

Working Note for ImageCLEFtoPicto 2026: A Statistical Backoff Model for Next-Pictogram Prediction

Notebook for the ImageCLEF Lab at CLEF 2026

Jishnu PL¹, Santhosh M¹, Saisaranya BL¹ and Anusha Jayasimhan¹

¹Department of Computer Science and Engineering, SSN College of Engineering, Chennai, India

Abstract

This paper describes our submission to the ImageCLEFtoPicto 2026 task, which requires predicting the next pictogram token given a partial pictogram sequence. We approached this as a statistical next-token prediction problem over a constrained vocabulary of 6,841 pictogram names. Our system builds a trigram-based backoff language model with Laplace smoothing, falling back to bigram and unigram probabilities when higher-order counts are unavailable. We further augment the ranking step with source-word boosting, which uses the original English sentence to promote pictograms whose names match words in the source text. Training on 63,675 sequences, our system achieved a Precision@1 of 0.0529, Precision@5 of 0.1339, Precision@10 of 0.1641, and an MRR of 0.324 on the official test set for the 2026 edition, placing us at Rank 19 on the leaderboard. The system runs entirely on CPU hardware, requires under 100 MB of memory, and processes the full test set in approximately 10–15 seconds, demonstrating that a simple statistical baseline can be both practical and reproducible for the pictogram prediction task.

Keywords

ImageCLEF, toPicto, pictogram prediction, n -gram language model, AAC, NLP

Source Code: <https://github.com/jishnupl07/S2J>

1. Introduction

Augmentative and Alternative Communication (AAC) systems help individuals with speech or language impairments express themselves by converting text or speech into sequences of pictographic symbols. The ImageCLEFtoPicto challenge, organised as part of the CLEF evaluation campaign, benchmarks systems that automatically generate such pictogram sequences from natural language input [1, 2].

The 2026 edition introduced a next-pictogram prediction subtask. Given a partial sequence of pictogram tokens, a system must predict the most likely token to follow. This framing is similar to language modelling: the goal is to learn the distributional patterns of pictogram sequences from training data and use them to rank candidates at inference time.

Our team, S2J, tackled this with a statistical backoff n -gram model. We chose this approach for three reasons. First, n -gram models are simple to implement, easy to inspect, and fast to train. Second, the constrained vocabulary of 6,841 pictogram names makes smoothing tractable. Third, we wanted to establish a clear, reproducible baseline against which neural approaches can be compared in future work.

2. Related Work

Statistical n -gram language models have a long history in natural language processing and remain a standard reference point for sequence prediction tasks. Early work on estimating n -gram probabilities from sparse data introduced the back-off strategy of falling to lower-order models when higher-order counts are unreliable [5], which was later refined with discounting schemes that adjust lower-order distributions based on context diversity rather than raw frequency [6]. A broad empirical comparison

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

✉ Jishnu2570010@ssn.edu.in (J. PL); santhosh2570020@ssn.edu.in (S. M); saisaranya2570012@ssn.edu.in (S. BL); anusha.jayasimhan@ssn.edu.in (A. Jayasimhan)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

of these and other smoothing techniques, including Laplace, Katz, and modified Kneser–Ney smoothing, established modified Kneser–Ney as a strong general-purpose choice for language modelling [7], and later work extended interpolated Kneser–Ney smoothing with skipped n -grams to better cope with sparse training data [9]. Good–Turing style frequency re-estimation has similarly been used to redistribute probability mass toward unseen events [8]. At web scale, simplified backoff schemes such as Stupid Backoff have been shown to approach the quality of Kneser–Ney smoothing while remaining far cheaper to train on very large corpora [10]. Foundational treatments of n -gram language modelling, smoothing, and evaluation are also summarised in standard NLP references such as Jurafsky and Martin’s textbook [11]. Laplace (add-one) smoothing in particular adds a constant of 1 to all counts before normalisation; while it is known to over-smooth in large vocabularies, it is computationally inexpensive and easy to reason about, which makes it a natural first choice for a baseline system.

Within AAC specifically, pictogram sequence generation and prediction have recently been explored with the ImageCLEFtoPicto task, first introduced as a text- and speech-to-pictogram translation benchmark [3] and continued in 2026 as part of the broader ImageCLEF multimodal evaluation campaign [1, 2]. On the neural side, transformer-based models have also been applied directly to the next-pictogram prediction problem, learning contextual embeddings over pictogram vocabularies in place of discrete count-based statistics [4]. These methods typically achieve higher accuracy but require significantly more computation. Our work follows the statistical baseline tradition to provide a lightweight reference point for the ImageCLEFtoPicto 2026 benchmark.

3. System Description

3.1. Dataset and Vocabulary

The organisers provided `train_next_picto.json` and `test_next_picto.json`. The training set contains 63,675 samples and the test set contains 1,986 samples. Each sample has four fields: a sample identifier (`id`), a source English sentence (`src`), a target pictogram sequence (`tgt`), and numerical pictogram identifiers (`pictos`).

We extracted a vocabulary of 6,841 unique pictogram token names from the training sequences. Before building the language model, every training sequence was prepended with two special start tokens (`<s>`), so that trigram counts are well defined even at the beginning of a sequence.

3.2. Statistical Backoff Model

Our core model is a trigram language model with Laplace smoothing and backoff. At prediction time the model selects the highest-order n -gram for which the context is non-empty, backing off to lower orders as needed.

Trigram prediction. When both preceding tokens are known, the probability of candidate token w given context (p_2, p_1) is:

$$P(w \mid p_2, p_1) = \frac{\text{count}(p_2, p_1, w) + 1}{\sum_v (\text{count}(p_2, p_1, v) + 1)} \quad (1)$$

Bigram fallback. When only one preceding token is available, or when no trigram count exists for the context, we fall back to:

$$P(w \mid p_1) = \frac{\text{count}(p_1, w) + 1}{\text{count}(p_1) + V} \quad (2)$$

where $V = 6,841$ is the vocabulary size.

Table 1

Official evaluation results for team S2J.

Metric	Score
Precision@1	0.0529
Precision@5	0.1339
Precision@10	0.1641
MRR	0.324

Unigram fallback. When no context is available, we use the smoothed unigram distribution:

$$P(w) = \frac{\text{count}(w) + 1}{N + V} \quad (3)$$

where N is the total number of training tokens.

All three distributions are defined over the full vocabulary, so every candidate can always receive a non-zero probability.

3.3. Source-Word Boosting

In addition to the n -gram score, we apply a source-word boosting step at inference time. For each test sample we tokenise the source English sentence and check whether each pictogram candidate name (or a substring of it) appears in the source token set. Candidates that match receive a multiplicative score boost before final ranking. This lightweight heuristic injects lexical signal from the original sentence without requiring any translation model or embedding lookup.

3.4. Inference Procedure

At inference time the procedure is as follows. Given a partial pictogram sequence of length k , we form the context using the last two tokens (tokens $k - 1$ and k). We then score all vocabulary candidates using the highest available n -gram order, apply source-word boosting, and return the top-10 ranked candidates as the prediction for that sample.

3.5. Implementation Details

The entire system was written in Python 3 using only the standard library modules `json`, `collections`, and `os`. No external machine-learning libraries were used. Training (counting all n -grams and storing them in Python dictionaries) completed in under two seconds on a standard laptop CPU. Memory usage remained below 100 MB throughout. Inference over the full 1,986-sample test set required approximately 10–15 seconds. All code and experimental scripts are available at the repository linked above.

4. Results

Our team achieved Rank 19 on the official ImageCLEFtoPicto 2026 leaderboard. Table 1 reports the four official evaluation metrics.

The gap between Precision@1 and Precision@10 is notable: the correct pictogram appears in the top-10 list roughly three times more often than it appears at rank 1. This suggests that the model’s ranking function places the correct answer in the neighbourhood of the right prediction but often does not rank it first.

4.1. Ablation Study

To understand the contribution of each component we ran controlled ablation experiments on a 20% held-out validation split drawn from the training data (stratified by sequence length). We evaluated

Table 2

Ablation results on the held-out validation set.

Configuration	MRR (val)
Full model (trigram + backoff + Laplace + boosting)	0.318
Without source-word boosting	0.290
Bigram only (no trigram)	0.241
Unigram only	0.112

four configurations using Mean Reciprocal Rank (MRR) as the metric, since it rewards models that place the correct answer higher in the ranked list.

The results show a clear ordering: adding trigram context over bigrams improves MRR by roughly 0.077, and source-word boosting adds a further 0.028. Unigram-only performance is substantially lower, confirming that sequential context is the main driver of prediction quality.

4.2. Observed Failure Patterns

Inspection of the prediction output file revealed a consistent and informative failure pattern. A cluster of pictogram tokens — specifically *revolting*, *tiredness*, *curved_line*, *mortar*, and *city_walls' interpretation_centre* — appears together in the top-10 list for a large fraction of test samples. This cluster corresponds to infrequent source contexts for which no useful trigram or bigram count exists, causing the model to fall back to unigram probabilities. Because these particular tokens happen to have moderate unigram frequencies in the training data, they dominate unigram-fallback predictions even when they are semantically unrelated to the input.

We identify three main failure modes:

- **Unigram fallback bias.** When the two preceding pictograms form a trigram context that was never seen in training, the model falls entirely to unigram probabilities and produces the same fallback cluster for many different inputs. This accounts for a substantial portion of the test samples.
- **High-frequency token dominance.** Pictograms such as *the*, *of*, *by*, and *metropolis* are extremely common in the training sequences. Even with trigram context available, these tokens receive high scores and displace less frequent but more contextually appropriate candidates.
- **Source-word mismatch.** Source-word boosting relies on exact string matching between the English source tokens and the pictogram names. When pictogram names are compound (e.g. *city_walls' interpretation_centre*) or when the English word and the pictogram name differ morphologically, no boost is applied and the signal from the source sentence is lost.

5. Discussion

One practical advantage of this approach is that it runs on any machine without special setup — no GPU, no external libraries, just standard Python. Training takes under two seconds and the full test set finishes in about 10–15 seconds, which makes the system easy to run and check.

The ablation numbers show that trigram context helps significantly compared to bigram or unigram alone. Source-word boosting also gave a small improvement, which makes sense — if the source English sentence contains the word “school” and a candidate pictogram is named *school*, ranking it higher is a reasonable thing to do. The matching is simple string comparison, but it still moved the needle.

The main problem we found is the fallback behaviour. When the model has not seen a particular two-pictogram context during training, it falls back to unigram counts. Because a small group of pictograms such as *revolting*, *tiredness*, and *curved_line* appear frequently enough in training to accumulate decent unigram scores, they end up in the top-10 list for many unrelated test samples. This is clearly wrong

and hurts Precision@1 quite a bit. A model that can generalise across unseen contexts — such as a small neural language model — would likely avoid this problem.

6. Conclusion

In this paper we described our system for the ImageCLEFtoPicto 2026 next-pictogram prediction task. We built a trigram backoff language model with Laplace smoothing and added a source-word boosting step to make use of the English sentence during inference. The system is fast and runs on CPU without any special hardware.

Our ablation experiments showed that trigram context and source-word boosting both help. Looking at the actual predictions, we also found a clear failure pattern: when the model encounters an unseen two-token context it falls back to a small cluster of unrelated high-frequency pictograms, which hurts Precision@1 quite a lot. This gives a clear direction for future work. A neural model that generalises across unseen contexts would likely fix this problem. We have released the full code so that others can build on this baseline.

Declaration on Generative AI

During the preparation of this work, the authors used **Claude (Anthropic)** and **ChatGPT (OpenAI)** in order to: **grammar and spelling check, paraphrasing and rewording of sentences for clarity, and assistance in formatting references**. After using these tools, the authors 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. Sánchez Salido, A. Barrón-Cedeño, A. García 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, September 21–24, 2026.
- [2] B. Ionescu, H. Müller, D.-C. Stanciu, A. Radu, R.-G. Bolborici, M. Negru, A.-F. Ene, V.-M. Vasilescu, A.-A. Nicolae, L.-D. Ştefan, M.-G. Constantin, M. Dogariu, A.-G. 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. Eryilmaz, 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.-N. 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, September 21–24, 2026.
- [3] 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: CLEF 2025 Working Notes, CEUR Workshop Proceedings, Madrid, Spain, 2025.
- [4] 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, 202:117231, 2022.
- [5] S. M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-35(3):400–401, 1987.

- [6] R. Kneser, H. Ney. Improved Backing-off for M-gram Language Modeling. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 1, pp. 181–184, 1995.
- [7] S. F. Chen, J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. *Computer Speech and Language*, 13(4):359–394, 1999.
- [8] W. A. Gale, G. Sampson. Good-Turing Frequency Estimation Without Tears. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.
- [9] R. Pickhardt, T. Gottron, M. Körner, P. G. Wagner, T. Speicher, S. Staab. A Generalized Language Model as the Combination of Skipped n -grams and Modified Kneser-Ney Smoothing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), 2014.
- [10] T. Brants, A. C. Popat, P. Xu, F. J. Och, J. Dean. Large Language Models in Machine Translation. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 858–867, Prague, Czech Republic, 2007.
- [11] D. Jurafsky, J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 3rd edition (draft), Chapter on N-gram Language Models.