# PAST-TIDE: Prototype-Anchored Statement Tuning with Topic-Invariant Normalization for Stance Detection

**Md. Shakhoyat Rahman Shujon**[1], **MD Jahid Hasan Jim**[1], **Md. Milon Islam**[1], **Md. Rezwanul Haque**[2], **Fakhri Karray**[2,3]
[1]Department of Computer Science and Engineering, Khulna University of Engineering & Technology
skt104.shujon@gmail.com, mdjahidhasanjim277@gmail.com, milonislam@cse.kuet.ac.bd
[2]Department of Electrical and Computer Engineering, University of Waterloo
[3]Department of Machine Learning, Mohamed bin Zayed University of Artificial Intelligence
{rezwan, karray}@uwaterloo.ca

**Abstract**

We introduce **PAST-TIDE**, our stance detection system addressing both subtasks of the StanceNakba Shared Task at NakbaNLP@LREC 2026. The main idea is statement tuning. We redefine stance as cloze-style masked language modeling (MLM), letting a verbalizer map label words to stance categories through the pre-trained MLM head rather than appending a randomly initialized classification head. We complement this with prototypical contrastive learning, which uses learnable class prototypes for batch-size independent contrastive training, and topic-conditional layer normalization for cross-topic Arabic stance detection. **PAST-TIDE** achieves macro-F1 scores of 0.75 for Subtask A and 0.74 for Subtask B on the official leaderboard, indicating that minimal architectural additions to a pre-trained model can remain competitive in low-resource settings.

**Keywords:** stance detection, prompt-based classification, contrastive learning, cross-topic generalization, low-resource multilingual NLP.

## 1. Introduction

Understanding a person's stance on a given topic remains a challenging problem in computational argumentation (Mohammad et al., 2016; Küçük and Can, 2020). The field has moved from handcrafted features to pre-trained language model fine-tuning (Ghosh et al., 2019; Li and Caragea, 2021) to prompt-based reformulations (Schick and Schütze, 2021; Gao et al., 2021), yet multilingual stance detection remains underexplored. Most of the existing work focuses on benchmarks in English alone with thousands of labeled samples, leaving open the question of whether prompt-based methods remain effective when labels are limited and the target language is not English. The StanceNakba Shared Task (Aldous et al., 2026) addresses exactly the above gap. It defines two subtasks: Subtask A, stance at the actor-level in English (Pro-Palestine/Pro-Israel/Neutral), and Subtask B, stance cross-topic in Arabic (Favor/Against/Neither). Both tasks use relatively small datasets, with 1,401 and 1,205 labeled samples, split into 70/15/15 for train, development, and test. A standard [CLS] classifier head lacks any pre-trained initialization of its own, and that pre-training/fine-tuning gap degrades performance at this scale.

Our architecture, **PAST-TIDE**, is based on one idea: the classification capacity already exists within the pre-trained MLM head, and the task is to access it rather than replace it. The major contributions of this paper are as follows. First, statement tuning redefines stance as cloze-style MLM, bypassing the randomly initialized [CLS] head with a multi-token verbalizer that averages predictions across register-diverse label words. Second, prototypical contrastive learning (PCL) replaces in-batch negatives with $K{=}3$ class prototypes, where these prototypes are updated by gradients and remain persistent, giving stable anchors even at small batch sizes. Third, topic-conditional layer normalization (T-CLN), inspired by conditional normalization in vision, generates per-topic normalization parameters for cross-topic transfer.

On the official leaderboard, **PAST-TIDE** placed $11^{th}$ in Subtask A (macro-F1: 0.75) and $7^{th}$ in Subtask B (macro-F1: 0.74), scoring macro-F1 of 0.79 on both during the testing phase. It is observed that architectures with more parameters (up to 2.6M additional weights) scored 30%–43% lower during development, whereas our proposed model adds fewer than 1.3M new parameters and no additional classification head. The source code is publicly available.[1]

The paper is organized as follows. In Section 2, we review related work, and in Section 3, we explain our proposed architecture. Sections 4 and 5 demonstrate experiments and results, and Section 6 provides the conclusion.

[1]https://github.com/Shakhoyat/PAST-TIDE



*Proceedings of the 2nd International Workshop on Nakba Narratives as Language Resources (Nakba-NLP 2026) @ LREC 2026, pages 252–257*
11 May 2026. ©ELRA Language Resources Association (ELRA), 2026

## 2. Related Work

Stance detection has been studied primarily in English, from the SemEval-2016 shared task (Mohammad et al., 2016) to cross-target generalization (Allaway and McKeown, 2020). Küçük and Can (2020) provide a survey of this field up to 2020. Arabic stance work is still limited, with the MARASTA corpus (Charfi et al., 2024) as one of the few multi-dialectal resources. Prompt-based classification via cloze-style MLM (Schick and Schütze, 2021; Gao et al., 2021) shows consistent few-shot gains, but its application to stance detection, particularly in Arabic, has gained little attention. Contrastive learning for text classification typically relies on in-batch negatives (Khosla et al., 2020), which become inconsistent at small batch sizes. Prototypical networks (Snell et al., 2017) offer an alternative, though the episodic formulation does not suit standard fine-tuning directly. Multilingual encoders like mDeBERTa-v3 (He et al., 2023) have made cross-lingual transfer practical, yet combining them with prompt-based methods under data scarcity remains challenging. **PAST-TIDE** is designed for this problem: statement tuning for stance in a low-resource multilingual setting, with persistent prototypes replacing in-batch contrastive signals, and lightweight topic conditioning for Arabic cross-topic transfer.

## 3. System Description

The proposed architecture covers both subtasks, with T-CLN switching on only for Subtask B (Fig. 1). The idea is that a pre-trained MLM head already understands words like *support* and *against*, and stance labels are just vocabulary. Hence, we use the head directly for classification rather than replacing it. The methods, including PCL, T-CLN, and R-Drop, then solve specific failure modes: gradient collapse in small batches, cross-topic drift, and a weak decision boundary.

### 3.1. Backbone

Our encoder is mDeBERTa-v3-base (He et al., 2023) (280M parameters, hidden dimension 768), chosen because it includes a pre-trained MLM head that we reuse directly. A single checkpoint is used for both subtasks.

### 3.2. Statement Tuning

Stance detection is redesigned as a cloze-style MLM task, instead of adding a randomly initialized classification head to the [CLS] token. For each input text $x$ and target $t$, we construct a statement:

> `[CLS]` $x$ `[SEP]` Regarding $t$, this author's stance is `[MASK].` `[SEP]`

The phrasing "this author's stance is `[MASK]`" was chosen because the MLM head has learned many sentences of the form "X is `[ADJ]`" during pre-training, so the slot naturally provides adjective-like predictions where `[SEP]` denotes the separator token. In other variations, we tried placing *pro* or *against* in syntactic positions where they rarely occur in natural text, and this caused lower performance.

At the `[MASK]` position, the pre-trained MLM head generates vocabulary logits, and a *verbalizer* maps these logits to class probabilities by aggregating log-probabilities of semantically relevant tokens as mentioned in (1).

$$s_k = \frac{1}{|V_k|} \sum_{w \in V_k} \log P_{\text{MLM}}(w \mid \texttt{[MASK]}) \qquad (1)$$

where, $V_k$ is the verbalizer set for class $k$, $V_0$ = {*support, favor, pro, yes*} (Pro/Favor), $V_1$ = {*oppose, against, anti, no*} (Against), $V_2$ = {*neutral, unclear, none*} (Neutral). Multi-subword tokens use the first subword (primary semantics in SentencePiece). This setup requires *zero new parameters*. The MLM head is already pre-trained and semantically aligned with the verbalizer tokens, which is exactly what makes it effective at this sample scale.

**Verbalizer Design:** Each class gets 3–4 tokens spanning different registers: *support* and *favor* are formal, while *pro* and *yes* are informal. Mixing registers helps because social media text varies widely in formality. We tried Arabic tokens for Subtask B, but mDeBERTa has a limited Arabic vocabulary, and the performance decreased. English tokens, when passed through the shared embedding space, worked well for both languages.

### 3.3. Prototypical Contrastive Learning

We introduce PCL (Khosla et al., 2020) to structure the representation space, where $K{=}3$ learnable class prototypes $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\} \in \mathbb{R}^{768}$. The loss pushes each sample's `[MASK]` embedding toward its ground-truth prototype as shown in (2).

$$\mathcal{L}_{\text{PCL}} = -\log \frac{\exp(\text{sim}(\mathbf{h}, \mathbf{p}_y)/\tau)}{\sum_{j=1}^{K} \exp(\text{sim}(\mathbf{h}, \mathbf{p}_j)/\tau)} \qquad (2)$$

where $\tau{=}0.1$ and $\mathbf{h}$ is the L2-normalized `[MASK]` hidden state. Unlike Supervised Contrastive Learning (SupCon), whose gradient quality depends on in-batch negatives (only $\sim$5.3 expected at batch size 8), PCL's denominator always has exactly $K{=}3$ terms, keeping optimization consistent even at batch size 1. In practice, this turned out to be more important than we initially expected. Early experiments with SupCon showed fold-level F1 varying by up to $\pm 8$ points, which we attributed to random batches in which one class was severely

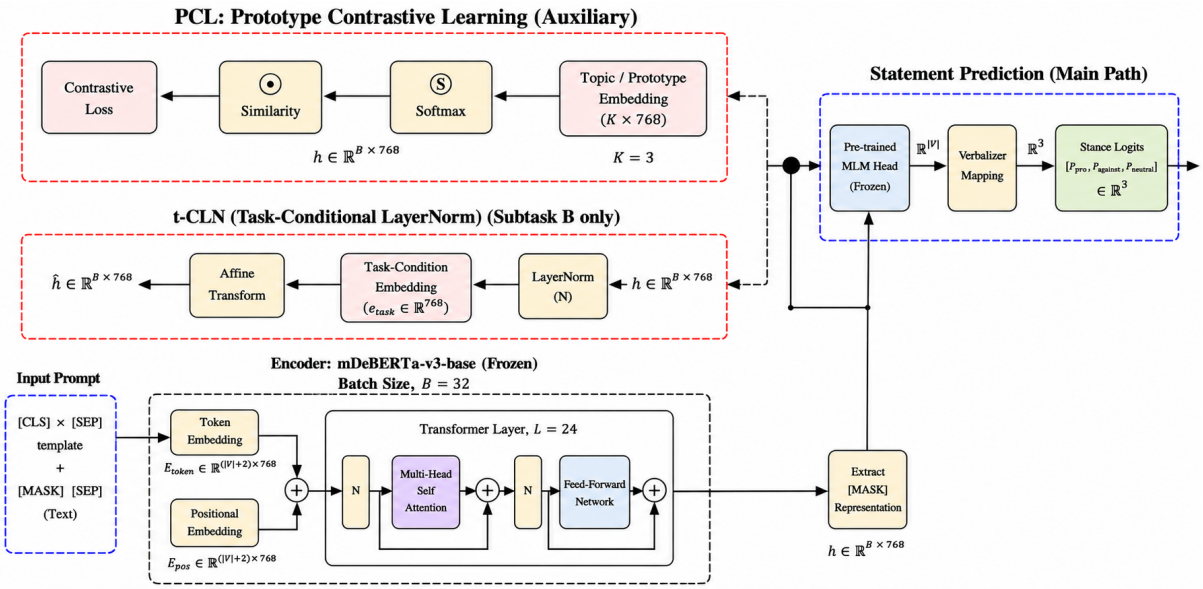


Figure 1: **PAST-TIDE** architecture. The cloze-wrapped input is passed to the encoder and the `[MASK]` hidden state passes through the pre-trained MLM head and verbalizer to generate stance logits. PCL prototypes branch from the same hidden state (auxiliary loss). T-CLN (Subtask B only) modifies the full encoder output before extraction.

underrepresented. PCL almost entirely eliminated this variance.

### 3.4. Topic-Conditional Layer Normalization

Subtask B requires stance detection across two Arabic topics with distinct vocabulary distributions. The static parameters $\gamma, \beta$ of standard layer normalization are replaced with topic-conditioned parameters generated by T-CLN (de Vries et al., 2017; Huang and Belongie, 2017), as shown in (3) and (4).

$$\mathbf{e}_t = \text{TopicEmbed}(t_{\text{id}}) \in \mathbb{R}^{64} \tag{3}$$

$$\text{T-CLN}(x, t) = \text{MLP}_\gamma(\mathbf{e}_t) \odot \frac{x - \mu}{\sigma} + \text{MLP}_\beta(\mathbf{e}_t) \tag{4}$$

where, $\text{MLP}_\gamma, \text{MLP}_\beta$ are two-layer networks ($64 \to 768 \to 768$, Tanh), initialized as identity ($\gamma_t \approx 1, \beta_t \approx 0$). T-CLN is applied after the last transformer layer and before the MLM head; it is only activated for Subtask B. The identity initialization is important: it means T-CLN starts as a no-operation and gradually learns topic-specific shifts during training. We found that the normalization parameters diverged early and destabilized the MLM head without this proper initialization.

### 3.5. R-Drop Regularization and Training

R-Drop (Liang et al., 2021) performs two forward passes with different dropout masks and minimizes their symmetric KL divergence $\mathcal{L}_{\text{R-Drop}}$, regularizing the decision boundary in low-resource settings. The total training loss is calculated in (5).

$$\mathcal{L} = \mathcal{L}_{\text{focal}} + 0.1 \cdot \mathcal{L}_{\text{PCL}} + 1.0 \cdot \mathcal{L}_{\text{R-Drop}} \tag{5}$$

where, $\mathcal{L}_{\text{focal}}$ is a class-weighted focal loss (Lin et al., 2017) ($\gamma$=2.0, label smoothing 0.1) applied to the verbalizer logits, averaged over both R-Drop passes. Training proceeds in two stages: Stage 1 (epochs 0–1) freezes the encoder and MLM head, updating only PCL prototypes and T-CLN. Stage 2 (epochs 2–9) unfreezes everything with layer-wise learning rate decay (Clark et al., 2020) (decay 0.95/layer, base LR $2 \times 10^{-5}$, $5\times$ multiplier for heads).

## 4. Experimental Setup

**Dataset:** The StanceNakba 2026 dataset (Aldous et al., 2026; Charfi et al., 2024) provides 1,401 English samples for Subtask A and 1,205 Arabic samples for Subtask B, each split into 70/15/15 for train, development, and test sets. Subtask A covers actor-level stance toward the Israeli-Palestinian conflict (Pro-Palestine/Pro-Israel/Neutral). The training data contains 980 samples (327/326/327 per class) with 210 development and 211 test samples. Subtask B is organized around two topics, *normalization with Israel* (577 samples) and *refugees in Jordan* (628 samples), with labels Favor/Against/Neither. The training split has 843 sam-

| Subtask | Rank | F1 | Acc | P | R |
|---|---|---|---|---|---|
| A (Actor-Level) | #11 | 0.745 | 0.749 | 0.763 | 0.749 |
| B (Cross-Topic) | #7 | 0.741 | 0.740 | 0.745 | 0.743 |

Table 1: Official leaderboard results (organizer evaluation). F1 = macro-F1, Acc = Accuracy, P = macro-precision, and R = macro-recall.

ples (286/298/259 per class), with 181 development and 181 test cases. The class distributions are almost balanced, though Subtask B's *Neither* class is somewhat smaller, making it difficult to learn. Both datasets are collected from social media, hence posts range from single-sentence to multi-paragraph arguments, and code-switching appears frequently in the Arabic data.

**Augmentation and Ensembling:** We double the training set using back-translation with MarianMT (Junczys-Dowmunt et al., 2018), pivoting through German[2] for Subtask A and English for Subtask B. Evaluation uses 5-fold stratified cross-validation. We save the best checkpoint per fold using early stopping with a patience of 3, then average the class probabilities across folds during the test phase. Arabic inputs go through a preprocessing step before tokenization: Alef and Ya forms are unified, diacritics are removed, and Tatweel characters are deleted.

**Implementation:** We utilized dual NVIDIA T4 GPUs (2×16 GB). The batch size is set to $8\times4=32$ after gradient accumulation, with a maximum sequence length of 256 tokens, FP16, and gradient checkpointing to stay within memory limits. We optimize with AdamW, increasing linearly for the first 10% of steps and then decaying on a cosine schedule.

## 5. Results and Analysis

Table 1 reports official leaderboard scores. All rows are server-evaluated submissions.

Table 2 shows the testing-phase results of the proposed architecture and its variants. During the testing phase, **PAST-TIDE** scored 0.79 macro-F1 on both subtasks, 16% above our best [CLS]-based system. The official leaderboard settled lower (0.75 / 0.74), consistent with the expected generalization gap. The testing phase scores were computed on a subset of the development data, while the final evaluation used the held-out test partition. The largest

[2]German's divergent word order (verb-final subordinate clauses, free constituent order) forces the round-trip to paraphrase rather than copy, unlike Romance pivots whose syntactic similarity to English yields near-identical back-translations. The Helsinki-NLP OPUS-MT EN↔DE models are also among the highest-quality available.

| Methods | A | B | Mean |
|---|---|---|---|
| Dual-Encoder ([CLS]) | 0.50 | 0.64 | 0.57 |
| Target-Calibrated ([CLS]) | 0.66 | 0.70 | 0.68 |
| Disentangled (MLM) | 0.46 | 0.64 | 0.55 |
| Frame-Anchored (SSA) | 0.48 | 0.42 | 0.45 |
| w/o T-CLN (MLM) | 0.75 | 0.74 | 0.75 |
| **PAST-TIDE (MLM)** | **0.79** | **0.79** | **0.79** |

Table 2: Macro-F1 scores of testing-phase submissions. SSA = Soft Semantic Anchors.

| | Subtask A | | | Subtask B | | |
|---|---|---|---|---|---|---|
| | Pro-P | Pro-I | Neut | Fav | Agn | Nei |
| Pro-P / Fav | **56** | 8 | 6 | **46** | 13 | 3 |
| Pro-I / Agn | 7 | **60** | 3 | 7 | **50** | 6 |
| Neut / Nei | 11 | 3 | **56** | 3 | 12 | **41** |

Table 3: Dev-set confusion matrices (210 / 181 samples). Rows: actual, columns: predicted, Pro-P = Pro-Palestine, Pro-I = Pro-Israel, Neut = Neutral, Fav = Favor, Agn = Against, and Nei = Neither.

gain came from statement tuning itself: the MLM head already has representations for words like *support* and *against*. Replacing SupCon with PCL reduced fold-level F1 variance by up to 8 points. The architectures with more parameters scored worse: the Disentangled variant added 2.6M parameters and reduced performance by 30%, and freezing the MLM head decreased performance by 43%.

**Confusion Matrix:** Table 3 demonstrates the confusion matrices for both subtasks. In Subtask A, most errors come from Neutral posts being misclassified as Pro-Palestine (11 of 14 total Neutral errors), while only 3 are predicted as Pro-Israel. This happens because words about ceasefire and civilian deaths are similar to words in Pro-Palestinian posts, so the model cannot separate them properly. The architecture confuses the two main classes almost equally (8 vs. 7). This means it can separate them, but has trouble with neutral posts. In Subtask B, the model over-predicts *against*: 13 pro instances are misclassified as *against* compared to 7 in the opposite direction, and 12 neutral instances are also predicted as *against*. We believe this happens because strong pro-stance Arabic writing uses emphatic words, dialect forms, and strong emotion, which looks similar to opposition, more than in English.

**Cross-Topic Transfer:** It is observed that removing T-CLN has no effect on Subtask A (single English topic), but drops Subtask B by 6% in the testing phase. This shows that the two Arabic topics are different in the representation space, and conditional normalization bridges them.

**Parameter Overhead:** In terms of overhead,

Subtask A adds just 2,304 new parameters (the PCL prototypes); Subtask B adds $\sim$1.28M for T-CLN. The MLM head and verbalizer contribute zero new weights, which is arguably the most important design choice. The entire classification capacity comes from weights that were already trained on hundreds of gigabytes of multilingual text.

## 6. Conclusion

**PAST-TIDE** obtained 0.75 and 0.74 macro-F1 on the official Subtask A and B leaderboards (macro-F1: 0.79 on both during the testing phase), by reusing rather than replacing the pre-trained MLM head. Our simpler proposed models outperformed larger ones with added parameters. The task design is more important than adding more modules to the model at this small data scale. Each module of the proposed architecture generalizes naturally: statement tuning to any label set with real vocabulary words, PCL to any setting where batch sizes are too small for in-batch negatives, and T-CLN to any multi-domain problem. In the future, we plan to automate verbalizer selection via gradient-based token search and test on benchmarks with finer granularity of the labels.

## Bibliographical References


Kholoud Khalil Aldous, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Kais Attia, and Wajdi Zaghouani. 2026. StanceNakba shared task: Actor and topic-aware stance detection in public discourse. In *Proceedings of the 15th International Conference on Language Resources and Evaluation (LREC'26)*, Palma, Spain.

Emily Allaway and Kathleen McKeown. 2020. Zero-shot stance detection: A dataset and model using generalized topic representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8913–8931. Association for Computational Linguistics.

Anis Charfi, Mabrouka Bessghaier, Andria Atalla, Raghda Akasheh, Sara Al-Emadi, and Wajdi Zaghouani. 2024. MARASTA: A multi-dialectal Arabic cross-domain stance corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11060–11069. ELRA and ICCL.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1xMH1BtvB.

Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C. Courville. 2017. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, volume 30, pages 6594–6604. Curran Associates, Inc.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830. Association for Computational Linguistics.

Shalmoli Ghosh, Prajwal Singhania, Siddharth Singh, Koustav Rudra, and Saptarshi Ghosh. 2019. Stance detection in web and social media: A comparative study. In *Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF 2019)*, pages 75–87. Springer.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=sE7-XhLxHA.

Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510. IEEE.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc.

Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Computing Surveys*, 53(1):1–37. Article 12.

Yingjie Li and Cornelia Caragea. 2021. Target-aware data augmentation for stance detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1850–1860. Association for Computational Linguistics.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 10890–10905. Curran Associates, Inc.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 4077–4087. Curran Associates, Inc.

## A. Limitations

Our design choices are based on cross-iteration failure analysis rather than a controlled ablation within a single architecture variant, owing to compute limits (dual T4, 9h on Kaggle). This means the improvements from each component may be mixed with other changes between runs, and a proper ablation study would make the results stronger. Verbalizer tokens were chosen by intuition about stance-relevant vocabulary. An automated search (e.g., gradient-based token selection) might find better tokens, particularly for Arabic where our lexical intuitions are weaker. Finally, the shared task data covers two specific geopolitical topics, and we have not yet tested whether the architecture transfers well to domains with different label semantics or number of classes.

## B. Hyperparameters

| Hyperparameter | Value |
|---|---|
| Backbone | mDeBERTa-v3-base |
| Max sequence length | 256 |
| Effective batch size | $8 \times 4 = 32$ |
| Learning rate (backbone) | $2 \times 10^{-5}$ |
| Head LR multiplier | $5\times$ |
| LLRD decay per layer | 0.95 |
| Weight decay | 0.01 |
| Warmup ratio | 0.1 |
| Focal loss $\gamma$ | 2.0 |
| Label smoothing | 0.1 |
| PCL $\tau$ / weight | 0.1 / 0.1 |
| R-Drop weight | 1.0 |
| T-CLN topic dim | 64 (Subtask B) |
| Training stages | 2 frozen + 8 fine-tuned |
| K-fold / patience | 5 / 3 |

Table 4: Hyperparameter configurations for **PAST-TIDE**.

## C. Ethics Statement

**PAST-TIDE** infers stance from surface-level linguistic cues; the predictions carry no implication about the authors' own views on the topics involved. All data were used strictly according to the shared task guidelines.

## D. Acknowledgements

We are grateful to the StanceNakba 2026 organizers for providing the datasets and evaluation infrastructure. All experiments were run on Kaggle's free GPU tier. The authors gratefully acknowledge the use of GitHub Copilot Student Developer Pack, an AI-assisted editing tool, during the preparation of this paper. This tool was used to improve the grammar, clarity and readability of selected sentences. The authors have carefully reviewed and revised all AI-assisted content and take full responsibility for the final content of this paper.