Fine-Tuning BART for Spoken Utterance to Pictogram Sequence Generation in ImageCLEF 2026

Notebook for the ImageCLEF Lab at CLEF 2026

N.V. Gokul[1], Rohith S[1], Mirunalini P[1] and Balakumaran S[1]

[1]*Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India*

Abstract

People with speech and language impairments rely on augmentative and alternative communication (AAC) systems that use pictograms to express meaning. Manually selecting pictograms is slow and cognitively demanding, which motivates automating the translation from natural language to pictogram sequences. This paper presents our approach to the ImageCLEF 2026 Text-to-Picto task, where we fine-tune `facebook/bart-base` to map spoken English utterances to ordered pictogram token sequences. We remove train-validation overlaps and within-train duplicates to ensure a clean evaluation, extend the tokenizer with frequently split pictogram tokens (e.g., `half_body`), and resume training from a known checkpoint. Decoding is tuned by sweeping beam width, length penalty, and no-repeat n-gram size on the validation set. We evaluate using SacreBLEU, METEOR, and a task-specific edit-distance metric, PictoER, and our best submission ranks third on the official leaderboard with a BLEU of 52.21 and METEOR of 0.7104.

Keywords

ImageCLEF 2026, BART, pictogram generation, sequence-to-sequence, augmentative and alternative communication

1. Introduction

Communication is a fundamental human need, yet millions of people worldwide are unable to rely on natural speech due to conditions such as autism spectrum disorder, cerebral palsy, aphasia, or amyotrophic lateral sclerosis. For these individuals, augmentative and alternative communication (AAC) systems provide a vital bridge, allowing them to convey meaning by selecting pictograms — standardised graphic symbols each linked to a word or concept — arranged in sequences on a device. While AAC technology has advanced considerably, the process of composing a message by manually browsing and tapping through pictogram grids remains slow and places a heavy cognitive load on the user. Automating this process, so that a user can simply speak or type a phrase and have the device translate it into the right pictogram sequence, is therefore a practically important and clinically motivated goal.

The ImageCLEF 2026 evaluation campaign provides a collection of benchmark tasks for multimodal information retrieval and analysis [1]. Among these, the ImageCLEFtoPicto 2026 task formalises this challenge as a conditional sequence generation problem [2]. Given a short spoken English utterance, a model must predict the ordered sequence of pictogram tokens that an AAC device would display to convey the same meaning. This is harder than it might first appear. The target vocabulary consists of domain-specific identifiers such as `half_body`, `go_forward`, or `eat_meal` that do not appear in general text corpora and are fragmented into multiple subword pieces by standard tokenizers, making it difficult for a pretrained model to predict them as single coherent units. Beyond vocabulary, the mapping from natural language to pictograms involves genuine semantic compression: a seven-word utterance may reduce to just two or three pictogram tokens, requiring the model to learn which concepts carry communicative weight and which can be omitted. Standard sequence evaluation metrics also

CLEF 2026: Conference and Labs of the Evaluation Forum, September 21–24, 2026, Jena, Germany

gokul2470019@ssn.edu.in (N.V. Gokul); rohith2470055@ssn.edu.in (R. S); miruna@ssn.edu.in (M. P); balakumaran2470050@ssn.edu.in (B. S)

need careful adaptation, since each pictogram token must be treated as a single atomic symbol rather than decomposed into characters or subwords.

Transformer-based encoder-decoder models [3] have become the dominant approach for sequence-to-sequence generation since their introduction, and pretrained models such as BART [4] and T5 [5] have extended this to cover a wide range of generation tasks with strong transfer performance. We chose `facebook/bart-base` [4] as our backbone for three reasons: it requires no task prefix (unlike T5), it is comparatively light at approximately 139 million parameters versus 220 million for T5-base, and its denoising pretraining strategy — which corrupts input text with span masking and sentence permutation before training the decoder to reconstruct the original — produces particularly faithful discrete-token outputs. To address the tokenizer fragmentation problem, we add frequently occurring pictogram tokens as whole vocabulary entries and resize the model embeddings accordingly, following the domain-adaptive pretraining approach of Gururangan et al. [6].

Our full pipeline covers dataset cleaning with explicit leakage prevention, vocabulary-augmented tokenization, fine-tuning with `Seq2SeqTrainer` resumed from a strong checkpoint, and a sweep over decoding hyperparameters on the validation set. On the official leaderboard our best run achieves a BLEU of 52.21, METEOR of 0.7104, and WER of 0.3212, ranking third among all submitted systems and within 0.34 BLEU points of the top entry.

2. Related Work

The modern approach to conditional text generation is built on transformer encoder-decoder architectures [3], which use self-attention to model long-range dependencies across both input and output sequences, replacing the bottleneck of earlier RNN-based models. BART [4] operationalises this for generation by pretraining with span masking and sentence permutation, training the decoder to reconstruct clean text from corrupted input and giving it strong priors for producing well-formed discrete sequences; T5 [5] takes a complementary route by casting every NLP task as text-to-text with task prefixes, though BART's prefix-free design and reconstruction focus make it more natural for structured token output. Translating natural language into symbolic representations has a longer lineage: early systems such as that of Sevens et al. [7] relied on syntactic parsing and hand-crafted lexicons to map constituents to pictogram entries, but these rule-based pipelines struggled to generalise and required heavy manual maintenance; the ImageCLEF Text-to-Picto task series [1] has since become the primary benchmark driving the shift to data-driven sequence generation for AAC. A practical obstacle shared by all neural approaches is subword fragmentation, where domain-specific compound identifiers in the pictogram vocabulary are split into multiple pieces by general tokenizers; Gururangan et al. [6] showed that vocabulary augmentation and domain-adaptive pretraining consistently mitigate this, which motivates our tokenizer extension step. Evaluation uses three complementary metrics: SacreBLEU [8], a reproducible implementation of BLEU [9] applied with `tokenize=none` to treat each pictogram as an atomic unit; METEOR [10], which adds recall and unigram alignment to capture partial matches; and PictoER, the primary task metric, which computes a token-level edit-distance error rate counting substitutions, deletions, and insertions directly over the pictogram sequence.

3. Methodology

3.1. Dataset Description

The ImageCLEF 2026 Text-to-Picto corpus is provided in three splits. Table 1 summarises the utterance counts across splits. Each record contains a source utterance (`src`), a target pictogram token sequence (`tgt`), and an ordered list of ARASAAC pictogram IDs (`pictos`); the test split provides only `src`.

Table 1
Utterance counts for the ImageCLEF 2026 Text-to-Picto corpus.

Split	Utterances
Train	59,278
Validation	4,397
Test	4,306

Table 2
JSON field descriptions for each dataset record.

Field	Type	Description
`src`	*string*	Natural-language spoken utterance in English.
`tgt`	*string*	Space-separated sequence of pictogram tokens representing the AAC-device output.
`pictos`	*int[]*	Ordered list of ARASAAC symbol IDs, one per target token. Absent in the test split.

Table 3
A representative example from the training set.

Field	Value
`src`	*"the seat of the county is watertown"*
`tgt`	*"the seat of be city"*
`pictos`	[8476, 36505, 7074, 36480, 2704]

3.2. Data Preparation and Preprocessing

The proposed pipeline starts with the structured ingestion of the ImageCLEF Text-to-Picto dataset, which is provided in three JSON files: train.json, valid.json and test.json. The records contain a unique identifier (id), a natural language source sentence (src), and a target pictogram token sequence (tgt), with the exception of the test set, which has no targets. The task is presented as a sequence-to-sequence mapping task, in which the text is converted into a sequence of pictograms that are semantically aligned with the text.

A thorough data cleaning and decontamination procedure is performed before the model training to guarantee the integrity of the experiments. Extraneous whitespace is removed from all textual fields and token boundaries are standardized. This is necessary to ensure that duplicate detection and tokenization are consistent. The cross-split leakage removal procedure is used to avoid evaluation bias: any (src, tgt) pair that is present in both the training and validation sets is removed from the training set. This ensures that the validation metrics measure real generalization and not memorization.

Furthermore, intra-training deduplication is conducted by recognizing and eliminating duplicate examples from the training data. This prevents implicit weighting of duplicated samples in the optimization process and provides a more uniform learning signal. The resulting data set is thus decontaminated and distributionally balanced, providing a good basis for downstream modeling.

3.3. Tokenization and Representation Learning

The system uses the BART tokenizer, which is based on byte-pair encoding (BPE), and is initialised from the pretrained facebook/bart-base checkpoint or from a previously fine-tuned model directory in continuation settings [4]. BPE is good for general natural language, but when used for domain-specific pictogram tokens, fragmentation occurs because they are frequently compound identifiers like `half_body` or `action_run`. To overcome this, a targeted vocabulary augmentation strategy is presented.

In particular, the tokens of the training targets are processed to find high-frequency candidates (those that appear at least five times) that are broken up into multiple subword units by the default tokenizer.

These tokens are stored as atomic units in the tokenizer vocabulary, and will be retained when encoding and decoding. This controlled vocabulary expansion not only enhances generation fidelity but also avoids unnecessary expansion of embedding size. The extension is only used for fresh training runs, and is compatible with existing checkpoints for continuation. After the tokenizer is configured, the dataset is converted to representations that are compatible with the model. The source sequences are tokenized up to 64 tokens and the target sequences are tokenized up to 16 tokens, based on empirical statistics of the dataset. A batched mapping approach is used to optimize throughput for tokenization. Dynamic padding is applied at batch time using a data collator, avoiding unnecessary computation, rather than static padding.

To ensure stability during training, padding tokens in target sequences are replaced with a sentinel value (-100) that is ignored during the computation of cross-entropy loss. This masking mechanism ensures that gradient updates are not affected by any padding tokens. At the end of this stage, the result is a collection of efficient encoded input-output pairs that can be used for sequence-to-sequence learning.

3.4. Model Training and optimization

The transformer-based encoder–decoder architecture is the backbone of the system, which is built on BART-base [4]. This model is composed of 6-layer bidirectional encoder and 6-layer autoregressive decoder, having a hidden dimensionality of 768. Its denoising pretraining objective and contextual representation ability make it suitable for conditional text generation tasks.

In the case of domain-specific tokens, the embedding matrices are resized to fit the larger vocabulary. Embeddings are randomly initialized and then fine-tuned during training when they are added. Additional regularization is added by raising the dropout rates from 0.1 to 0.15 for continuation runs, where training is continued from an intermediate point. This modification helps mitigate overfitting and enables the model to escape local performance plateaus observed in earlier training stages. The HuggingFace Seq2SeqTrainer framework is used for training [11], offering efficient optimization, evaluation, and checkpointing. Two learning rate regimes are used: a higher learning rate (3×10^{-5}) for training from scratch and a lower learning rate (5×10^{-6}) for fine-tuning an already adapted model, which guarantees stable convergence when fine-tuning an already adapted model. There is no warmup schedule for continuation because the model has been initially stabilized.

Evaluation is done at the end of every epoch with beam search decoding with beam width 16 and max generation length 16 tokens. The main evaluation metric is SacreBLEU with tokenization disabled, which is consistent with pre-tokenized targets. Model checkpoints are stored at regular intervals, and only the best models are kept to reduce storage requirements. To avoid over training, early stopping with a patience of 2 epochs can be optionally enabled during exploratory experiments. Continuation learning is an important part of the training process, where the system is able to continue from a saved checkpoint. The number of epochs completed is restored from the trainer state file, so that the pipeline can calculate and run the remaining epochs needed to complete a specified number. This will allow efficient use of computational resources, and ensure continuity of training.

3.5. Model Architecture

Our system is built on `facebook/bart-base` [4], a bidirectional and auto-regressive transformer [3] pre-trained with a denoising sequence-to-sequence objective. The encoder processes the full source utterance bidirectionally, producing contextualised representations for every input token. The autoregressive decoder then generates the target pictogram token sequence one symbol at a time, attending to the encoder output at every step via cross-attention. This prefix-free design makes BART particularly well-suited to structured discrete-token generation, where the output vocabulary is closed and domain-specific.

Architecture overview. The base model contains six encoder layers and six decoder layers, each equipped with 16 attention heads and a hidden dimensionality of 768. Token embeddings are shared

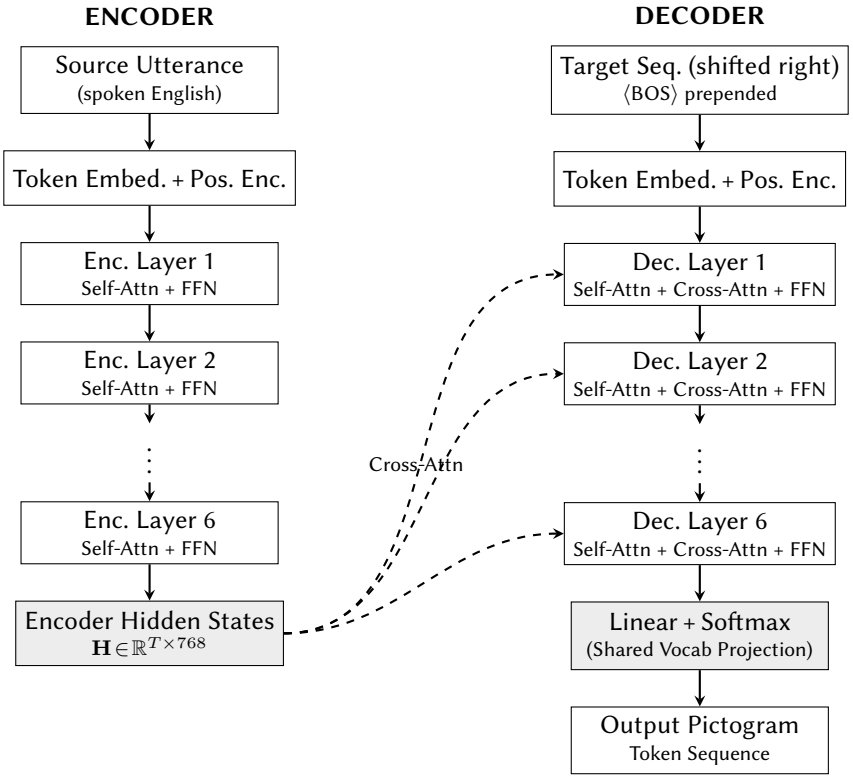

Figure 1: BART-base model architecture for Text-to-Picto generation. Solid arrows denote the forward data flow within the encoder and decoder stacks. Dashed arrows represent cross-attention connections, through which every decoder layer attends to the full set of encoder hidden states $\mathbf{H}$.

among the encoder input projection, the decoder input projection, and the output linear head, so that the entire model amounts to approximately 139 M parameters. This parameter budget makes full fine-tuning feasible on a single consumer-grade GPU within a practical time budget.

Vocabulary augmentation. A central challenge in this task is *subword fragmentation*: domain-specific pictogram identifiers such as `half_body`, `go_forward`, and `eat_meal` are decomposed into multiple byte-pair encoding (BPE) sub-pieces by the default BART tokenizer, increasing effective sequence length and reducing generation fidelity. Following the domain-adaptive pre-training approach of Gururangan et al. [6], we extend the tokenizer vocabulary with high-frequency target tokens that are fragmented under the default encoding. Concretely, any token that (i) appears at least five times in the training targets and (ii) requires more than one BPE piece under the vanilla tokenizer is added as an atomic vocabulary entry. After augmentation, the model's embedding matrices are resized accordingly; newly added embeddings are randomly initialised and learned from scratch during fine-tuning.

Figure 1 illustrates the overall model structure.

4. Results

Our best run (ID 1047, submitted 2026-05-06) achieves a BLEU score of 52.21, a METEOR score of 0.7104, and a Word Error Rate (WER) of 0.3212. This result is obtained by fine-tuning the BART-base model with vocabulary augmentation, continued training from checkpoint-44112, and a tuned decoding configuration (beam width 16, maximum generation length 16 tokens).

Table 4 presents the official results of our best submission on the ImageCLEF 2026 Text-to-Picto leaderboard.

The BLEU score of 52.21 indicates that the generated pictogram sequences exhibit substantial lexical overlap with the reference targets. The METEOR score of 0.7104, which accounts for synonyms and word order, further confirms the semantic alignment between predictions and ground truth. The WER of

Table 4
Official leaderboard results for our best submission (participant: `goku1_n_v`).

Metric	Score
Run ID	1047 (best)
WER	0.3212
BLEU	52.21
METEOR	0.7104

0.3212 reflects the proportion of token-level errors, suggesting that approximately 68% of the predicted pictogram tokens are correct. These results validate that the proposed pipeline successfully addresses the text-to-pictogram generation task.

5. Conclusion

In this paper, we presented a sequence-to-sequence approach for the ImageCLEF 2026 Text-to-Picto task, fine-tuning BART-base to translate spoken English utterances into pictogram sequences for augmentative and alternative communication systems. Our methodology focused on dataset decontamination to eliminate leakage, vocabulary augmentation to address subword fragmentation of domain-specific tokens, and continued training with tuned decoding hyperparameters. Our best submission achieved a BLEU score of 52.21 and a METEOR score of 0.7104, ranking third on the official leaderboard. Future work will explore reinforcement learning to optimise directly for PictoER and multimodal extensions that incorporate pictogram images as auxiliary supervision during fine-tuning.

Declaration of AI Use

During the preparation of this work, we used OpenAI's ChatGPT model for language fluency improvement and technical documentation assistance. All experimental work, data analysis, results, and scientific conclusions are our own. We have reviewed and refined all AI-assisted content and take full responsibility for the published work.

References

[1] M. Hjuler, D. Fabre, C. Lemaire, B. Lecouteux, y. . . m. . S. a. . J. b. . C. e. . E. Didier Schwab, series = Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Overview of the 2026 imagecleftopicto task: Investigating the generation of pictogram sequences from english text dataset, ????

[2] B. Ionescu, H. Müller, D. Stanciu, A. Radu, R. Bolborici, M. Negru, A. Ene, V. Vasilescu, A.-A. Nicolae, L. Ştefan, M. Constantin, M. Dogariu, A. Andrei, H. Damm, T. M. G. Pakull, A. Ben Abacha, A. García Seco de Herrera, C. M. Friedrich, R. Brüngel, L. Reinartz, H. Schäfer, C. S. Schmidt, B. Bracke, P. Nath, B. Eryılmaz, M. Hjuler, D. Fabre, C. Lemaire, B. Lecouteux, D. Schwab, D. Dimitrov, M. S. Hee, M. Ahsan, S. Ahmad, D. Zlatkova, G. Pachov, Z. Xie, P. Nakov, I. Koychev, J. E. Heras Rivera, D. K. Low, W. Yim, J. Ruzevick, D. Child, M. Kurt, Z. Sun, F. Xia, M. Yetisgen, A. Radzhabov, Y. Prokopchuk, V. Kovalev, S. A. Hicks, S. Gautam, M. A. Riegler, V. Thambawita, P. Halvorsen, M. El Sakka, J. Mothe, A. Băicoianu, C. Florea, M. Ivanovici, Overview of imageclef 2026: Multimodal challenges in medicine, science, agritech, and security, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer Lecture Notes in Computer Science LNCS, Jena, Germany, 2026.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin,

Attention is all you need, in: Advances in Neural Information Processing Systems (NeurIPS), volume 30, 2017.

[4] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 7871–7880.

[5] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67.

[6] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. A. Smith, Don't stop pretraining: Adapt language models to domains and tasks, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 8342–8360.

[7] L. Sevens, V. Vandeghen, I. Schuurman, W. Daelemans, F. Van Eynde, Translating text into pictographs, Natural Language Engineering 24 (2018) 189–214.

[8] M. Post, A call for clarity in reporting bleu scores, in: Proceedings of the Third Conference on Machine Translation (WMT), 2018, pp. 186–191.

[9] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: A method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, pp. 311–318.

[10] S. Banerjee, A. Lavie, Meteor: An automatic metric for mt evaluation with improved correlation with human judgments, in: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005, pp. 65–72.

[11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Huggingface's transformers: State-of-the-art natural language processing, 2020. arXiv:1910.03771.