Infimobius at #SMM4H-HeaRD 2026: Multi-Seed DeBERTa Ensemble for Flu Vaccination and Testing Status Classification

Suhani Singh Charan
CSED, IITR
suhani_sc@cs.iitr.ac.in

Pradyumn Kejriwal
CSED, IITR
pradyumn_k@cs.iitr.ac.in

Raksha Sharma
CSED, IITR
raksha@cs.iitr.ac.in

Rudra
CSED, IITR
rudra@cs.iitr.ac.in

Abstract

This paper describes FLUENS (**Flu EN**semble **S**ystem), our submission to the Social Media Mining for Health (SMM4H) 2026 Shared Task 3, which targets fine-grained classification of flu vaccination and flu testing statuses from tweets. FLUENS builds on the `microsoft/deberta-v2-xlarge` pre-trained language model and employs a multi-seed ensemble strategy in which five models, each initialized with a different random seed and trained on the full training set, are aggregated through soft-voting over averaged softmax probabilities. We additionally incorporate balanced class weights to mitigate severe label imbalance and apply a two-stage learning rate schedule that separately controls the encoder and classification head. On the development set, FLUENS achieves a macro F1 of 79.64% and micro F1 of 85.56% on the flu vaccination sub-task, and a macro F1 of 96.35% and micro F1 of 97.04% on the flu testing sub-task, substantially outperforming a `roberta-base` baseline across all metrics.

1 Introduction

Social media platforms such as Twitter provide a rich, real-time stream of self-reported health information. Mining these data for public health intelligence, particularly regarding influenza vaccination uptake and testing behaviour, has become an active area of research (**?**). However, automatically classifying the fine-grained status of flu vaccination or flu testing from individual tweets presents several challenges: (i) the informal, noisy nature of social media text, (ii) highly imbalanced label distributions, and (iii) subtle semantic distinctions between categories such as *Currently-Vaccinated* versus *Previously-Vaccinated*.

The SMM4H 2026 Shared Task 3 formalises this problem as two parallel five-class text classification tasks, one for flu vaccination status and one for flu testing status. In this paper, we present FLUENS, which addresses these challenges through three key design decisions:

1. **Adopting a larger language model**: replacing a standard `roberta-base` encoder with the substantially larger `microsoft/deberta-v2-xlarge` model, which leverages disentangled attention and enhanced mask decoding (**?**);

2. **Multi-seed ensembling**: training five models with distinct random seeds on the full training data and averaging their softmax outputs at inference time;

3. **Class-balanced training**: dynamically computing inverse-frequency class weights to counteract severe label imbalance.

2 Task Description

SMM4H 2026 Task 3 consists of two independent sub-tasks, each formulated as a five-class classification problem over tweets:

Flu Vaccination Status (Flu Shot). Given a tweet mentioning flu vaccination, assign one of the following labels: *Currently-Vaccinated*, *Currently-Unvaccinated*, *Previously-Vaccinated*, *Possibly-Vaccinated*, or *Other*.

Flu Testing Status (Flu Test). Given a tweet mentioning flu testing, assign one of the following labels: *Currently-Positive*, *Currently-Negative*, *Previously-Positive*, *Previously-Negative*, or *Other*.

The training set comprises 1,977 tweets for flu vaccination and 990 tweets for flu testing, with corresponding development sets of 270 and 135 tweets, respectively. Both sub-tasks exhibit substantial class imbalance, with the *Other* class dominating and minority classes (e.g., *Previously-Vaccinated*, *Previously-Positive*) being underrepresented.

Proceedings of the 11th Social Media Mining for Health Research and Applications (SMM4H-HeaRD 2026) Workshop and Shared Tasks, pages 260–263

3 System Description

This section describes the complete pipeline underlying FLUENS. We begin with the text preprocessing steps applied to all models (§**??**), then introduce the baseline system used for comparison (§**??**), followed by the architecture and design of FLUENS (§**??**), and finally the training configuration (§**??**).

3.1 Text Preprocessing

Following common practice for social media NLP, we apply lightweight normalisation to the raw tweets before tokenization: all user mentions matching the pattern `@USER*` are replaced with `@user`, and all URL tokens matching `HTTPURL*` are replaced with `http://url`. HTML entities such as `&` are decoded, and extraneous whitespace is collapsed. No other preprocessing (e.g., lowercasing, stemming) is performed, preserving the semantic cues that the pre-trained model can exploit.

3.2 Baseline System

Our baseline is a standard fine-tuning pipeline using `roberta-base` (**?**) via the HuggingFace Transformers library (**?**). For each sub-task, we independently fine-tune a `roberta-base` model with the default `AutoModelForSequenceClassification` head for 3 epochs using a batch size of 16 and a maximum sequence length of 128 tokens. Standard cross-entropy loss is used without any class weighting or additional regularisation. The best checkpoint is selected based on evaluation loss.

3.3 FLUENS

FLUENS addresses the shortcomings of the baseline through three key improvements described below.

Backbone Model. We replace `roberta-base` (125M parameters) with `microsoft/deberta-v2-xlarge` (∼900M parameters), which incorporates two architectural enhancements over standard transformer models (**?**): (i) *disentangled attention*, which separates content and position information into distinct vectors and computes attention using disentangled matrices, and (ii) *enhanced mask decoding*, which incorporates absolute position information at the decoding layer. We extract the `[CLS]` token representation from the final hidden layer and pass it through a custom two-layer classification head consisting of a dropout layer ($p = 0.1$), a linear projection to half the hidden dimension ($h/2$), a GELU activation (**?**), a second dropout layer ($p = 0.1$), and a final linear projection to the number of classes.

Multi-Seed Full-Data Ensemble. A common approach to build ensembles in NLP is k-fold cross-validation, where each fold uses a fraction of the data for validation, effectively reducing the amount of training data per model. Given our limited dataset sizes ($\leq$1,977 samples), we instead adopt a *multi-seed full-data* strategy: we train five separate models, each using 100% of the training data but with different random seeds (42, 123, 456, 789, 2024) governing parameter initialisation, data shuffling, and dropout masks.

At inference time, each model produces a softmax probability distribution over the label set. These five distributions are averaged element-wise (soft voting), and the class with the highest averaged probability is selected as the final prediction:

$$\hat{y} = \arg\max_{c} \frac{1}{N} \sum_{i=1}^{N} P_i(y = c \mid x) \qquad (1)$$

where $N = 5$ is the number of seed models and P_i denotes the softmax output of the i-th model. This soft-voting mechanism captures agreement and uncertainty across models more effectively than hard voting (majority label).

Class-Balanced Training. To address label imbalance, we compute balanced class weights inversely proportional to class frequency using the `sklearn compute_class_weight` utility (**?**). These weights are integrated directly into the cross-entropy loss, increasing the gradient contribution from minority classes.

3.4 Training Details

Each constituent model in FLUENS is optimised with AdamW (**?**) using differential learning rates: 1×10^{-5} for the pre-trained encoder parameters and 3×10^{-5} for the classification head. We apply a cosine learning rate schedule with a 10% warmup ratio. Training proceeds for up to 12 epochs with early stopping (patience of 5) based on macro F1 on the development set. The effective batch size is 24 (batch size 12 $\times$ gradient accumulation of 2), and the maximum sequence length is set to 192 tokens. Mixed-precision (FP16) training (**?**) is employed to reduce memory consumption and accelerate training. Gradients are clipped to a maximum norm

	Flu Shot		Flu Test	
System	Mi-F1	Ma-F1	Mi-F1	Ma-F1
Baseline	82.59	74.74	88.89	73.00
FLUENS	**85.56**	**79.64**	**97.04**	**96.35**
Δ	+2.97	+4.90	+8.15	+23.35

Table 1: System performance (%) on the development set. Mi-F1 = Micro F1; Ma-F1 = Macro F1. Δ denotes absolute improvement of FLUENS over the baseline.

Label	P	R	F1
Flu Vaccination (Flu Shot)			
Other	95.51	87.63	91.40
Possibly-Vaccinated	77.42	77.42	77.42
Currently-Unvaccinated	95.38	91.18	93.23
Currently-Vaccinated	80.33	87.50	83.76
Previously-Vaccinated	45.83	61.11	52.38
Flu Testing (Flu Test)			
Other	100.00	95.83	97.87
Currently-Negative	91.67	100.00	95.65
Currently-Positive	88.89	100.00	94.12
Previously-Negative	88.89	100.00	94.12
Previously-Positive	100.00	100.00	100.00

Table 2: Per-class precision (P), recall (R), and F1 (%) of FLUENS on the development set.

of 1.0, and weight decay of 0.01 is applied to all non-bias parameters.

4 Results

4.1 Main Results

We evaluate both the baseline and FLUENS on the gold-standard development sets using the official evaluation script, which reports micro-averaged F1, precision, and recall. We additionally report macro-averaged F1, which gives equal weight to all classes regardless of size and is therefore more informative for imbalanced datasets.

Table **??** presents the main results. FLUENS outperforms the baseline across both sub-tasks and both metrics. The improvements are particularly striking for the flu testing sub-task, where macro F1 jumps from 73.00% to 96.35% (+23.35 absolute points), indicating dramatically improved performance on minority classes. For the flu vaccination sub-task, the gains are more moderate but still substantial: +2.97 micro F1 and +4.90 macro F1.

4.2 Per-Class Analysis

Table **??** shows the per-class performance of FLUENS. For the flu testing sub-task, the model achieves perfect or near-perfect recall across all five classes, with the only errors being minor precision reductions in the *Currently-Positive* and *Previously-Negative* classes (88.89%). The *Previously-Positive* class, despite having only 4 samples, achieves perfect F1.

For the flu vaccination sub-task, which is inherently more challenging due to larger class count and subtler semantic distinctions, performance is strong for the majority classes (*Other*: 91.40 F1, *Currently-Unvaccinated*: 93.23 F1) but lower for the rarest class *Previously-Vaccinated* (52.38 F1), which has only 18 development samples. Notably, even this lowest-performing class shows improvement over the baseline (52.38 vs. 51.85 F1), and the *Possibly-Vaccinated* class improves markedly (77.42 vs. 62.96 F1).

5 Analysis

Why the large gain on Flu Test? The flu testing sub-task benefits disproportionately from FLUENS for two reasons. First, it has a smaller label space with more semantically distinct categories—testing *positive* versus *negative* involves clear lexical signals (e.g., "positive", "negative", "came back"). Second, the baseline `roberta-base` model severely struggled with minority classes (e.g., *Previously-Positive*: 40.00 F1), likely due to the absence of class weighting and the small training set (990 samples). The combination of a larger model, balanced class weights, and ensembling in FLUENS effectively addresses these issues.

Ensemble diversity. By using different random seeds, each model encounters the training data in a different order and has different dropout masks and parameter initialisations. This creates sufficient diversity for the ensemble to smooth out individual model errors. The full-data training strategy ensures that no model is disadvantaged by seeing fewer training examples, a critical consideration given our limited data.

Limitations

FLUENS has several limitations. First, the *Previously-Vaccinated* class in the flu vaccination task remains challenging (52.38 F1), indicating that additional methods may still be required for

extremely rare classes. Second, FLUENS relies on a very large pre-trained model (∼900M parameters), and training an ensemble of five such models is computationally expensive for both experimentation and deployment. Third, model selection and early stopping are based on the same development splits, which may introduce some degree of development-set overfitting when datasets are small. Fourth, several reported per-class results are estimated from very few examples (e.g., only 4 *Previously-Positive* instances in flu testing), so these estimates can be unstable.

Acknowledgments

We thank the SMM4H 2026 shared task organisers for providing the datasets and evaluation scripts. We also acknowledge the creators of the DeBERTa and RoBERTa models and the HuggingFace Transformers library for making large-scale pre-trained language models accessible for research.

A Hyperparameter Summary

Table **??** summarises the hyperparameters used for the baseline and FLUENS.

Hyperparameter	**Baseline**	**FLUENS**
Pre-trained model	roberta-base	deberta-v2-xlarge
Max sequence length	128	192
Batch size	16	12
Gradient accumulation	1	2
Effective batch size	16	24
Epochs	3	12
Learning rate (encoder)	5e-5	1e-5
Learning rate (head)	5e-5	3e-5
LR scheduler	linear	cosine
Warmup ratio	0.0	0.1
Weight decay	0.0	0.01
Dropout	0.1	0.1
Class weighting	No	Yes (balanced)
FP16	No	Yes
Early stopping	No	Yes (patience=5)
Ensemble seeds	—	42, 123, 456, 789, 2024

Table 3: Hyperparameter comparison between the baseline and FLUENS.