Hiwiy at ImageCLEFtoPicto 2026: Text-to-Picto Translation and Next-Pictogram Prediction

Notebook for the ImageCLEF Lab at CLEF 2026

Jiancheng Huang[1], Xueyang Qin[1] and Zhongyuan Han[1,*]

[1]Foshan University, Foshan, China

Abstract

ImageCLEFtoPicto 2026 evaluates assistive systems that convert English utterances into ARASAAC pictogram terms and predict the next pictogram while a user is composing a pictogram sequence. For this evaluation task, we propose a text-to-pictogram and sequence-prediction framework that combines a fine-tuned text-to-text Transformer with several next-pictogram models. The Text-to-Picto component uses `google-t5/t5-large` to generate pictogram-term sequences from source utterances, while the Next-Pictogram component models partial pictogram histories through text-only decoding, pictogram-ID modeling, dual-modal token–ID fusion, pretrained GPT-2 decoding, and SASRec-style sequential recommendation with text fusion. On the official evaluation data, our best submitted runs achieved rank 8/23 for Text-to-Picto and rank 3/29 for Next-Pictogram Prediction. The results indicate that combining lexical pictogram tokens with discrete pictogram-ID histories is effective for next-pictogram ranking.

Keywords

ImageCLEFtoPicto, text-to-pictogram translation, next pictogram prediction, sequential recommendation

1. Introduction

ImageCLEFtoPicto focuses on Augmentative and Alternative Communication (AAC) scenarios in which spoken or written language must be made accessible through pictograms [1, 2]. The 2026 track contains two subtasks. Text-to-Picto requires a system to translate an English utterance into a sequence of pictogram terms, with each term being associated with one or more ARASAAC pictographic images. Next-Pictogram Prediction requires a system to predict and rank the most probable next pictogram keyword token given a partial pictogram sequence. Both subtasks are built around the same practical goal: supporting communication between verbal users and AAC users by producing relevant, understandable pictogram sequences.

A central challenge in this track is that pictogram generation is not only ordinary text generation. Each output token must correspond to an allowed pictogram term, while the sequence should also preserve the structure of pictogram-based communication. For next-pictogram prediction, purely lexical models can use the surface keyword context but ignore the discrete pictogram identities, whereas purely ID-based models capture item transitions but lose lexical information from the pictogram terms. This mismatch makes it difficult to align the semantic information carried by keyword tokens with the sequential information carried by pictogram IDs.

To address this issue, we participated in both subtasks with models that treat pictogram prediction as constrained sequence generation and sequence ranking. For Text-to-Picto, we fine-tuned `google-t5/t5-large` as a text-to-text translation model from English utterances to pictogram-term sequences. For Next-Pictogram Prediction, we compared text-only Transformer decoding, pictogram-ID prediction, dual-modal token–ID Transformer fusion, GPT-2-medium fine-tuning, and SASRec-based sequential recommendation with optional text fusion. This working note describes the submitted

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

*Corresponding author.

huanghiwiy@gmail.com (J. Huang); qinxueyang@snnu.edu.cn (X. Qin); hanzhongyuan@gmail.com (Z. Han)

0009-0001-6787-3338 (J. Huang); 0000-0003-2802-5608 (X. Qin); 0000-0001-9860-9872 (Z. Han)

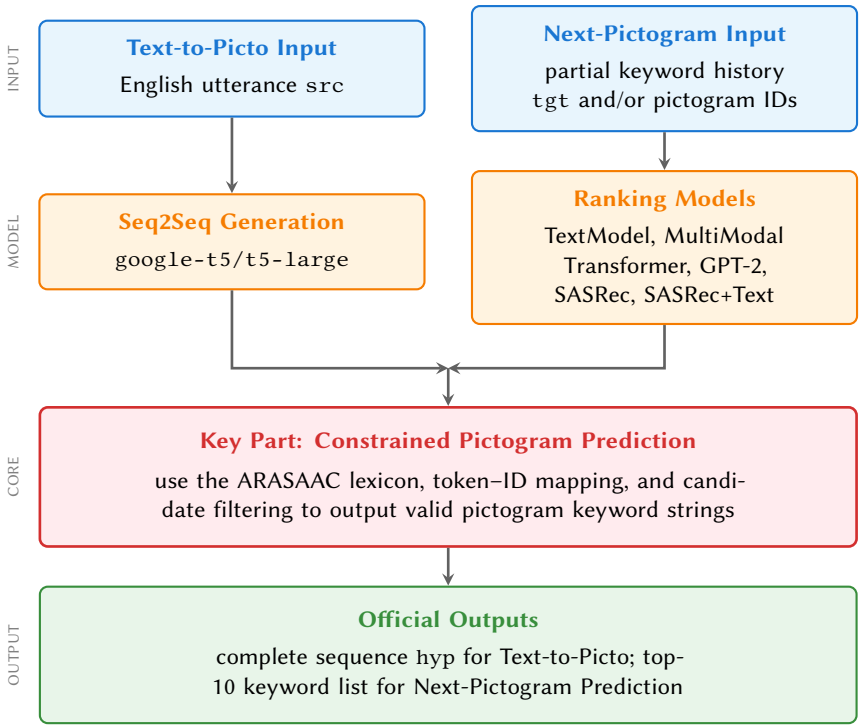

Figure 1: Overview of our methods for the two ImageCLEFtoPicto 2026 subtasks. The **constrained pictogram prediction** module (highlighted in red) marks the main shared constraint: model scores must be converted into valid pictogram keyword outputs.

systems, reports the official results, and discusses the gap between local validation behavior and the hidden test leaderboard.

2. Related Work

The Text-to-Picto subtask can be treated as a constrained translation problem, where the source language is English text and the target language is a sequence of pictogram terms. Transformer-based encoder–decoder models [3] and text-to-text pretraining such as T5 [4] are natural choices for this setting because they can model variable-length inputs and outputs.

The Next-Pictogram Prediction subtask is also closely related to autoregressive language modeling and sequential recommendation. Decoder-only language models, including GPT-style models [5], estimate the next token from previous context. Sequential recommenders such as SASRec [6] use self-attention over item histories and are well aligned with the task when each pictogram is viewed as an item in a user-composed sequence. Our experiments compare these perspectives within the same benchmark.

3. Method

Figure 1 summarizes our overall method. The two subtasks share the same pictogram lexicon and output space, but they use different input conditions. Text-to-Picto starts from an English utterance and generates a complete pictogram-term sequence. Next-Pictogram Prediction starts from a partial pictogram history and ranks the next keyword token. The main design choice in our systems is whether to represent the history as text tokens, pictogram IDs, or both.

3.1. Task Formalization

Text-to-Picto. For Text-to-Picto, each example is represented as (x_i, y_i, r_i). Here x_i is an English source utterance, $y_i = (w_{i,1}, \ldots, w_{i,m})$ is the target sequence of pictogram keyword tokens, and $r_i = (p_{i,1}, \ldots, p_{i,m})$ is the aligned sequence of ARASAAC pictogram IDs. The model learns a conditional generator f_θ that produces a hypothesis sequence $\hat{y}_i = f_\theta(x_i)$. The submitted JSON file contains the utterance ID and the generated hyp sequence.

Next-Pictogram Prediction. For Next-Pictogram Prediction, each instance contains a partial pictogram sequence. Given a prefix $c_i = (w_{i,1}, \ldots, w_{i,t})$ and, when available, the aligned ID prefix $q_i = (p_{i,1}, \ldots, p_{i,t})$, the system ranks candidate next pictogram keyword tokens. The goal is to output a list $L_i = (\hat{w}_{i,1}, \ldots, \hat{w}_{i,10})$ of up to 10 unique keyword strings, ordered from the most likely to the least likely next pictogram. During local validation, we form prefix–target pairs from the training data and evaluate whether the held-out next keyword appears in the ranked list. For official test prediction, the full provided prefix is used and each model produces a top-10 list after candidate filtering.

3.2. Models and Algorithms

T5 sequence-to-sequence model. For Text-to-Picto, we fine-tune the pre-trained `google-t5/t5-large` model. The source utterance is prefixed with the instruction `translate to pictogram:`, so the task is cast as text-to-text generation. The maximum source length is 128 tokens and the maximum target length is 64 tokens. Training uses AdamW with a linear warmup schedule. At inference time, we generate the pictogram-term sequence with beam search using beam size 5.

Text-only Transformer model. For Next-Pictogram Prediction, the TextModel variants use only the `tgt` keyword sequence and ignore pictogram IDs. The vocabulary is built from keyword tokens observed in the training data and test prefixes, together with `<pad>`, `<bos>`, `<eos>`, and `<unk>`. Each sequence is encoded as `<bos>` followed by the keyword tokens and `<eos>`, with maximum length 64. The model embeds tokens into a 256-dimensional space, adds positional information, and passes the sequence through a causal Transformer stack with 8 attention heads, feed-forward size 1024, GELU activation, dropout 0.1, and pre-layer normalization. The submitted `TextModel-l4` and `TextModel-l8` runs differ in the number of Transformer layers. The model is trained with next-token cross entropy using AdamW (`lr=3e-4`, batch size 128), and checkpoints are selected by local Top@1 accuracy. At inference time, logits from the final prefix position are ranked and filtered by the lexicon-derived allowed keyword set.

Dual-modal Transformer model. The MultiModal Transformer uses both the keyword sequence and the corresponding pictogram-ID sequence. The keyword vocabulary is derived from lexicon `sense_keys`, while pictogram IDs are embedded through a separate ID embedding table. For each sample, the word stream is encoded with `<bos>` and `<eos>` markers, and the ID stream uses 0 as the special boundary value while keeping the original pictogram IDs elsewhere. The word embedding and ID embedding are summed, positional encoding is added, and the fused representation is processed by a causal Transformer encoder. Each encoder layer uses 256 hidden units, 8 attention heads, feed-forward size 1024, GELU activation, dropout 0.1, and pre-layer normalization. The output head predicts keyword vocabulary IDs. During validation and inference, scores outside the allowed keyword set are masked before the top-10 candidates are selected. The submitted `MultiModal-l4`, `MultiModal-l8`, and `MultiModal-l24` runs correspond to 4-, 8-, and 24-layer versions.

GPT-2-medium decoder. We also fine-tune Hugging Face's `gpt2-medium` as a pretrained causal language model for next-keyword prediction. The tokenizer padding token is set to the end-of-sequence token, and each `tgt` sequence is truncated to 64 tokens. For each training batch, the prefix tokens are used as input and the shifted next tokens are used as labels. The model is optimized with AdamW using `lr=3e-5`, batch size 8, and up to 10 epochs. At inference time, the model scores the next token at the final context position. The top 100 token IDs are decoded, normalized, deduplicated, and filtered by the allowed keyword set until 10 candidates are collected.

SASRec sequential recommender. We reproduce the SASRec-style self-attentive sequential rec-

Table 1
Original dataset statistics for Text-to-Picto.

Split	Proportion	Samples	Notes
Training	87.2%	59,278	Used for model training
Validation	6.5%	4,397	Used for local validation and model selection
Testing	6.3%	4,306	Used for official evaluation

Table 2
Original dataset statistics for Next-Pictogram Prediction.

Dataset	Proportion	Samples
Training	97.0%	63,675
Testing	3.0%	1,986

ommendation model [6] by treating each pictogram ID as an item. The preprocessing first builds an ID-to-keyword mapping from training pairs, with lexicon `sense_keys` and normalized lemmas as fallbacks. Training data is expanded into prefix–target pairs: for each pictogram sequence, every position after the first is treated as a target and the preceding IDs form the history. Histories are truncated or padded to `max_len=16`. The model contains item embeddings, learned positional embeddings, dropout 0.2, two causal self-attention blocks, and final layer normalization. The last valid hidden state is scored against all item embeddings, and cross entropy is optimized with AdamW (`lr=3e-4`, weight decay `1e-5`) and gradient clipping at 1.0. For official output, predicted pictogram IDs are converted back to keyword strings. When one ID corresponds to multiple keywords, we choose the most frequent keyword observed for that ID in the training data as the representative output.

SASRec with text fusion. SASRec+Text keeps the same ID-based next-item prediction objective but augments each ID history with the aligned keyword-token history. A separate text vocabulary is built from training `tgt` tokens. Before the attention stack, the model combines scaled item embeddings, weighted text embeddings, and positional embeddings: item_embedding(seq) $\times$ item_scale $+ \alpha \times$ text_embedding(text_seq) $+$ pos_embedding. Both α and item_scale can be learned. The output head remains item-based: the last hidden state is multiplied by the item embedding matrix to score next pictogram IDs. Validation reports both ID-ranking metrics, such as HR, NDCG, and MRR, and token-level Top@N after converting IDs to keyword candidates.

4. Experiment

4.1. Experimental Setup

4.1.1. Experimental Dataset

The dataset for Text-to-Picto is shown in Table 1, containing 59,278 instances, with a validation set of 4,397 instances and a test set of 4,306 instances. The training set accounts for 87.2%, the validation set for 6.5%, and the test set for 6.3%.

The training set for Next-Pictogram Prediction is shown in Table 2, containing 63,675 instances, while the test set contains 1,986 instances. The training set accounts for 97.0%, and the test set for 3.0%.

4.1.2. Data Processing

For Text-to-Picto, we directly use the provided training, validation, and test sets without additional processing.

For Next-Pictogram Prediction, we further split the original training set into a new training set and a local validation set, with proportions of 90% and 10%, respectively. The validation set is composed of

Table 3
Data split statistics after partitioning the training set.

Split	Proportion	Samples	Notes
Training	87.3%	57,308	Used for model training
Validation	9.7%	6,367	Used for local validation
Testing	3.0%	1,986	Used for official evaluation

Table 4
Official leaderboard result for Text-to-Picto.

Model	Rank	PER ↓	BLEU ↑	METEOR ↑
google-t5/t5-large	8/23	0.3333	50.3802	0.6967

the first 10% of samples extracted from the training set, while the test set remains unchanged for easy submission and evaluation. The statistics of the split dataset are presented in Table 3.

4.1.3. Evaluation Metrics

Text-to-Picto is evaluated with BLEU(sacreBLEU, hereinafter referred to as BLEU), METEOR, and PER(Picto-term Error Rate). BLEU measures the number of common n-grams between the translation hypothesis and the reference translation. METEOR performs an alignment between the translation hypothesis and the reference translations, going beyond simple word matching. It takes into account not only direct matches but also matches based on synonyms, morphological variations, and paraphrases. This evaluation captures additional semantic information that is not encoded in the BLEU score. PER is a metric derived from WER. Instead of evaluating the number of errors at the word level, it focuses on the number of errors of tokens, each linked to an ARASAAC pictogram. For PER, lower is better; for BLEU and METEOR, higher is better.

Next-Pictogram Prediction is evaluated with Top@N Accuracy and semantic similarity. Top-N Accuracy (N = 1, 3, 5, 10) The fraction of instances where the ground-truth token appears among the top-N candidates. Reported for all four values of N. Top-1 Accuracy is the primary ranking metric. Semantic Similarity Mean cosine similarity between the sentence embedding of the top-1 predicted token and the ground-truth token, computed using `all-MiniLM-L6-v2`. This gives partial credit for semantically close predictions.

4.2. Experimental Results and Analysis

The submission results of model t5 are shown in Table 4. Our model achieved the 8th place in the official ranking, indicating that pre-trained language models have certain advantages in the text-to-pictogram translation task. Among them, the BLEU and METEOR metrics performed well, reaching 50.3802 and 0.6967 respectively, both surpassing the good level thresholds of 30 and 0.4 stated in the official documentation.

The official results submitted by all models are shown in Table 5. Our best model is the 24-layer multimodal encoding model, which ranked 3rd out of 29 submissions. Among them, the 24-layer multimodal encoding model improved the top-1 metric by 2.63% compared to the 8-layer multimodal encoding model, which in turn improved by 9.86% compared to the 4-layer model, suggesting that increasing model depth can improve performance. Only text-only models performed poorly, and both encoding and decoding models showed suboptimal performance, indicating that relying solely on textual information may not be sufficient for accurately predicting the next pictogram. And the dual-modal Transformer encoder model using both text and pictogram IDs performs better than the Transformer decoder model using only text, indicating that pictogram ID information is helpful for predicting the next pictogram.

Table 5
Official leaderboard results for included Next-Pictogram Prediction runs, ranked according to the scores of top@1 from high to low.

Model	Rank	Top@1	Top@3	Top@5	Top@10	SemSim
MultiModal-l24	**3/29**	**0.0780**	**0.1239**	**0.1415**	**0.1697**	**0.3337**
MultiModal-l8	6/29	0.0760	0.1168	0.1324	0.1586	0.3320
MultiModal-l4	9/29	0.0710	0.1133	0.1365	0.1692	0.3296
SASRec+Text	26/29	0.0227	0.0599	0.0831	0.1284	0.2914
TextModel-l8	27/29	0.0201	0.0418	0.0569	0.0866	0.2734
TextModel-l4	28/29	0.0176	0.0408	0.0549	0.0725	0.2664
GPT2-medium	29/29	0.0171	0.0433	0.0624	0.0967	0.2765

Table 6
Local validation records from Next-Pictogram Prediction run folders, ranked according to the scores of top@1 from high to low.

Model	Top@1	Top@3	Top@5	Top@10	SemSim
MultiModal-l24	0.2052	0.3134	0.3470	0.3948	0.4527
MultiModal-l8	0.2046	0.3180	0.3553	0.4081	0.4509
MultiModal-l4	0.2030	0.3017	0.3414	0.3896	0.4420
TextModel-l8	0.1482	0.2245	0.2471	0.2803	0.3792
TextModel-l4	0.1464	0.2378	0.2664	0.3069	0.3834
SASRec+Text	0.1465	0.2220	0.2550	0.2962	—
SASRec	0.1319	0.2103	0.2427	0.2844	—
GPT2-medium	0.0871	0.1505	0.1832	0.2288	0.3344

Table 6 lists the local validation records stored in the run folders. These validation scores are not directly comparable to official test scores, but they explain the model selection process. Similarly, the 24-layer multimodal encoding model outperformed the 8-layer and 4-layer variants on the local validation set, which is consistent with the official test ranking. There is a significant gap between the official test results and the local validation results, indicating that the model may be overfitting on the training set or that the test set distribution differs from the training set, necessitating further improvements in the model's generalization capabilities and validation strategies.

Among these models, only SASRec and SASRec+Text derive the probability of pictogram-ID based on sequence recommendation predictions, while the other models predict keywords based on text tokens. Therefore, in our local validation, we not only computed the official metrics Top@N for SASRec and SASRec+Text, but also computed the HR@N metrics for sequence recommendation, as shown in Table 7. The HR@N metric for sequence recommendation is calculated based on the predicted list of pictogram-IDs, while the Top@N metric is calculated based on the predicted list of keywords, requiring a mapping from pictogram-IDs to keyword strings. From the results, it can be seen that the HR@N metrics of SASRec and SASRec+Text are significantly higher than the Top@N metrics, and the mapping from pictogram-IDs to keyword strings leads to a performance degradation.

5. Conclusions

This paper presented our systems for the ImageCLEFtoPicto 2026 Text-to-Picto and Next-Pictogram Prediction subtasks. For Text-to-Picto, we fine-tuned `google-t5/t5-large` to translate English utterances into pictogram-term sequences. For Next-Pictogram Prediction, we compared several sequence modeling strategies, including text-only Transformer models, dual-modal token–ID Transformer models, GPT-2-medium, SASRec, and SASRec with text fusion.

Our best official results ranked 8/23 in Text-to-Picto and 3/29 in Next-Pictogram Prediction. The results show that combining keyword tokens with pictogram-ID information is useful for next-pictogram

Table 7
Local validation records of the SASRec model.

Evaluation metric	SASRec	SASRec+Text
Top@1	0.1319	0.1465
HR@1	0.1849	0.2071
Top@3	0.2103	0.2220
HR@3	0.2891	0.3067
Top@5	0.2427	0.2550
HR@5	0.3325	0.3443
Top@10	0.2844	0.2962
HR@10	0.3875	0.4011

ranking, especially in the 24-layer dual-modal Transformer model. At the same time, the gap between local validation and official test performance suggests that future work should focus on stronger generalization, more robust validation splits, and better conversion between pictogram IDs and keyword outputs.

Acknowledgments

This work is supported by the Guangdong Province Basic and Applied Basic Research Fund (Guangdong-Foshan Joint Funds, 2024A1515140026)

Declaration on Generative AI

During the preparation of this work, we used GPT-5.5 in order to: Translation and polishing, and used Kimi-k2.6 in order to: Color and beautify the structural diagram. After using these tool, we reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. Hjuler, D. Fabre, C. Lemaire, B. Lecouteux, D. Schwab, Overview of the 2026 imagecleftopicto task: Investigating the generation of pictogram sequences from english text dataset, in: E. S. Salido, A. Barron-Cedeno, A. Garcia Seco de Herrera, S. MacAvaney, J. M. Struss (Eds.), CLEF 2026 Working Notes, Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Jena, Germany, 2026.

[2] B. Ionescu, H. Muller, D.-C. Stanciu, A. Radu, R.-G. Bolborici, M. Negru, A.-F. Ene, V.-M. Vasilescu, A.-A. Nicolae, L.-D. Stefan, M.-G. Constantin, M. Dogariu, A.-G. Andrei, H. Damm, T. M. G. Pakull, et al., 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, Jena, Germany, 2026.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017.

[4] 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.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI technical report, 2019.

[6] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: Proceedings of the 2018 IEEE International Conference on Data Mining, 2018, pp. 197–206.