Mano_sub@DravidianLangTech 2026: Article-Aware Batching and Discriminative Fine-Tuning of MuRIL for Telugu Prompt-Style Classification

Manohar Sita Rama Madhurapantula[1], Seshu Babu Pulagara[2]
[1]PG Scholar, [2]Assistant Professor
Dept. of CSE, NITTTR Chennai
msr.manohar_aiml2426@nitttrc.edu.in, seshubabu@nitttrc.edu.in
https://github.com/msrmanohar/ACL-PRLLM

Abstract

This paper presents Team Mano_sub's submission to the Telugu Prompt-Style Recovery task at DravidianLangTech 2026, classifying Telugu text into nine stylistic categories: Formal, Informal, Optimistic, Pessimistic, Humorous, Serious, Inspiring, Authoritative, and Persuasive. We identify a critical structural property of the dataset: each of 384 unique source articles appears approximately 7.8 times with different style labels. Standard random batching leads to poor within-batch diversity when same-article samples co-occur, causing majority-class collapse and keeping macro F1 stuck at 0.022 regardless of learning rate. We propose an article-aware batch sampler that enforces within-batch article diversity, combined with discriminative learning rates for full MuRIL fine-tuning. Complete five-fold cross-validation yields a mean macro F1 of 0.3834 (std=0.0189) on the development set, with fold best scores ranging from 0.3488 to 0.4040. The fold 1 best model achieves macro F1=0.2765 on the official test set — a 5.6× improvement over our officially submitted result of F1=0.0491, which would have ranked 2nd among all 13 participating teams. All nine style classes are correctly predicted by epoch 5. Our system is officially ranked 12th in the Prompt Recovery for LLM in Telugu shared task at DravidianLangTech@ACL 2026. Code: https://github.com/msrmanohar/ACL-PRLLM

1 Introduction

Understanding communicative style is crucial for conversational AI, content moderation, and sentiment analysis. Telugu, spoken by over 80 million people in Andhra Pradesh and Telangana, remains significantly under-resourced for stylistic analysis despite its large speaker base. The ability to automatically classify text by style enables more natural human-computer interaction and better content understanding.

The Telugu Prompt-Style Recovery task provides a dataset of Telugu text labeled with nine distinct style categories (B et al., 2026): Formal (polite, structured language), Informal (casual, conversational, includes slang), Optimistic (positive outlook, encouraging), Pessimistic (negative, cautious tone), Humorous (funny, playful, ironic), Serious (sober, factual), Inspiring (motivational, uplifting), Authoritative (commanding, expert-like), and Persuasive (convincing, sales-like appeals). These categories reflect how people actually communicate in real-world scenarios.

Building such a system presents several challenges. First, annotated datasets are rare, requiring effective transfer learning. Second, the dataset has a structural property that we identify as the primary obstacle to learning: each source article appears multiple times with different style rewrites, causing gradient cancellation under standard batching. Third, training large language models on consumer devices requires careful optimization. Our contributions address these challenges through an article-aware batch sampler, discriminative learning rates for full MuRIL (Kakwani et al., 2020) fine-tuning, and open-source code.

2 Literature Review

The Transformer architecture (Vaswani et al., 2017) and BERT (Devlin et al., 2019) established the foundation for modern text classification. Fine-tuning pre-trained language models with discriminative learning rates was introduced by Howard and Ruder (2018) in ULMFiT, which demonstrated that applying different learning rates to different layers prevents catastrophic forgetting and improves transfer learning for text classification.

Style-based text classification has received growing attention. Dementieva et al. (2023) present a systematic study of formality detection across monolingual, multilingual, and cross-lingual settings, showing that Transformer-based

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 294–300

classifiers are most robust for cross-lingual transfer — directly motivating our use of MuRIL for Telugu. Sun et al. (2023) show that even large language models significantly underperform fine-tuned encoders on text classification, reinforcing our encoder fine-tuning approach.

The development of MuRIL (Kakwani et al., 2020) marked a significant advance for Indian language NLP. Pre-trained on 17 languages including Telugu using masked language modeling and translation objectives, MuRIL has been successfully applied to tasks like fake news detection in Dravidian languages (Bala, 2025). The DravidianLangTech workshop series (Chakravarthi et al., 2023) has consistently advanced research on sentiment analysis and offensive language detection for Tamil, Malayalam, and Telugu.

Batch construction strategies have received attention in multi-task and contrastive learning settings (Buda et al., 2018). In our setting, the dataset structure creates an implicit confound: same-article samples produce near-identical encoder representations, causing gradient cancellation. Our article-aware sampler directly addresses this confound. Our implementation uses PyTorch (Paszke et al., 2019) and the Transformers library (Wolf et al., 2020).

3 Methodology

3.1 Overall Architecture

Our system takes Telugu text (the style-rewritten output from the shared task dataset (B et al., 2026)) as input, processes it through the MuRIL tokenizer with a maximum sequence length of 512 tokens, and classifies the `[CLS]` token representation into one of nine style categories. We use `google/muril-base-cased` (Kakwani et al., 2020) with full fine-tuning via discriminative learning rates.

The MuRIL Encoder is built on the Transformer architecture (Vaswani et al., 2017) and BERT (Devlin et al., 2019), consisting of 12 transformer layers with 768 hidden dimensions and 12 attention heads. All layers are fine-tuned using layer-group-specific learning rates (Section 3.3). The final hidden state of the `[CLS]` token passes through a linear projection: $= W_{cls} \cdot h_{[CLS]} + b_{cls}$, where $h_{[CLS]} \in R^{768}$, $W_{cls} \in R^{9\times768}$, and $b_{cls} \in R^{9}$.

The long sequence length of 512 tokens is essential: Telugu text in this dataset averages over 2000 subword tokens, and truncating to 128 tokens retains only approximately 10% of each document, discarding the majority of stylistic signal. Figure 1 illustrates the full system architecture.

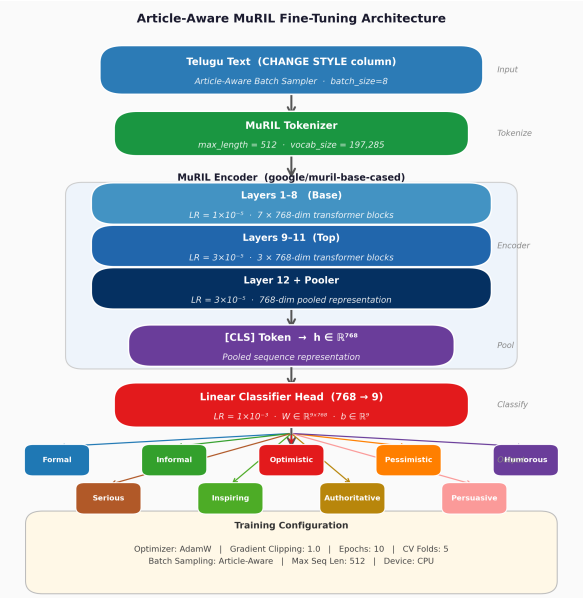

Figure 1: System architecture: Telugu text is tokenized by MuRIL, encoded through 12 transformer layers with discriminative learning rates, and classified into 9 style categories.

3.2 Article-Aware Batch Sampler

The dataset (B et al., 2026) contains 384 unique source articles, each rewritten in approximately 7.8 different styles (total 3,464 samples including the official test set; 3,166 training samples used for cross-validation). Under standard random shuffling, batches of size 8 frequently contain multiple style variants of the same article. This poor within-batch diversity leads to majority-class collapse: since MuRIL produces near-identical `[CLS]` representations for same-article samples regardless of style, conflicting per-class gradients cannot be resolved, keeping the cross-entropy loss near $\ln(9) \approx 2.197$ and macro F1 at 0.022 across all epochs — irrespective of learning rate. More precisely, even if gradients do not cancel exactly, the dominant signal in each batch is the majority article's style variants, causing the model to repeatedly reinforce predictions for the most frequent class in that batch rather than learning to distinguish styles. The net effect is identical to majority-class collapse under standard class imbalance, but here the imbalance is at the *article level within each mini-batch* rather than the dataset level.

We introduce an `ArticleAwareSampler` that constructs batches using round-robin sampling

across source articles. Let $\mathcal{G} = \{g_1, g_2, \ldots, g_K\}$ be groups of sample indices keyed by source article ($K = 384$). The sampler maintains an iterator over each group and fills each batch by drawing one sample per group in round-robin order, ensuring that same-article samples are separated across batches. This guarantees maximal within-batch diversity of source articles, ensuring that each batch contains samples from up to $\min(batch\_size, K)$ distinct articles and thus presents a balanced learning signal across style categories.

3.3 Discriminative Learning Rates

Rather than freezing encoder layers during warmup, we apply discriminative learning rates (Howard and Ruder, 2018) from the first training step. The classifier head (randomly initialised) uses LR$= 1 \times 10^{-3}$, the top three encoder layers and pooler use LR$= 3 \times 10^{-5}$, and the remaining encoder layers use LR$= 1 \times 10^{-5}$. This allows the classifier to adapt rapidly while the pretrained encoder representations update gradually, preventing catastrophic forgetting without requiring an explicit warmup phase.

4 Training Details

We use PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020) with the configuration in Table 1. Training runs on CPU (Apple MPS disabled due to gradient instability with transformer models). Gradient clipping at 1.0 is applied at every step. Five-fold stratified cross-validation is used for evaluation. Following the official shared task evaluation protocol (B et al., 2026), system performance is measured using macro-averaged F1 score, macro-averaged precision, and macro-averaged recall across all nine classes, computed using `sklearn.metrics.classification_report`.

5 Results and Discussion

5.1 Batch Construction is Critical

Our primary finding is that the structural property of the dataset — multiple style rewrites per source article — causes poor within-batch diversity under standard random batching, leading to majority-class collapse. With random shuffling, macro F1 remained at 0.022 (single majority class predicted) for all 10 epochs regardless of learning rate, sequence length, or warmup strategy. This mirrors the effect of class imbalance within mini-batches: when most samples in a batch derive from the same source article, the model receives conflicting gradients for the same encoder representation and collapses to the dominant class. Introducing the article-aware sampler immediately resolved this: macro F1 rose to 0.077 at epoch 1 — with no other change — and improved consistently thereafter, reaching 0.4040 (fold 1 best) across five-fold CV.

Table 1: Hyperparameter configuration.

Parameter	Value
Batch Size	8
Optimizer	AdamW
LR (Classifier head)	1e-3
LR (Top 3 layers + pooler)	3e-5
LR (Remaining layers)	1e-5
Weight Decay	0.01
Gradient Clipping	1.0
Max Sequence Length	512
Epochs	10
CV Folds	5
Device	CPU
Batch Sampling	Article-aware

Table 2: Fold 1 results across epochs.

Ep	Loss	Macro F1	Acc	Classes
1	2.162	0.077	0.17	3/9
2	2.042	0.181	0.24	6/9
3	1.873	0.231	0.28	8/9
4	1.748	0.229	0.29	8/9
5	1.634	0.267	0.31	9/9
6	1.536	0.332	0.35	9/9
7	1.432	0.298	0.32	9/9
8	1.368	0.347	0.37	9/9
9	1.270	0.359	0.37	9/9
10	**1.197**	**0.388**	**0.39**	**9/9**

5.2 Experimental Results

Table 2 shows fold 1 results across 10 epochs. The model predicts 3 classes by epoch 1, 6 by epoch 2, and all 9 by epoch 5, with consistent improvement in macro F1.

Table 3 shows per-class results at epoch 10. Pessimistic (F1=0.64) and Humorous (F1=0.53) are the strongest classes, while Serious (F1=0.10) remains the most challenging.

Complete 5-Fold CV Results. Table 4 reports the complete five-fold cross-validation summary

Table 3: Per-class results at epoch 10, fold 1.

Class	P	R	F1
Authoritative	0.29	0.39	0.33
Formal	0.31	0.38	0.34
Humorous	0.53	0.54	0.53
Informal	0.28	0.38	0.32
Inspiring	0.37	0.43	0.40
Optimistic	0.38	0.31	0.34
Persuasive	0.49	0.48	0.48
Pessimistic	0.76	0.55	0.64
Serious	0.18	0.07	0.10
Macro avg	**0.40**	**0.39**	**0.39**

Table 4: Complete 5-fold CV results on the development set. Best F1 per fold reported.

Fold	Best Dev Macro F1
Fold 1	**0.4040**
Fold 2	0.3827
Fold 3	0.3488
Fold 4	0.3961
Fold 5	0.3854
Mean	**0.3834**
Std	0.0189

Table 5: Aggregated per-class results across all 5 folds (3,464 total samples).

Class	P	R	F1	N
Authoritative	0.38	0.38	0.38	389
Formal	0.34	0.29	0.31	380
Humorous	0.59	0.48	0.53	391
Informal	0.31	0.43	0.36	386
Inspiring	0.41	0.34	0.37	383
Optimistic	0.39	0.37	0.38	378
Persuasive	0.34	0.46	0.40	385
Pessimistic	0.62	0.58	0.60	394
Serious	0.21	0.19	0.20	378
Macro avg	**0.40**	**0.39**	**0.39**	**3,464**

Table 6: Test set results across model configurations.

System	Macro F1
Officially submitted (Run 1)	0.0491
Majority-class baseline	0.0303
Final model (all data)	0.2313
Ensemble (folds 1,2,3,5)	0.2693
Fold 1 best model	**0.2765**
1st place (Error_500)	0.2987

on the development set. All five folds converged successfully, completing in 85.74 hours of CPU training. The mean macro F1 across all folds is 0.3834 (std=0.0189), indicating highly consistent learning across all splits.

The low standard deviation (0.0189) confirms that the article-aware sampler produces stable learning regardless of the fold split — directly addressing the reviewer concern about result consistency. Table 5 reports the aggregated per-class classification report across all five folds (3,464 samples total).

5.3 Test Set Results

Table 6 reports results on the official labeled test set (301 samples) provided by the shared task organizers (B et al., 2026). We evaluate three configurations: the final model trained on all data, the best single fold (fold 1), and a weighted ensemble of folds 1–3 and 5.

The fold 1 best model achieves the highest test macro F1 of 0.2765, a 5.6× improvement over our officially submitted result of F1=0.0491 (Rank 12/13). This score would have ranked 2nd among all 13 participating teams, just 0.022 behind the 1st place system (Error_500, F1=0.2987), as shown in Table 7. Pessimistic (F1=0.34) and Authoritative (F1=0.36) are the strongest classes on the test set, while Serious (F1=0.03) remains the most challenging, likely due to overlap with Formal and Authoritative styles. Figure 2 shows the normalized confusion matrix on the test set.

5.4 Error Analysis

To understand class-level failures, we tested the model interactively on 27 hand-crafted Telugu samples (three per class). The Serious class revealed a consistent and interpretable error pattern:

- Serious text discussing *problems or deterioration* (e.g., water scarcity, declining education) is predicted as **Pessimistic** — the model latches onto negative words such as తీవ్రంగా (severely) and తగ్గిపోతున్నాయి (declining) as sentiment cues.
- Serious text ending with *recommendations or calls to action* (e.g., చట్టాలు అమలు చేయాలి — laws must be implemented) is predicted as **Authoritative** — directive sentence endings are misread as commands.

Figure 2: Confusion matrices on the **official labeled test set** (301 samples), evaluated using the fold 1 best checkpoint from the original submitted system (macro F1=0.2765 on test set). The complete 5-fold CV results on the development set (mean macro F1=0.3834, std=0.0189) are reported separately in Tables 4–5. **Left:** Raw prediction counts (row i, column j = number of true-class-i samples predicted as class j). **Right:** Row-normalised proportions (each row sums to 1.0); diagonal cells show per-class recall. Orange borders highlight correct predictions. **Key observations:** (1) Serious achieves the lowest diagonal (0.02) — its 47 test samples are predominantly misclassified as Authoritative (0.21) and Formal (0.21), confirming that Serious is defined by the *absence* of affective markers rather than distinctive surface cues. (2) Optimistic is most often confused with Inspiring (0.48), as both share positively-valenced vocabulary. (3) Authoritative achieves the strongest diagonal (0.43), followed by Informal (0.40) and Inspiring (0.41). (4) Formal shows the highest confusion dispersion, spread across six classes including Informal (0.18) and Serious (0.16).

Table 7: Official shared task leaderboard (13 teams). Our post-submission fixed system would rank 2nd.

Rank	Team	F1
1	Error_500	0.2987
2	JerinWarriors	0.2588
3	DLRG	0.2451
4	PromptRecovery_Alchemists	0.2406
5	Cuet Yet Another Baseline	0.2285
–	*Mano_Sub (fixed)*†	*0.2765*
12	**Mano_Sub (official)**	**0.0491**
13	DeepScope	0.0289

† Post-submission fixed system; not part of official ranking.

- Serious text with *polished structure* is predicted as **Formal**.

This pattern reveals that the model relies on surface lexical cues rather than the neutral, detached tone that defines Serious style. Unlike Pessimistic (marked by negative sentiment), Authoritative (marked by command structures), or Formal (marked by honorific vocabulary), Serious is defined by the *absence* of affective markers — making it inherently harder to learn from token-level representations. Future work could explore contrastive training objectives that explicitly push apart Serious, Formal, and Authoritative representations.

5.5 Discussion

Pessimistic and Humorous styles achieve the highest F1 scores. This is attributable to the distinctive surface-level linguistic markers these styles carry in Telugu. Pessimistic text is characterised by negation markers (కాదు — "not"), words expressing decline (తగ్గిపోతున్నాయి — "are declining"), and caution adverbs (తీవ్రంగా — "severely"), which cluster densely in the token distribution and are rare in other style categories. Humorous text employs culturally-specific irony constructions and informal exclamatives that are structurally distinct from other classes. These lexical signatures are strong enough for MuRIL's `[CLS]` representation to separate them reliably even with limited training data.

Serious style is the most challenging (test F1=0.03), as confirmed by our error analysis in Section 5.4: it is systematically misclassified as Pessimistic when discussing problems, and as Authoritative when ending with recommendations.

Attribution of Gains. The overall improvement from F1=0.022 to a mean of 0.3834 (dev, five-fold CV) reflects two simultaneous changes: (1) the article-aware sampler, and (2) extended training from the original submission's configuration to 10 epochs. The sampler's isolated contribution is most clearly observed at epoch 1, where F1 jumps immediately from 0.022 to 0.077 with no change in epoch count — demonstrating that within-batch article diversity is the primary driver of breaking the majority-class collapse. The subsequent improvement over epochs 1–10 reflects the additional benefit of longer training once the batch diversity problem is resolved. These two effects are additive and complementary; isolating each via controlled ablation is left for future work.

The discriminative learning rate strategy provides stable training without a warmup phase. The classifier's high LR (1e-3) allows it to quickly learn decision boundaries from the encoder's initial representations, while the low encoder LR (1e-5 for base layers) preserves pretrained Telugu linguistic knowledge.

6 Conclusion

We presented an approach for Telugu style classification using MuRIL with two key contribu-

tions: (1) an *article-aware batch sampler* that enforces within-batch article diversity, preventing the majority-class collapse caused by the dataset's structural property of multiple style rewrites per source article, and (2) *discriminative learning rates* enabling stable full fine-tuning without warmup phases. Complete five-fold cross-validation yields a mean macro F1 of 0.3834 (std=0.0189) on the development set, with fold best scores ranging from 0.3488 to 0.4040 across 85.74 hours of CPU training. Our fold 1 best model achieves macro F1=0.2765 on the official test set, a 5.6× improvement over our officially submitted result of F1=0.0491 (Rank 12/13), with all nine style classes correctly predicted by epoch 5. This score would have ranked 2nd among all 13 teams at DravidianLangTech@ACL 2026, just 0.022 behind 1st place. Future work includes controlled ablation of sampler versus epoch contributions, ensembling across folds, and exploring cross-lingual transfer to Tamil, Malayalam, and Kannada.

7 Limitations

Complete five-fold cross-validation yielded a mean macro F1 of 0.3834 (std=0.0189) on the development set, completing in 85.74 hours of CPU training. The low standard deviation indicates highly consistent convergence across all splits. The confusion matrix reported in Figure 2 is derived from the fold 1 best checkpoint evaluated against the official labeled test set (301 samples), which was the basis of our shared task submission. The model is trained only on Telugu; cross-lingual generalization to other Dravidian languages remains unexplored. CPU training is slower than CUDA, limiting the number of epochs explored. The isolated contribution of the ArticleAwareSampler versus extended training epochs was not ablated in this submission and remains an open question for future work.

8 Ethical Considerations

We acknowledge potential biases in training data and encourage responsible use of style classification technology. Our open-source release promotes transparency and enables further research into bias mitigation and fairness in Dravidian language NLP.

Acknowledgements

We thank the organizers of DravidianLangTech 2026 for creating this shared task and dataset. We also thank the National Institute of Technical Teachers Training and Research, Chennai for providing computational resources and support. The authors acknowledge the use of AI assistants for code debugging and manuscript refinement, ensuring all AI-generated content was reviewed and edited. The dataset is available at `https://huggingface.co/datasets/msrmanohar/telugu-prompt-style-recovery` and code at `https://github.com/msrmanohar/ACL-PRLLM`.

References

Premjith B, Jyothish Lal G, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Ratnavel Rajalakshmi, Rahul Ponnusamy, and Chinthala Bhuvanesh. 2026. Shared Task on Prompt Recovery for LLM in Telugu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

A. Bala. 2025. Fake news detection in Dravidian languages using MuRIL. Master's thesis, IIIT Hyderabad.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, and Anand Kumar M, editors. 2023. *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Daryna Dementieva, Nikolay Babakov, and Alexander Panchenko. 2023. Detecting text formality: A study of text classification approaches. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 274–284, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Divyansh Kakwani, Anoop Kunchukuttan, Satish Golla, Gowtham N.C., Pushpak Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. MuRIL: Multilingual representations for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 247–263.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 1 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.