T5 Fine-tuning for Text-to-Picto and Interpolated N-gram Modeling for Next Pictogram Prediction

Notebook for the ImageCLEF Lab at CLEF 2026

Kaiyi Li[1], Zhongyuan Han[1,*] and Fuchuan Ye[1]

[1]Foshan University, Foshan, China

Abstract

The ImageCLEFtoPicto 2026 task focuses on augmentative and alternative communication (AAC) and includes two sub-tasks: translating English text into pictogram term sequences and predicting the next pictogram term in a given sequence. This paper presents two methods for these sub-tasks. For the Text-to-Picto sub-task, we formulate it as a text-to-text generation problem and fine-tune a pre-trained T5-small model to transform English source sentences into pictogram term sequences. For the Next Pictogram Prediction sub-task, we formulate it as a next pictogram token prediction problem and adopt a fixed interpolation smoothed n-gram method to rank candidate tokens.On the evaluation data of the first task, the fine-tuned T5-small model achieves a sacreBLEU score of 51.34, a METEOR score of 70.71, and a PictoER score of 32.56 on the validation set, significantly outperforming the direct copy baseline. On the evaluation data of the second task, the n-gram interpolation model achieves a Top-1 accuracy of 0.193341 and a Top-10 accuracy of 0.395791 in the Next Pictogram Prediction task.

Keywords

T5-small, n-gram, pictogram, AAC, CLEF

1. Introduction

ImageCLEFtoPicto 2026 is organized as part of the ImageCLEF 2026 lab at CLEF 2026 [1]. The task focuses on augmentative and alternative communication (AAC) systems and aims to help people with difficulties in language expression or comprehension communicate through pictograms using automatic methods. In AAC systems, pictograms can represent words, phrases, or concrete concepts. ImageCLEFtoPicto 2026 includes two sub-tasks: Text-to-Picto and Next Pictogram Prediction [2].

Automatically generating pictogram sequences and predicting the next pictogram can reduce the communication effort required from AAC users. However, the two sub-tasks involve different modeling challenges. Text-to-Picto generation requires mapping natural language expressions to symbolic pictogram terms, while Next Pictogram Prediction requires modeling local dependencies within partially constructed pictogram sequences. Compared with ordinary word sequences, pictogram term sequences are more symbolic and may contain task-specific collocation patterns [3].

2. Related Work

ImageCLEFtoPicto 2025 was described in the task overview paper [4]. Hjuler and Fabre proposed a T5-based method for French Text-to-Picto and Speech-to-Picto translation [5]. They studied the performance of a Transformer-based approach for French Text-to-Picto and Speech-to-Picto translation, where a pre-trained Google T5 model was fine-tuned on the provided dataset. To solve the Speech-to-Picto task, this model was combined with a pre-trained ASR model and gave promising results [5].

Similarly, in ImageCLEFtoPicto 2024, there was also a Text-to-Picto translation task for converting French text into corresponding pictogram terms. Koushik et al. demonstrated the effectiveness of a

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

*Corresponding author.

liky8502@gmail.com (K. Li); hanzhongyuan@gmail.com (Z. Han); yfc2097639788@gmail.com (F. Ye)

0009-0006-2500-1036 (K. Li); 0000-0001-8960-9872 (Z. Han); 0009-0002-1467-0358 (F. Ye)

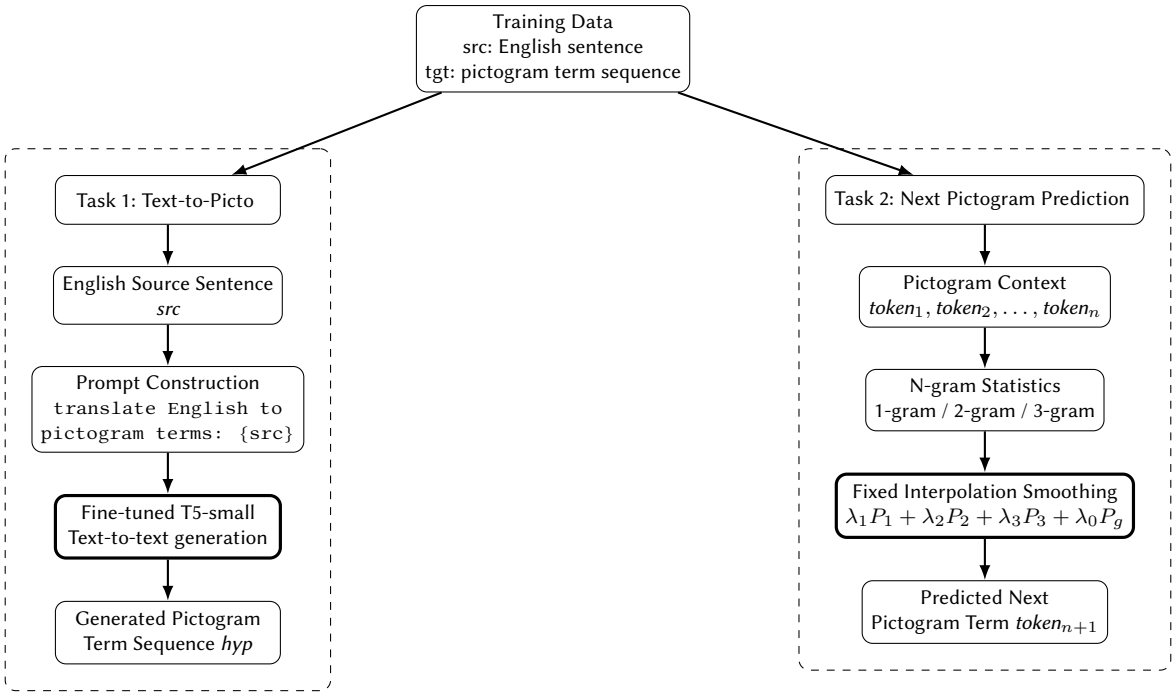

Figure 1: Overall framework of the proposed method.

transformer-based model in translating French text into meaningful pictogram sentences. The proposed model achieved a PictoER score of 13.9, BLEU score of 74.3 and a METEOR score 87.0 [6].

Compared with previous ImageCLEFtoPicto editions, the 2026 task continues to include the Text-to-Picto sub-task, but the source language shifts from French to English [4, 2]. Moreover, the second sub-task changes from Speech-to-Picto to Next Pictogram Prediction.

3. Method

3.1. Overall Framework

The goal of the first sub-task is to generate the corresponding pictogram term sequence *hyp* given an English source sentence *src*. Although the task name contains "Picto", this sub-task does not directly generate images. Instead, it generates a textual term sequence that can be mapped to ARASAAC pictogram resources. In this paper, we formulate this sub-task as a text-to-text sequence generation problem and fine-tune a pre-trained T5-small model for this purpose.

The goal of the second sub-task is to predict the next most likely pictogram term according to the pictogram keyword sequence that has already been generated. Different from a complete sentence generation task, the prediction target of this sub-task is a discrete pictogram keyword token. Since pictogram sequences contain strong local collocation patterns, we adopt an n-gram-based statistical language model and combine it with fixed interpolation smoothing to integrate contextual information from different n-gram orders [7].

3.2. Text-to-Picto Method

3.2.1. Task Modeling

The input of the Text-to-Picto sub-task is an English source sentence *src*, and the output is a target pictogram term sequence *tgt*. We formulate this sub-task as a text-to-text sequence generation problem:

$$src \rightarrow tgt$$

For example:

Input: `this version burst onto the internet in an unofficial way`

Output: `past the make above internet by no`

This task can be regarded as a special form of machine translation, where ordinary English text is translated into pictogram-oriented textual representations.

3.2.2. Copy Baseline

To verify whether the proposed model-based method is effective, we first construct a simple copy baseline. This baseline does not use any trained model. Instead, it directly takes the input sentence *src* as the predicted result *hyp*:

$$hyp = src$$

This method serves as the simplest control experiment and is used to determine whether the fine-tuned model has truly learned the transformation rules from English text to pictogram term sequences.

3.2.3. T5-small Fine-tuning Method

T5 is a text-to-text generation model, which is suitable for tasks where both the input and the output are represented as text. Therefore, we adopt the pre-trained T5-small model as the base model and fine-tune it on `train.json`.

During data construction, a task-specific prompt is added before each input sentence:

```
translate English to pictogram terms:
```

The final input format is:

```
translate English to pictogram terms: {src}

The target output is the corresponding pictogram term sequence: {tgt}
```

For example:

Input: `translate English to pictogram terms: She is going to school tomorrow`

Output: `she go school tomorrow`

In this way, the model can learn the mapping relationship from English source sentences to pictogram term sequences.

3.3. Next Pictogram Prediction Method

3.3.1. Task Modeling

The goal of the Next Pictogram Prediction task is to predict the next most likely pictogram term according to the existing pictogram keyword sequence. Given a context sequence:

$$context = token_1, token_2, \ldots, token_n$$

the model needs to predict:

$$token_{n+1}$$

Different from the complete sequence generation setting in Text-to-Picto, this task is closer to the next-token prediction problem in language modeling. Since pictogram keyword sequences contain strong local ordering patterns and fixed collocations, we adopt an n-gram statistical language model for this task.

Table 1
Duplicate sample checking between the training and validation sets.

Item	Number
Training samples	59278
Validation samples	4397
Duplicate samples between train and valid	0

3.3.2. N-gram Statistical Modeling

We first count the occurrence frequencies of following tokens under different n-gram contexts based on the *tgt* sequences in the training set. For a given context and a candidate token, the conditional probability is defined as:

$$P(token \mid context) = \frac{count(context, token)}{count(context)}$$

where $count(context, token)$ denotes the number of times that the token appears after the given context, and $count(context)$ denotes the total number of occurrences of this context in the training set.

For example, if the following sequences appear multiple times in the training set:

```
I want drink water
I want drink juice
I want drink water
```

then after the context `I want drink`, the token `water` appears more frequently than `juice`. Therefore, the model tends to predict `water`.

4. Experimental Setup

4.1. Dataset

This experiment uses the official dataset provided by ImageCLEFtoPicto 2026. For the Text-to-Picto task, the official data mainly includes three files: `train.json`, `valid.json`, and `test.json`. For the Next Pictogram Prediction task, the official data mainly includes `train_next_picto.json` and `test_next_picto.json`.

4.1.1. Text-to-Picto Dataset

For the Text-to-Picto sub-task, each sample in `train.json` and `valid.json` contains an *id*, an English source sentence *src*, a target pictogram term sequence *tgt*, and the corresponding *pictos* sequence field. In this task, *src* is used as the model input, and *tgt* is used as the supervised training target. The `test.json` file only contains *id* and *src*, without reference answers. Therefore, the model needs to generate the predicted result *hyp* according to the input *src*.

In the local experiment, we checked whether there were duplicate samples between the training set and the validation set. The result is shown in Table 1.

This result indicates that there is no sample overlap between the training set and the validation set. Therefore, the validation results can better reflect the model performance on unseen data.

4.1.2. Next Pictogram Prediction Dataset

For the Next Pictogram Prediction sub-task, each sample in `train_next_picto.json` contains an *id*, an English source sentence *src*, a target pictogram term sequence *tgt*, and the corresponding *pictos* sequence field. In this work, we mainly use the *tgt* field for modeling. Each *tgt* represents a complete pictogram keyword sequence. We split the sequence into an input context and a prediction target,

Table 2
Parameter settings for the Text-to-Picto sub-task.

Parameter	Setting
Base model	T5-small
Training data	`train.json`
Validation data	`valid.json`
Maximum input/output length	128
Learning rate	3e-4
Batch size	4
Number of epochs	12
Decoding method	Beam Search
`num_beams`	4
Best checkpoint	`checkpoint-177840`

where the last token is used as the ground-truth answer and the preceding tokens are used as the model input.

For example:

tgt: `the seat of be city`

Input: `the seat of be`

Answer: `city`

The `test_next_picto.json` file contains *id*, *tgt*, and *pictos* fields. The final submission requires a JSON file in which each test instance corresponds to one entry. Each entry must contain the instance name *id* and a ranked list of up to ten candidate tokens, ordered from the most likely to the least likely.

In the local experiment, we use the first 10% of `train_next_picto.json` as the local validation set and the remaining 90% as the training set.

4.2. Parameter Settings

The Text-to-Picto sub-task is fine-tuned using the pre-trained T5-small model, which contains approximately 60M parameters. The main parameter settings are shown in Table 2.

We choose T5-small as the base model for the Text-to-Picto sub-task. Its parameter size is relatively small, and the training cost is low, making it suitable for full fine-tuning and repeated validation under limited computational resources. The Text-to-Picto task has a limited data scale, and using a smaller model can reduce the risk of overfitting while ensuring that experiments can be stably reproduced in a local environment.

The batch size is set to 4 due to local computational resource and GPU memory constraints. During the T5-small fine-tuning process, a larger batch size would significantly increase GPU memory usage and may lead to training instability or failure to run. Therefore, we adopt batch size=4 as a setting that can stably complete training under the current hardware environment. This setting was not obtained through large-scale hyperparameter search, but rather selected as a trade-off to ensure that training can run and results can be reproduced.

For the Next Pictogram Prediction sub-task, the maximum n-gram order is set to 3. Therefore, 1-gram, 2-gram, and 3-gram models are jointly used for prediction. The n-gram model is a non-neural statistical model and therefore has no trainable neural parameters. Its size is determined by the number of stored 1-gram, 2-gram, and 3-gram frequency tables. The global probability weight is set as:

$$\lambda_0 = 0.05$$

The function of $\lambda_0 = 0.05$ is to give a very small smoothing probability to the global high-frequency tokens, so as to prevent high-order n-gram contexts from having no chance of being a candidate when

they haven't been seen before. The remaining 95% of the weight is linearly distributed according to the n-gram order:

$$\lambda_1 = 0.95 \times \frac{1}{1+2+3}$$
$$\lambda_2 = 0.95 \times \frac{2}{1+2+3}$$
$$\lambda_3 = 0.95 \times \frac{3}{1+2+3}$$

The 3-gram model receives a higher weight because it contains longer contextual information. At the same time, 1-gram and 2-gram probabilities are retained to provide supplementary information when higher-order contexts are sparse.

4.3. Training Settings

4.3.1. Text-to-Picto Training Settings

In the Text-to-Picto sub-task, we formulate the task as a text-to-text sequence generation problem. During training, the *src* field in `train.json` is used as the input text, and the corresponding *tgt* field is used as the target output. To make the task objective explicit, we add a task-specific prompt before each input sentence. The input and output formats are defined as follows:

Input format: `translate English to pictogram terms: {src}`

Target output: `{tgt}`

The model is fine-tuned using the Hugging Face Transformers framework. During training, the tokenizer is used to encode both the input text and the target text, and the target sequence is used as the supervised label. The model updates its parameters on the training set, and after each epoch, it is evaluated on the validation set and checkpoints are saved.

4.3.2. Next Pictogram Prediction Training Settings

In the Next Pictogram Prediction sub-task, no neural network model is trained. Instead, the model is constructed by counting n-gram co-occurrence frequencies from the training set. The pictogram term sequence *tgt* is first extracted from each training sample, and the occurrence frequencies of following tokens are counted under contexts of different lengths. In this way, 1-gram, 2-gram, and 3-gram statistical tables are constructed.

For each context, the model records the possible candidate tokens that may appear after it and their corresponding frequencies. These frequencies are then used to calculate conditional probabilities. This statistical process is equivalent to constructing a corpus-based statistical language model.

4.4. Model Selection

4.4.1. Text-to-Picto Model Selection

During the training process of the Text-to-Picto sub-task, checkpoints from different training stages are saved. Since checkpoints from different epochs may have different generalization ability, we evaluate several late-stage checkpoints on the validation set and select the final model according to three official metrics: sacreBLEU, METEOR, and PictoER [2, 8, 9]. Higher sacreBLEU and METEOR scores indicate better generation quality, while a lower PictoER score indicates fewer pictogram-token-level errors.

The purpose of this comparison is to select the final checkpoint for test set prediction rather than to analyze the full training curve. Therefore, we mainly compare late-stage checkpoints around convergence. The results are shown in Table 3. As shown in Table 3, `checkpoint-177840` achieves the best performance among the evaluated checkpoints, with the highest sacreBLEU and METEOR scores and the lowest PictoER.

Table 3
Comparison of different checkpoints on the validation set.

Checkpoint	Epoch	sacreBLEU ↑	METEOR ↑	PictoER ↓
checkpoint-163020	11	51.14	70.57	32.58
checkpoint-177840	12	51.34	70.71	32.56
checkpoint-192660	13	51.20	70.58	32.80

4.4.2. Next Pictogram Prediction Model Selection

The Next Pictogram Prediction sub-task adopts a fixed interpolation smoothed n-gram method. We choose the n-gram model with a maximum order of 3 and combine 1-gram, 2-gram, 3-gram, and global token probabilities. The reason for choosing this method is that higher-order n-grams can capture more specific contextual collocation patterns, while lower-order n-grams and global probabilities can alleviate the data sparsity problem. Additionally, we tried different maximum n-gram orders, and finally chose 3-gram because it performed better in the official submission results; higher-order n-grams may capture longer contexts, but they did not bring better overall rankings in the official evaluation.

4.5. Prediction Settings

For the Text-to-Picto sub-task, we use the T5-small checkpoint with the best validation performance to perform inference on the test set. Given an input sentence *src*, the model generates the corresponding pictogram term sequence *hyp*.

For the Next Pictogram Prediction sub-task, the model calculates the comprehensive score of all candidate tokens and ranks them in descending order. The top 10 candidates are then selected as the output. If the number of candidates is less than 10, the globally high-frequency target tokens in the training set are used to complete the candidate list.

4.6. Evaluation Metrics

For the Text-to-Picto sub-task, three official metrics are used for evaluation:

- **sacreBLEU**: measures the number of common n-grams between the translation hypothesis (*hyp*) and the reference translation (*tgt*). Higher scores indicate better n-gram overlap between the generated and reference sequences.
- **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 (such as lemmas and word roots), and paraphrases. This metric captures additional semantic information that is not encoded in the BLEU score. Higher scores indicate better generation quality.
- **PictoER**: an official task-specific metric derived from Word Error Rate (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. Lower scores indicate fewer pictogram-level token errors.

For the Next Pictogram Prediction sub-task, two official evaluation metrics are used:

- **Top-N Accuracy** ($N = 1, 3, 5, 10$): the fraction of instances where the ground-truth token appears among the top-N candidates. Top-1 accuracy is the primary ranking metric.
- **Semantic Similarity**: the 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 metric gives partial credit for semantically close predictions, e.g., predicting “city” when the answer is “metropolis.”

Table 4
Validation results of the Text-to-Picto sub-task.

Method	sacreBLEU ↑	METEOR ↑	PictoER ↓
Baseline	4.27	35.71	94.21
T5-small fine-tuned	51.34	70.71	32.56

Table 5
Effect of the maximum n-gram order on the local validation set.

Max order	Top-1	Top-3	Top-5	Top-10	Sem. Sim.
1	0.138841	0.261191	0.312863	0.374274	0.371577
2	0.182975	0.286477	0.331867	0.392493	0.407599
3	0.193341	**0.286791**	**0.332025**	**0.395791**	0.417126
4	0.193812	0.285849	0.331867	0.393592	**0.417460**
5	0.192869	0.284121	0.330611	0.391707	0.416021
6	**0.194283**	0.281294	0.330925	0.390922	0.413850
7	0.193184	0.279252	0.328883	0.390451	0.412053
8	0.192869	0.276268	0.326527	0.389980	0.410866

5. Results and Discussion

5.1. Text-to-Picto Results And Analysis

We first evaluate the copy baseline and the fine-tuned T5-small model on the validation set. The baseline directly copies the input English sentence as the predicted pictogram term sequence without using any trained model. The results are shown in Table 4.

The copy baseline obtains only 4.27 sacreBLEU and a PictoER of 94.21, confirming that direct copying cannot effectively generate target pictogram sequences. The fine-tuned T5-small model significantly outperforms the baseline, with sacreBLEU increasing from 4.27 to 51.34, METEOR from 35.71 to 70.71, and PictoER decreasing from 94.21 to 32.56, suggesting that the model is capable of capturing the transformation patterns from English text to pictogram sequences.

As shown in Table 3, `checkpoint-177840` performs slightly better on sacreBLEU, METEOR, and PictoER, and is therefore selected as the final model for test set prediction.

Official test set results. The best-performing checkpoint (checkpoint-177840) is used to generate predictions on the official test set (`test.json`). On the official evaluation leaderboard, the submitted results achieve a WER of 0.3255, a BLEU score of 51.34, and a METEOR score of 0.704, ranking 4th among all participants. These official scores are highly consistent with the validation set performance (sacreBLEU 51.34, METEOR 70.71, PictoER 32.56), confirming that the fine-tuned T5-small model generalizes well to unseen test data.

5.2. Next Pictogram Prediction Results And Analysis

To analyze the effect of different context lengths, we evaluate maximum n-gram orders from 1 to 8 on the local validation set. The results are shown in Table 5.

According to the official evaluation, Top-1 accuracy is the primary ranking metric [10]. As shown in Table 5, increasing the maximum n-gram order from 1 to 3 improves all metrics, indicating that longer local pictogram contexts are useful. However, further increases do not yield consistent gains. Although max-order 6 achieves the highest Top-1 accuracy (0.194283), the improvement over max-order 3 (0.193341) is only 0.001. Meanwhile, max-order 3 obtains the best Top-3, Top-5, and Top-10 accuracy, and its Semantic Similarity (0.417126) is nearly identical to the best value (0.417460 for max-order 4). We therefore select max-order 3 as the final configuration, achieving a Top-1 accuracy of 0.193341 and a Top-10 accuracy of 0.395791 on the local validation set.

Official test set results. The max-order 3 model achieves a Top-1 accuracy of 0.0785, a Top-3 accuracy of 0.1284, a Top-5 accuracy of 0.148, a Top-10 accuracy of 0.1934, and a Semantic Similarity of

0.3236, ranking 2nd among all participants. The official test set scores are lower than the local validation results, which is expected since the local set shares the same training distribution while the official test set is fully unseen data.The 2nd-place ranking demonstrates the effectiveness of the fixed interpolation smoothed n-gram method.

6. Conclusion and Future Work

This paper proposes different methods for two sub-tasks in ImageCLEFtoPicto 2026. For the Text-to-Picto sub-task, we formulate it as a text-to-text sequence generation problem and fine-tune a pre-trained T5-small model. The experimental results show that the fine-tuned T5-small model significantly outperforms the direct copy baseline. On the validation set, the T5-small model achieves a sacreBLEU score of 51.34, a METEOR score of 70.71, and a PictoER score of 32.56. For the Next Pictogram Prediction sub-task, this paper proposes and implements a fixed interpolation smoothed n-gram method. This method predicts the next pictogram term by counting the co-occurrence relationships between contexts of different lengths and following tokens in the training set, and by combining 1-gram, 2-gram, 3-gram, and global token probabilities with fixed weights. The experimental results show that this method achieves a Top-1 accuracy of 0.193341 and a Top-10 accuracy of 0.395791 on the local validation set.

Future work can be carried out from several aspects. First, for the Text-to-Picto sub-task, stronger pre-trained generation models, such as FLAN-T5-small, T5-base, or FLAN-T5-base, can be explored to improve the text generation ability of the model [11, 12]. Additionally, different batch sizes (e.g., 4, 8, 16) and gradient accumulation methods can be investigated to further analyze their impact on model performance. For the Next Pictogram Prediction sub-task, future work can further optimize the interpolation weights. For example, the optimal weights can be automatically searched on the validation set instead of being manually set as linear weights. More advanced smoothing methods, such as Kneser–Ney smoothing, can also be explored to better handle unseen contexts and sparse data [13, 7]. Finally, both sub-tasks can further explore joint modeling of the source text *src* and the pictogram sequence *tgt*. For the Next Pictogram Prediction task, if the model can use both the original text semantics and the existing pictogram context, it may predict semantically appropriate candidate pictograms more accurately.

Acknowledgments

This work is supported by the Guangdong Province Basic and Applied Basic Research Fund (Guangdong-Foshan Joint Funds, Grant No. 2024A1515140026). The authors would like to thank the organizers of the ImageCLEFtoPicto 2026 task for providing the dataset and evaluation platform.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly to check grammar and spelling, paraphrase, and reword. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] 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.

[2] 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: CLEF 2026 Working Notes, Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Jena, Germany, 2026.

[3] J. A. Pereira, D. Macêdo, C. Zanchettin, A. L. I. de Oliveira, R. do Nascimento Fidalgo, Pictobert: Transformers for next pictogram prediction, Expert Systems with Applications 202 (2022) 117231.

[4] C. Macaire, D. Fabre, B. Lecouteux, D. Schwab, Overview of the 2025 imagecleftopicto task–investigating the generation of pictogram sequences from text and speech, in: CLEF2025 Working Notes. CEUR Workshop Proceedings, CEUR-WS. org, Madrid, Spain, 2025.

[5] M. J. Hjuler, I. Fabre, Parlez-vous picto? A transformer-based approach for text-to-picto and speech-to-picto translation in french, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum, CLEF 2025, Madrid, Spain, 9-12 September 2025, CEUR Workshop Proceedings, CEUR-WS.org, 2025, pp. 2461–2475. URL: https://ceur-ws.org/Vol-4038/paper_192.pdf.

[6] A. Koushik, J. Morrison, P. Mirunalini, J. Aditya, A transformer based approach for text-to-picto generation, in: Conference and Labs of the Evaluation Forum, 2024. URL: https://api.semanticscholar.org/CorpusID:271801485.

[7] S. F. Chen, J. Goodman, An empirical study of smoothing techniques for language modeling, Computer Speech & Language 13 (1999) 359–394.

[8] M. Post, A call for clarity in reporting bleu scores, in: Proceedings of the third conference on machine translation: Research papers, 2018, pp. 186–191.

[9] 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.

[10] AI4MediaBench, Imagecleftopicto 2026: Next-pictogram prediction, 2026. URL: https://ai4media-bench.aimultimedialab.ro/competitions/11/#overview/page/2, accessed: 2026-05-26.

[11] 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.

[12] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al., Scaling instruction-finetuned language models, Journal of Machine Learning Research 25 (2024) 1–53.

[13] R. Kneser, H. Ney, Improved backing-off for m-gram language modeling, in: 1995 international conference on acoustics, speech, and signal processing, volume 1, IEEE, 1995, pp. 181–184.