, predictions from all ten fold models are averaged at the probability level, and discipline decoding is constrained to the predicted field's subtree, guaranteeing taxonomic consistency by construction. A confidence-thresholded pseudo-labeling stage augments the training set with high-confidence test predictions and contributes one additional DeBERTa model to the ensemble. On the official test set our two submitted runs obtain identical scores: Exact Match of 0.92, 0.81, 0.86, and 0.82 on the Natural Sciences, Life Sciences, Engineering Sciences, and Social Sciences field areas respectively (tied-best on Natural Sciences and within 0.02 of the best submission on the other three), but 0.00 on Humanities, for a macro-average of 0.68 (fourth of seven participating teams). A prediction-level analysis explains why: the ensemble reproduces the training marginals almost exactly and never predicts any of the ten discipline labels with at most 15 training examples, so the Humanities field area (0.2% of the training data, yet 20% of the macro-averaged metric) is never emitted at all. We use this outcome to examine how, under macro-averaged evaluation with a severely long-tailed label distribution, the treatment of the rarest classes rather than encoder capacity determines the final ranking.

Keywords

scientific text classification, hierarchical classification, multi-task learning, adversarial training, model ensembling, DeBERTa, SciBERT, SimpleText

1. Introduction

Automatic classification of scientific literature into a research-area taxonomy supports a wide range of downstream tasks, including expertise retrieval, bibliometric analysis, recommender systems, and the upstream processing of pipelines for scientific text simplification. The SimpleText track at CLEF situates this problem at the entry point of its broader simplification effort: identifying a paper's research area is a prerequisite for adapting simplification, summarization, and faithfulness verification components to the conventions and terminology of the underlying scientific domain [1]. Task 3 of SimpleText 2026 formalizes this as a two-level classification problem [2]: given the title and abstract of a paper, predict its broad *field area* (5 classes) and its specific *discipline area* (23 classes), where each discipline is the child of exactly one field.

The task exhibits three properties that distinguish it from standard text classification benchmarks and that drive our system design.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

pratikpriyanshu12345@gmail.com (P. Priyanshu)

0009-0006-6192-3767 (P. Priyanshu); 0009-0003-8393-5337 (S. Chandna)

Macro-averaged evaluation. The official evaluation reports Exact Match separately for each gold field area and ranks systems by the macro-average over the five field areas (a string-similarity variant is also published). Macro-averaging weights all five field areas equally regardless of their frequency: *Humanities*, which accounts for 0.2% of the training data, carries exactly the same weight in the final ranking as *Natural Sciences*, which accounts for 76.5%.

Hierarchical structure. Discipline labels are nested under field labels by a strict parent function $\pi : D \to F$. Unconstrained two-head classifiers may emit field–discipline pairs that violate this hierarchy, a structurally invalid output that any taxonomy-aware downstream component would reject. Enforcing hierarchical consistency is therefore both a quality requirement and a source of additional supervision.

Severe class imbalance. The training distribution is long-tailed at both levels. *Natural Sciences* accounts for 76.5% of training examples; *Humanities* for only 0.2% (23 examples). At the discipline level, two classes (*Physics*, *Mathematics*) cover approximately 73% of training data, while five classes have between 2 and 12 examples. Aggregate accuracy is therefore dominated by a small number of head classes, motivating explicit treatments for the long tail.

We address these properties through a single end-to-end pipeline with five components:

1. A **multi-task classification head** that jointly predicts field and discipline from a shared encoder representation, encouraging features that respect the taxonomic structure.
2. A **two-encoder ensemble** pairing DeBERTa-v3-large [3] for general linguistic competence with SciBERT [4] for in-domain scientific vocabulary.
3. **Fast Gradient Method (FGM) adversarial training** [5] on input embeddings, applied at every training step.
4. **Hierarchy-constrained inference** that restricts discipline predictions to the predicted field's subtree, guaranteeing taxonomic consistency.
5. A **confidence-thresholded pseudo-labeling stage** that augments training with high-confidence test predictions.

The contributions of this paper are threefold. First, we present a complete system for hierarchical scientific text classification that combines the above components and is reproducible on a single GPU. Second, we report the official evaluation results, on which our runs are tied-best or within 0.02 of the best submission on four of the five field areas while placing fourth of seven teams on the macro-average. Third, we present a prediction-level analysis of this gap, showing that the ensemble reproduces the training marginals almost exactly but never emits any label from the long tail of the distribution, and quantifying how a single rare field area, evaluated on roughly ten test instances, determines the leaderboard ordering.

2. Related Work

Hierarchical text classification. Two broad strategies have been developed for classifying text into a hierarchical taxonomy: *local* approaches that train an independent classifier per node or per level, and *global* approaches that predict the complete label path jointly [6]. Subsequent work has explored hierarchy-aware loss functions [7] and label embeddings that respect the taxonomy graph [8]. Our approach is a global method with a soft hierarchy signal during training (a multi-task objective over both levels) and an explicit hierarchy constraint at inference. This design avoids modifying the training loss surface while still guaranteeing taxonomically consistent outputs.

Transformer encoders for scientific text. BERT [9] established pre-trained transformer encoders as the default for text classification. SciBERT [4] retrains the same architecture on a scientific corpus,

producing in-domain subword vocabulary and improved performance on biomedical and computer-science benchmarks. DeBERTa [10] introduces disentangled attention and an enhanced mask decoder, with DeBERTaV3 [3] adding gradient-disentangled embedding sharing. Our ensemble combines a general-purpose DeBERTaV3 encoder with the in-domain SciBERT encoder to draw on both broad linguistic competence and scientific terminology coverage.

Multi-task and multi-head classification. Multi-task learning [11] encourages shared representations to generalize across related tasks. In the present setting, field and discipline prediction are not independent (the field is a deterministic function of the discipline), and sharing the encoder while splitting only the classification head exploits this structural relationship.

Adversarial training in NLP. Adversarial training, originally developed for image classification [12], has been adapted to NLP via input-embedding perturbations. Miyato et al. [5] introduced the Fast Gradient Method (FGM) for text classification, and PGD [13] generalizes it to multi-step perturbations. We adopt FGM as a single-step, computationally efficient regularizer.

Ensembling and pseudo-labeling. Cross-validation ensembling [14] averages predictions across folds to reduce variance, and combining heterogeneous encoder families is a standard strategy for further error decorrelation. Pseudo-labeling [15] reuses confidently predicted test instances as additional training data and has been shown to be most effective when combined with confidence thresholding [16].

3. Task and Data

3.1. Task Definition

Given the title and abstract of a scientific paper, predict (a) its field area $f \in F$ and (b) its discipline area $d \in D$, where each discipline has exactly one parent field via the relation $\pi : D \to F$. The official evaluation partitions the test set by gold field area and reports, for each of the five field areas, the Exact Match (EM) rate of the submitted labels on that partition, together with the macro-average over the five field areas, which is the ranking metric. A string-similarity variant of the same table is also published; on the 2026 leaderboard its values coincide with the EM scores at the published precision, so we report EM throughout.

3.2. Taxonomy

$|F| = 5$ field areas (*Engineering Sciences*, *Humanities*, *Life Sciences*, *Natural Sciences*, *Social Sciences*) and $|D| = 23$ discipline areas. The five fields contain 7, 5, 3, 4, and 4 disciplines respectively. We pre-compute the inverse map $\pi^{-1} : F \to 2^D$ once at the start of training and use it during inference (Section 4.5).

3.3. Data

Training: 12,301 labeled papers. Public test: 3,106 unlabeled papers. Each instance gives the title and abstract; train rows also include the gold field and discipline. Title+abstract length (in characters) is mean 1,048, median 1,014, 95th percentile 1,839, max 3,792, so a 384-token window comfortably covers the great majority of inputs.

3.4. Class Imbalance

Field distribution: *Natural Sciences* 76.5%, *Engineering Sciences* 13.0%, *Life Sciences* 7.4%, *Social Sciences* 2.9%, *Humanities* 0.2% (only 23 examples). Discipline distribution: *Physics* (4,995) and *Mathematics* (4,001) together cover roughly 73% of training, while the rarest five disciplines (Civil Engineering,

Materials Science, Archeology, Literary Studies, Process Engineering) have between 2 and 12 training examples each. Stratified splitting is therefore essential, and aggregate accuracy figures are difficult to interpret without per-class results alongside them.

4. System Description

4.1. Pipeline Overview

For each encoder $E \in \{\text{DeBERTa-v3-large}, \text{SciBERT}\}$, we train one multi-task classifier per fold of a stratified 5-fold split, yielding 10 fold models in total. At test time, every model produces a field probability vector $p_f^{(m)}$ and a discipline probability vector $p_d^{(m)}$. We average those across all models to get $\bar{p}_f$ and $\bar{p}_d$, then run hierarchy-constrained decoding to pick the final (f, d) pair. A pseudo-labeling round then trains one additional DeBERTa model on the original training data plus high-confidence test predictions, and we re-average.

4.2. Input Representation

Each instance is encoded as "`<title> [SEP] <abstract>`" truncated to 384 tokens. Placing the title first ensures it is never lost to truncation. The `[SEP]` marker indicates the boundary between the two segments without requiring a separate segment ID.

4.3. Multi-Task Classifier

Let $h \in \mathbb{R}^H$ be the encoder's CLS representation. Two linear heads $W_f \in \mathbb{R}^{|F| \times H}$ and $W_d \in \mathbb{R}^{|D| \times H}$ produce field and discipline logits respectively. We use multi-sample dropout [17] with $K = 3$ independent dropout masks: for each head, we apply each mask, project, and average the resulting logits. The additional cost is negligible, and the effect resembles ensembling K slightly different sub-networks within a single forward pass.

The training objective is a weighted sum of two cross-entropy losses,

$$\mathcal{L} = \lambda_f \cdot \mathrm{CE}(z_f, y_f) + \lambda_d \cdot \mathrm{CE}(z_d, y_d), \tag{1}$$

with $\lambda_f = 0.3$ and $\lambda_d = 0.7$. The higher weight on the discipline term reflects that discipline classification is the harder of the two sub-tasks (23 classes vs. 5) and that discipline correctness implies field correctness whenever hierarchy-constrained inference is applied (Section 4.5). The field term is retained with a non-negligible weight to regularize the shared encoder representation toward features that remain discriminative at the coarser level.

4.4. FGM Adversarial Training

After each standard forward–backward pass, we apply FGM [5] perturbations to the input-embedding weights θ_e:

$$\delta = \epsilon \cdot \frac{\nabla_{\theta_e} \mathcal{L}}{\|\nabla_{\theta_e} \mathcal{L}\|_2}, \tag{2}$$

We temporarily set $\theta_e \leftarrow \theta_e + \delta$, redo the forward–backward pass with the same batch, accumulate gradients, and then restore the original embeddings before the optimizer step. We use $\epsilon = 1.0$. The cost is roughly one extra forward–backward pass per step, doubling step time but adding no parameters.

4.5. Hierarchy-Constrained Inference

Given ensemble-averaged probabilities $\bar{p}_f$ and $\bar{p}_d$, naive arg-max decoding can produce a pair where the predicted discipline does not live under the predicted field, which is a structurally illegal output. We decode in two steps instead:

Table 1
Training hyperparameters for the two encoders.

Setting	DeBERTa-v3-large	SciBERT
Learning rate	2×10^{-5}	3×10^{-5}
Batch size	8	32
Gradient accumulation	2 (eff. 16)	1
Epochs	5	5
Max sequence length	384	384
Optimizer	AdamW	AdamW
Weight decay	0.01	0.01
LR schedule	Cosine, 10% warmup	Cosine, 10% warmup
Gradient clip	1.0	1.0
Multi-sample dropout	K=3	K=3
FGM ϵ	1.0	1.0
Mixed precision (AMP)	Yes	Yes
Gradient checkpointing	Yes	No
λ_f / λ_d	0.3 / 0.7	0.3 / 0.7

1. $\hat{f} = \arg\max_f \bar{p}_f$.
2. $\hat{d} = \arg\max_{d \in \pi^{-1}(\hat{f})} \bar{p}_d[d]$, masking all off-branch disciplines to $-\infty$ before the arg-max.

This guarantees $\pi(\hat{d}) = \hat{f}$ by construction. The constraint costs nothing at training time: the model is free to make hierarchy-inconsistent predictions during training (and often does), and consistency is enforced only at the final decoding step.

4.6. Ensembling

We average *probabilities*, not logits, across all fold models with equal weight. Probability averaging is more robust than logit averaging when models have differently calibrated temperatures, which we observed informally between SciBERT and DeBERTa folds.

4.7. Confidence-Thresholded Pseudo-Labeling

After the initial ensemble produces test predictions, we compute a joint confidence score $c_i = (\max_f \bar{p}_f^{(i)}) \cdot (\max_d \bar{p}_d^{(i)})$ for each test instance i and take all instances with $c_i \geq 0.90$. These are added to the training set with their predicted labels. We then train one fresh DeBERTa-v3-large model on the augmented data (a single fold with a 5% held-out validation slice, 3 epochs, learning rate 1×10^{-5}), obtain its test-set predictions, and append them to the ensemble with double weight before re-averaging. The intuition is that the test distribution may not perfectly match training, and these confident test examples carry information about the test marginal that the original training set lacks.

5. Experimental Setup

5.1. Data Splits

We use stratified 5-fold cross-validation on the discipline label, so every fold preserves the discipline distribution of the whole training set. Field-stratified folds gave near-identical splits in informal tests; we use discipline-stratified throughout.

Table 2
Official SimpleText 2026 Task 3 results: Exact Match per gold field area and macro-average (the ranking metric). Our runs in bold. The published string-similarity scores coincide with the EM scores at this precision and are omitted.

Run	Nat.	Life	Eng.	Soc.	Hum.	Macro
Data Dumplings	0.91	0.75	0.79	0.84	0.60	0.78
Stuttgart AI	0.92	0.78	0.85	0.76	0.30	0.72
UBCS_3	0.92	0.76	0.86	0.80	0.20	0.71
UBCS_2	0.92	0.75	0.82	0.79	0.20	0.69
UBCS_1	0.92	0.75	0.83	0.73	0.20	0.69
SRH Reliable AI_2 (ours)	**0.92**	**0.81**	**0.86**	**0.82**	**0.00**	**0.68**
SRH-Reliable AI_1 (ours)	**0.92**	**0.81**	**0.86**	**0.82**	**0.00**	**0.68**
AIIRLab_2	0.91	0.82	0.87	0.75	0.00	0.67
AIIRLab_1	0.92	0.80	0.78	0.82	0.00	0.66
UAmsterdam_1	0.81	0.54	0.86	0.48	0.40	0.62
Writerslogic Inc	0.79	0.32	0.11	0.62	0.00	0.37
UAmsterdam_2	0.54	0.00	0.00	0.00	0.00	0.11

5.2. Hyperparameters

5.3. Compute

All experiments ran on a single NVIDIA H200 NVL GPU on the SRH University DataLab. Gradient checkpointing keeps DeBERTa training comfortably within VRAM. Wall-clock time for the full pipeline (10 fold models + the pseudo-labeling round) is dominated by the five DeBERTa-v3-large folds, whose per-step cost is doubled by the FGM pass; the complete pipeline trains within a single day on this GPU.

5.4. Software

The implementation uses PyTorch 2.x with HuggingFace Transformers. The complete pipeline is contained in a single Jupyter notebook released alongside this paper (see the Reproducibility paragraph in the Conclusion).

6. Results

6.1. Official Test Results

Table 2 reports the complete official leaderboard: per-field-area Exact Match and the macro-average ranking metric for all twelve submitted runs from seven teams. Both of our submitted runs obtain identical published scores.

Two observations frame the rest of this section. First, on four of the five field areas our system is at or near the leaderboard ceiling: tied-best on Natural Sciences (0.92), and within 0.01–0.02 of the best submission on Life Sciences (0.81 vs. 0.82), Engineering Sciences (0.86 vs. 0.87), and Social Sciences (0.82 vs. 0.84). No other run is in the top two on all four of these field areas. Second, on Humanities the system scores exactly 0.00, and this single cell reduces the macro-average to 0.68, fourth of the seven teams. The next two subsections show that this is not a near-miss on hard instances but a structural property of the system: the model never predicts the Humanities field area at all.

6.2. Prediction-Level Analysis

Gold labels for the test set are not public, but the submitted predictions themselves support several exact statements.

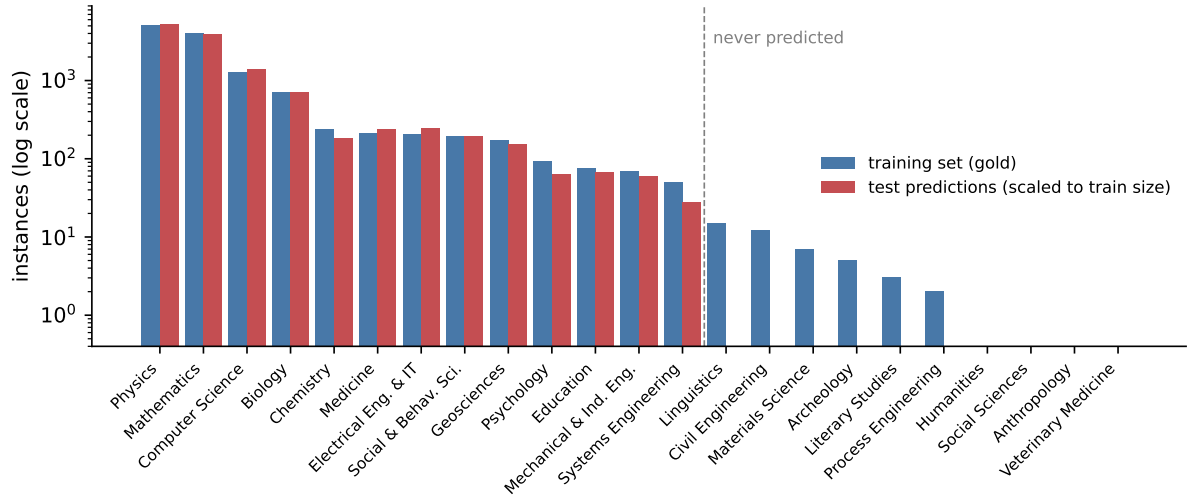

Figure 1: Gold training distribution vs. predicted test distribution (rescaled to the training-set size) per discipline, log scale. Disciplines are ordered by training frequency. Every discipline right of the dashed line (≤ 15 training examples) is never predicted.

Hierarchy consistency. All 3,106 submitted predictions satisfy $\pi(\hat{d}) = \hat{f}$: zero violations, as guaranteed by the constrained decoder of Section 4.5. Structural validity of the output space therefore holds exactly, not approximately.

Marginal fidelity. The predicted field-area distribution tracks the training distribution closely: *Natural Sciences* 75.9% (train 76.5%), *Engineering Sciences* 13.9% (13.0%), *Life Sciences* 7.6% (7.4%), *Social Sciences* 2.6% (2.9%), and *Humanities* 0.0% (0.2%). In aggregate, the ensemble estimates the training marginals very accurately.

The tail is never predicted. Figure 1 compares the gold training distribution with the (rescaled) predicted test distribution at the discipline level. Every discipline with at least 50 training examples is predicted at approximately its training-set rate. None of the ten disciplines with at most 15 training examples is ever predicted. This includes all five Humanities disciplines (Linguistics: 15 training examples, Archeology: 5, Literary Studies: 3, Anthropology and Humanities: 0) as well as Civil Engineering (12), Materials Science (7), Process Engineering (2), Veterinary Medicine (0), and the Social Sciences discipline label (0). Because no Humanities discipline is ever emitted and the decoder enforces $\pi(\hat{d}) = \hat{f}$, the Humanities *field* is never emitted either, which fully explains the 0.00 cell in Table 2.

6.3. Error Analysis: The Humanities Partition

The published Humanities column in Table 2 takes only values that are multiples of 0.10 (0.60, 0.40, 0.30, 0.20, 0.00), consistent with a gold Humanities partition of ten test instances, about 0.3% of the test set carrying 20% of the ranking metric. The implied counterfactuals are straightforward to compute. Our macro-average deficit to the winning run is $0.778 - 0.682 = 0.096$. Had our system labeled five of the ten Humanities instances correctly, its macro-average would have been 0.78, level with the winning entry; six correct (the winner's own Humanities score) would have placed it first at 0.80, since every other cell of our row is already at or within 0.02 of the leaderboard ceiling. Conversely, the winning run's advantage comes entirely from those instances: it trails our system on Life Sciences, Engineering Sciences, and Natural Sciences. Under this evaluation, the ranking among the top systems was decided almost exclusively by behavior on a small number of instances from a field area with 23 training examples.

7. Discussion

7.1. What the Results Establish

Head-class performance is saturated. On Natural Sciences, nine of the twelve submitted runs score 0.91–0.92, across encoder families and system designs that differ substantially. The same compression, slightly weaker, holds for Life, Engineering, and Social Sciences among the top six runs. Encoder capacity and training sophistication are evidently not the differentiator on the head of the distribution; the leaderboard ordering is decided on the tail.

Hierarchy-constrained inference guarantees valid outputs at no cost. The constraint guarantees, and the submitted file confirms with zero violations in $3{,}106$ predictions, that every output is a taxonomically legal field–discipline pair, at no training cost and no measurable cost to head-class accuracy (where, as noted, our system sits at the ceiling). Any downstream consumer of these predictions can rely on structural validity unconditionally.

Accurate marginals do not imply tail coverage. Section 6.2 shows the ensemble reproduces the training marginals almost exactly while assigning no test predictions at all to classes with ≤ 15 training examples. The two facts are related: a system can match aggregate marginals near-perfectly while being certain to score 0.00 on every rare class.

7.2. Why the Ensemble Never Predicts the Tail

Four mechanisms in our design compound against rare classes, and we highlight them because each is a standard, individually well-motivated choice. (i) *Unweighted cross-entropy*: with 0.2% prevalence, the loss contribution of Humanities examples is negligible, and the loss-minimizing response to an ambiguous instance is a head class. (ii) *Probability averaging across 10–11 models*: ensembling reduces variance, but rare-class predictions are precisely the low-agreement, high-variance events that averaging suppresses; a single fold model occasionally predicting Linguistics is overruled by ten that do not. (iii) *Confidence-thresholded pseudo-labeling*: at threshold 0.90, the pseudo-labels added to training are overwhelmingly head-class instances, so this stage strictly reinforces the existing imbalance. (iv) *Field-first constrained decoding*: a Humanities discipline can only be emitted if the field head first predicts Humanities, so tail suppression at the field level forecloses the discipline level entirely. None of these mechanisms is an implementation error; taken together, they produce a system optimized for the aggregate accuracy that the training distribution rewards, and penalized by the macro-averaged metric that the task ranks on.

7.3. Limitations

Three limitations of the present study are worth stating explicitly. First, gold labels for the test set are not released, so our test-time analysis is restricted to the official aggregate scores and to properties of the predictions themselves; instance-level error attribution is not possible. Second, we do not report per-component ablations: isolating the contribution of FGM, multi-sample dropout, the second encoder, and pseudo-labeling requires additional training runs that could not be repeated before the working-notes deadline, so the official scores evaluate the pipeline as a whole and the individual contributions remain entangled. Third, hierarchy-constrained inference, as formulated, predicts the field first and then the discipline conditional on the field; as discussed above, this compounds tail suppression across levels. A discipline-first decoding with a learned field–discipline reconciliation step, or a joint structured-output objective that internalizes the hierarchy during training, are natural directions for future work.

7.4. Implications for Hierarchical Scientific Classification

Our results have two implications for hierarchical text classification under severe class imbalance. The first is that decoupling consistency enforcement from the training objective, by constraining only at inference time, is a low-cost and effective design choice that does not require modifying the loss surface and can be applied on top of any probabilistic two-head classifier. The second is that, in the regime where the long tail of the class distribution contains classes with very few training examples (here, as few as 2–15 at the discipline level and 23 at the field level), the achievable accuracy ceiling is shaped less by the choice of encoder than by the treatment of the imbalance itself. Future systems on tasks with comparable distributions would likely benefit more from explicit long-tail interventions (class-balanced loss, focal loss, distillation from a stronger teacher with broad world knowledge) than from further scaling of the ensemble.

8. Conclusion

We presented a hierarchy-constrained multi-task ensemble for scientific research area classification under the SimpleText 2026 Task 3 evaluation setting. The system combines two complementary transformer encoders (DeBERTa-v3-large, SciBERT), stratified 5-fold cross-validation, FGM adversarial training, multi-sample dropout, taxonomy-constrained inference, and confidence-thresholded pseudo-labeling, and is reproducible on a single GPU. On the official test set our two submitted runs obtain identical scores: Exact Match of 0.92 (Natural Sciences, tied-best), 0.81 (Life Sciences), 0.86 (Engineering Sciences), and 0.82 (Social Sciences), placing top-two on all four, but only 0.00 on Humanities, for a macro-average of 0.68 and fourth place among seven teams. Our prediction-level analysis traces the Humanities zero to a structural property shared by several standard design choices: the ensemble reproduces the training marginals almost exactly and never predicts any label with at most 15 training examples, so a field area comprising 0.2% of the training data, and by our estimate ten test instances, decided the final ranking.

Three directions appear most promising for further work on this problem. First, replacing the post-hoc hierarchy constraint with a structured-output objective that internalizes the taxonomy during training, for instance through hierarchy-aware label embeddings or a level-conditional decoder. Second, explicit interventions on the long tail of the discipline distribution (class-balanced or focal losses, balanced batch sampling, targeted data augmentation for the lowest-frequency classes, or few-shot prompting of a large language model reserved specifically for suspected rare-class instances), which our analysis shows are the dominant determinant of ranking on this task. Third, distillation of soft labels from a larger instruction-tuned model with broad scientific knowledge, used either as an additional ensemble member or as a teacher for the existing encoder ensemble.

Reproducibility. Our code, the exact submitted predictions, and the data setup are publicly available.[1] The pipeline is contained in a single Jupyter notebook; one H200-class GPU is sufficient to reproduce all experiments.

Declaration on Generative AI. During the preparation of this work, the authors used generative AI tools to assist with code refactoring and manuscript editing. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

Acknowledgments

Compute resources were provided by the University DataLab at SRH University Heidelberg. We thank the SimpleText 2026 organizers for curating the task and releasing the data.

[1] https://github.com/Pratik25priyanshu20/simpletext-2026-research-area-classification

References

[1] L. Ermakova, S. Bertin, R. McCreadie, I. Ovchinnikova, L. Kosseim, Overview of the CLEF SimpleText track: Scientific text simplification, generation, and evaluation, in: Working Notes of CLEF, 2024. See also https://simpletext-project.com/.

[2] SimpleText Organizers, SimpleText track task 3: Research area classification, https://github.com/Gautamshahi/SimpleText-Track-Task-3-Research-Area-Classification, 2026.

[3] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, in: International Conference on Learning Representations (ICLR), 2023.

[4] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, in: Proceedings of EMNLP-IJCNLP, 2019.

[5] T. Miyato, A. M. Dai, I. Goodfellow, Adversarial training methods for semi-supervised text classification, in: International Conference on Learning Representations (ICLR), 2017.

[6] C. N. Silla, A. A. Freitas, A survey of hierarchical classification across different application domains, Data Mining and Knowledge Discovery 22 (2011) 31–72.

[7] J. Wehrmann, R. Cerri, R. C. Barros, Hierarchical multi-label classification networks, in: International Conference on Machine Learning (ICML), 2018.

[8] Y. Mao, J. Tian, J. Han, X. Ren, Hierarchical text classification with reinforced label assignment, in: Proceedings of EMNLP-IJCNLP, 2019.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019.

[10] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced BERT with disentangled attention, in: International Conference on Learning Representations (ICLR), 2021.

[11] R. Caruana, Multitask learning, Machine Learning 28 (1997) 41–75.

[12] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations (ICLR), 2015.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations (ICLR), 2018.

[14] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, Advances in Neural Information Processing Systems (NIPS) (1995).

[15] D.-H. Lee, Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: ICML Workshop on Challenges in Representation Learning, 2013.

[16] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, C. Raffel, FixMatch: Simplifying semi-supervised learning with consistency and confidence, in: Advances in Neural Information Processing Systems (NeurIPS), 2020.

[17] H. Inoue, Multi-sample dropout for accelerated training and better generalization, arXiv preprint arXiv:1905.09788 (2019).