Explaining novel senses using definition generation with open language models

Mariia Fedorova[1], Andrey Kutuzov[1], Francesco Periti[2], Yves Scherrer[1]
[1]University of Oslo, Norway, [2]KU Leuven - Flanders Make, Belgium
Correspondence: mariiaf@ifi.uio.no

Abstract

We apply definition generators based on open-weights Large Language Models (LLMs) to the task of explaining novel word senses, taking target word usages as an input. To this end, we employ the datasets from the AXOLOTL'24 shared task on explainable semantic change modeling, which features Finnish, Russian and German languages. We fine-tune and provide publicly open-source models performing higher than the best submissions of the aforementioned shared task, which employed closed proprietary LLMs. In addition, we find that encoder-decoder definition generators perform on par with their decoder-only counterparts.

1 Introduction and related work

Recent NLP advancements have sparked interest in the computational modeling of semantic change. Thus far, the research community has primarily focused on identifying words that have changed their meaning over time. Existing approaches are primarily based on vector token representations (embeddings) and thus often do not enable the *interpretation* of the novel senses a word has gained (Periti and Montanelli, 2024). Only recently, with the advent of new generative language models, the research community has begun to turn its attention to the interpretation of detected semantic change. One step in this direction was the AXOLOTL'24 shared task on explainable semantic change modeling (Fedorova et al., 2024b).

The shared task was focused on the analysis of diachronic semantic shifts between two time periods, a challenge typical for historical linguists and lexicographers. It consisted of two separate subtasks, given a set of target word x usages (examples):

1. find the usages of x in novel senses;
2. provide human-readable descriptions (such as definitions) of the novel senses.

In this paper, we apply LLM-based definition generators (Noraset et al., 2017; Gardner et al., 2022; Segonne and Mickus, 2023) to the second subtask of AXOLOTL'24, where the participants were asked to create descriptions or definitions for novel word senses. 'Novel' here means a sense which is present in a corpus from the newer ('second') time period, but is not mentioned in a dictionary covering the older ('first') time period. In the simplest form, the task is to provide a correct definition of a novel sense y of a target word x, given a bunch of x usages belonging to y. The performance of the systems was evaluated with BLEU (Papineni et al., 2002) and BERTScore (Zhang et al., 2020), comparing the generated definitions to manually annotated gold definitions. An example instance from the AXOLOTL'24 shared task is given in Table 8 in the Appendix.

The AXOLOTL'24 shared task was offered in three languages: Finnish, Russian and German, with the latter as a 'surprise language' without training and validation datasets. Three teams participated in subtask 2, achieving promising performance, but still leaving ample room for improvements. One of the teams used GlossBERT (Huang et al., 2019) fine-tuned with adapters to match usage examples to senses and definitions retrieved from Wiktionary. Two other teams prompted the GPT 3.5 language model to generate senses and definitions.

Thus, the best-performing systems relied on a closed proprietary LLM and a lexical database, respectively. However, the use of a closed model is not ideal as it lacks transparency, and limits accessibility for researchers. Similarly, the use of existing lexical resources such as WordNet or Wiktionary at test time contradicts the goal of identifying novel senses, as genuinely new meanings are, by definition, absent from established ontologies. For this reason, in this paper we concentrate on the generative approach only and evaluate defini-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 22294–22302
November 4-9, 2025

tion generators built on three different instruction-tuned open-weights LLMs: `mT0`, `Aya-101` and `TowerInstruct`. Our contributions are as follows:

1. We provide open alternatives to the generative systems used by the AXOLOTL'24 participants.
2. We explore the differences between performance of models sharing the same architecture, pretraining procedure (base pretraining with further instruction tuning) and base pretraining data, but having different number of parameters and sizes of the instruction datasets (`mT0` and `Aya-101`).
3. We investigate how the outputs from fine-tuned encoder-decoder (`mT0` and `Aya-101`) and decoder-only (`TowerInstruct`) models differ both in terms of automatic metrics and human evaluation.
4. We investigate how much fine-tuning data is needed for a reliable definition generator.

Our code is publicly released on GitHub[1] and model adapters are available on HuggingFace[2].

2 Data

In order to adapt an existing generative LLM for the task of generating definitions, one needs to fine-tune it on a corresponding dataset. In this work, we utilize two resources: the AXOLOTL'24 training data, and definitions and examples from Dbnary (Sérasset, 2012), a lexicographic resource derived from Wiktionary.

For Finnish and Russian, we fine-tune two versions of each model:

1. on AXOLOTL'24 data only **(a)**,
2. on a combination of AXOLOTL'24 and Dbnary data **(a+d)**.

Since no German training data was provided in AXOLOTL'24, we use only Dbnary for fine-tuning the German models.

Some of the AXOLOTL'24 test set senses and definitions ultimately come from Wiktionary. Thus, to avoid data contamination, we removed from our Dbnary datasets *all the words (along with their senses, definitions and usages) also present in the AXOLOTL'24 test and development sets for all languages*.

[1] https://github.com/ltgoslo/MultilingualDefGen

[2] https://huggingface.co/collections/ltg/definition-modeling-6580c4598ecea67c7d5b1970

Table 1 shows the sizes of data splits in our experiments (in example-definition pairs). The validation set for German comes from Dbnary, while the validation sets for Russian and Finnish come from AXOLOTL'24. Test sets for all languages come solely from AXOLOTL'24 (for more statistics, see Section 3 and Appendix A3 of Fedorova et al., 2024b).

Split	Russian	Finnish	German
Train **(a)**	6,494	93,139	–
Train **(a+d)**	180,072	119,980	322,937
Dev **(a)**	2,026	6,554	–
Dev **(d)**	–	–	19,398
Test **(a)**	2,126	6,725	1,152

Table 1: Data split sizes used in our experiments (example-definition pairs). **a** stands for AXOLOTL'24, **d** stands for Dbnary.

Dbnary greatly increases the amount of fine-tuning data. As shown in Section 4, including it improves the performance.

3 Definition generators

This section motivates the choice of models to fine-tune and describes the training setup.

3.1 TowerInstruct

`TowerInstruct-7B`[3] is Llama-2-7B decoder-only model (Touvron et al., 2023) enhanced with continued pretraining on a mix of 20 billion tokens of monolingual data in ten different languages – English, Portuguese, Spanish, French, German, Dutch, Italian, Korean, Chinese, and Russian – as well as other bilingual data (including Finnish) (Alves et al., 2024) and further fine-tuned on instructions relevant for translation. The choice of a Llama-based model is motivated by its usage in previous works (Periti et al., 2024). We refer to our models fine-tuned from it as `TowerDictionary`.

3.2 mT0-XL

`mT0-XL`[4] is a version of the multilingual encoder-decoder `mT5` model (originally pre-trained on 108 languages) that was instruction fine-tuned on the `xP3` dataset, containing instructions for 13 training tasks in 46 languages with English prompts (Muennighoff et al., 2023). This model is 3.7B parameters

[3] https://hf.co/Unbabel/TowerInstruct-7B-v0.2

[4] https://hf.co/bigscience/mT0-xl

in size, about half the size of `TowerInstruct-7B`. This model was also used in previous works on fine-tuning definition generators (Giulianelli et al., 2023; Kutuzov et al., 2024; Fedorova et al., 2024a). We refer to our models fine-tuned from it as `mT0DefGen`.

3.3 Aya-101

`Aya-101`[5] has the same architecture and pretraining dataset as `mt0-XL` (and even reuses its tokenizer), but is larger in size (13B parameters) and instruction-tuned on 101 languages, including our three languages of interest (Üstün et al., 2024).[6] We refer to our fine-tuned models as `Aya-101-DefGen`.

3.4 Instruction fine-tuning

We employed Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023) applied to all linear layers of the models. Hyperparameters are provided in Appendix A. For each language, the same prompt (to be found in Appendix C) was used in all experiments. We compared several generation strategies and chose beam search for all models.

3.5 Aggregating different definitions for the same sense

The fine-tuned models generate one definition per usage example, but this is not sufficient to solve AXOLOTL'24 subtask 2, which expects predictions in the form of 'target–sense–definition' triplets. Since the AXOLOTL'24 datasets contain more than one usage per sense in most cases, this means that the definitions generated for all usages of one sense must be 'aggregated' to produce a single definition (the 'sense label').

We implemented this aggregation in a very straightforward manner inspired by Giulianelli et al. (2023). Definitions for all usages of a given sense are embedded using the Sentence Transformers model[7] (Reimers and Gurevych, 2020). Then, the 'prototypical embedding' is found by computing the average of all definition embeddings, and the definition with the embedding closest to the prototypical one (by cosine similarity) is chosen as the sense label.

In addition, we ensure that the definitions are unique across senses: we try to avoid cases when two different senses a and b are assigned one and the same sense label. For this purpose, before assigning a sense label we check whether this definition has already been assigned to another sense of the same target word. If the answer is positive, the current sense label candidate is discarded, and the definition next closest to the prototypical embeddings is chosen. Thus, we loop over candidate definitions sorted by their frequency until a generated definition is found which has not been assigned to any sense yet. Only if no such definition is found among the usages of the current sense, we fall back to the most prototypical definition (this results in non-unique sense labels). In our experiments, using this technique resulted in small but consistent improvements across all models and languages.

We have done preliminary experiments to see if instruction-tuned models could summarize the definitions of the same sense, but they turned out to be insufficient to solve this task.

4 Results

The fine-tuned definition generators were used to create definitions for the usages with novel senses from the AXOLOTL'24 subtask 2 test sets, which were then aggregated as described above. In this setup, we assumed that subtask 1 (finding these usages) is already solved. Quoting the shared task organizers, "*the evaluation of Subtask 2 therefore limits itself to evaluating the validity of provided glosses*" (Fedorova et al., 2024b).

The resulting 'target–sense–definition' triplets were evaluated by the official AXOLOTL'24 scoring code.[8] Table 2 reports the performance of `TowerDictionary` `mT0DefGen` and `Aya-101-DefGen` models, as well as the best results achieved by *generative* models for each language from the AXOLOTL'24 leaderboard. We report BLEU and BERTScore, both in the range of $0 - 100$.

The winner Definition generators fine-tuned on open-weights LLMs outperform the best AXOLOTL'24 Subtask 2 *generative* submissions for all three languages under analysis. While `TowerDictionary` and `Aya-101-DefGen` perform better than `mT0DefGen`, it is not possible to define a winner based only on BLEU and BERTScore, since differences between `TowerDictionary` and `Aya-101-DefGen` are not statistically significant

[5] https://hf.co/CohereForAI/aya-101

[6] We do not use its successor `Aya-23`, since it was not optimized for Finnish.

[7] https://hf.co/sentence-transformers/distiluse-base-multilingual-cased-v1

[8] https://github.com/ltgoslo/axolotl24_shared_task/blob/main/code/evaluation/scorer_track2.py

Model	Russian	Finnish	German
TowerDictionary **(a)**	3.98 / 65.64 / 64.98 / 65.22	**5.53** / 66.19 / 67.58 / **66.81**	– / –
TowerDictionary **(a+d)**	**7.22** / 69.49 / 68.78 / **69.04**	4.57 / 65.57 / 67.34 / 66.37	**2.25** / 64.26 / 68.27 / **66.12**
mT0DefGen **(a)**	6.46 / 66.97 / 65.84 / 66.31	4.54 / 63.98 / 66.54 / 65.17	– / –
mT0DefGen **(a+d)**	6.01 / 67.95 / 66.25 / 67.00	**5.54** / 64.72 / 66.67 / 65.61	**2.28** / 63.07 / 65.87 / **64.38**
Aya-101-DefGen **(a)**	6.69 / 68.50 / 67.74 / 68.00	**5.99** / 66.85 / 69.00 / **67.83**	– / –
Aya-101-DefGen **(a+d)**	**6.37** / 68.98 / 67.57 / **68.18**	5.15 / 66.05 / 68.08 / 66.98	**2.39** / 64.33 / 67.74 / **65.96**
Best AXOLOTL'24	2.68 / – / – / 65.64	2.32 / – / – / 67.46	1.00 / – / – / 65.24

Table 2: AXOLOTL'24 subtask 2 scores (BLEU / BERTScore precision * 100 / BERTScore recall * 100 / BERTScore F1 * 100); '**a**' stands for 'fine-tuned on AXOLOTL', '**a+d**' for 'fine-tuned on AXOLOTL+Dbnary'. Best AXOLOTL'24 are the best *generative* approaches used by participants. The best results are highlighted with bold (may be more than one for a language, if the difference is not statistically significant.)

(as per t-test[9]). For this reason, we have conducted a manual error analysis.

4.1 Qualitative analysis

We annotated the generated definitions according to three criteria:

1. The definition has **fluency issues** (Snover et al., 2006): it contains repetitions, wrong dictionary labels such as *'colloquial'* or *'metaphoric'*, wrong punctuation, or the sentence is grammatically incorrect.

2. The definition has **adequacy issues**: the definition contains factual mistakes (e.g., *'Eurasian jay: a bird of the Felidae family'*), it refers to the incorrect sense of the word, or is too broad (e.g., *'Eurasian jay: a small bird'*) or too narrow.

3. The definition is **circular**: the generated definition contains the target word ('definiendum') itself (e.g., *'a table is a sort of a table'*).

Fluency and adequacy issues were annotated manually on random samples of definitions for Finnish (30 samples) and Russian (32 samples), and on the entire test set for German (26 samples). Circularity was detected automatically on the full test sets. Table 3 shows the results of the error analysis.

3.7B vs 13B parameters The comparison of `mT0DefGen` and `Aya-101-DefGen` shows that `mT0DefGen` more often generates semantically incorrect definitions and is more prone to repetitions. A German example: for the sense *'verringern, reduzieren'* (cut down, reduce) of the target word *'abbauen'* `mT0DefGen` generated *'Transitiv: etwas entfernen, entfernen'* (transitive: to remove something, remove), while `Aya-101-DefGen` generated *'Transitiv: etwas Transitiv: etwas reduzieren, verringern'* (transitive: to reduce something, to cut down). For the sense *'etw. Aufgebautes (z.B. Krämerbude) zerlegen, abbrechen'* (to disassemble, to demolish smth. built (e.g. grocer's shop)) `mT0DefGen` generated *'Transitiv, auch reflexiv:; etwas reduzieren, verringern'* (transitive, also reflexive:;to reduce something, to cut down), while `Aya-101-DefGen` output *'Transitiv: etwas entfernen, zerstören'* (transitive: to remove something, to destroy). Thus, outputs of `Aya-101-DefGen` are adequate, while senses in `mT0DefGen`'s outputs are swapped. Therefore, we recommend to prefer `Aya-101-DefGen` upon `mT0DefGen` despite its larger size.

Encoder-decoder vs. decoder-only As for the model architecture itself (encoder-decoder or decoder-only), our results correspond with the findings of the related works that both types of models are suitable for the task. However, `TowerDictionary` performs better in terms of fluency, while `Aya-101-DefGen` provides better adequacy. There is no clear winner in terms of circularity.

Amount of fine-tuning data Augmenting the AXOLOTL'24 training set with Dbnary always improves the results of Russian models in BERTScore. While BLEU on AXOLOTL-only data is higher for encoder-decoder models, the manual inspection shows that Russian models trained on AXOLOTL-only data overfitted to excessively output dictionary

[9] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

Model	Russian			Finnish			German		
	Fluency	Adeq.	Circ.	Fluency	Adeq.	Circ.	Fluency	Adeq.	Circ.
TowerDictionary (a)	**18.8**	40.6	15.5	13.33	86.67	15.1	–	–	–
TowerDictionary (a+d)	53.1	50.0	18.4	**6.67**	76.67	**14.8**	**3.9**	57.7	**3.3**
mT0DefGen (a)	87.5	50.0	15.6	43.33	0.8	21.5	–	–	–
mT0DefGen (a+d)	56.3	43.8	26.8	66.67	86.67	21.5	30.8	76.9	4.9
Aya-101-DefGen (a)	90.6	40.6	**10.6**	26.67	83.33	19.2	–	–	–
Aya-101-DefGen (a+d)	43.8	**37.5**	19.9	33.33	**73.33**	20.7	15.4	**53.9**	4.9

Table 3: Share of definitions with fluency-related issues, adequacy-related issues, and containing circularity (%) – lower values are better. 'a' stands for 'fine-tuned on AXOLOTL', 'a+d' for 'fine-tuned on AXOLOTL+Dbnary'.

labels, which explains their low fluency. Since these labels are common in gold data, they boost BLEU, but do not add much to understanding the word's semantics and may cause higher metrics even if the sense is wrong. Thus, for the Russian split, BERTScore is a more reliable metric and the dataset of 7K instances was not sufficient.

For Finnish, the results are controversial across the models. The reason might be very different sources of data: while in case of Russian both AXOLOTL and Dbnary are sampled from Wiktionary, in case of Finnish AXOLOTL data were borrowed from a historical dictionary. Thus, the data domain should be still preferred over data quantity when training definition generators. This observation also holds for German: its ground truth definitions are also not from Wiktionary and completely lack dictionary labels, while the output of both `mT0DefGen` and `Aya-101-DefGen` is full of 'metaphoric' etc.

Circular definitions For Russian, `Aya-101-DefGen` avoids circularity better than the other two models. For German and Finnish, encoder-decoder models are less prone to circularity than the decoder-only model. Also, fine-tuning on Dbnary has different effects across languages, again proving the importance of high-quality training data.

4.2 Comparison with AXOLOTL'24 submissions

We compared definitions generated by our models to those of the AXOLOTL'24 Wooper-NLP team, which submitted predictions for the highest number of target words. The first notable difference is that Wooper-NLP's definitions are twice as long as the gold answers in terms of character counts, while the length of the generations from our models is better aligned with the ground truth. We also looked through the examples described as problematic in the shared task paper (Fedorova et al., 2024b, Appendix C6). The definitions generated by our fine-tuned models seem to avoid the problem that '*the model doesn't stop after producing the definition, but continues with an explanation or excessive details*'. Also, our definitions are not overly narrow because of repeating named entities from the usage examples (this explains why GPT3.5's definitions are too long and ours are not). The predictions of the fine-tuned models are also less prone to grammatical and spelling errors, and loan translations, which proves that large proprietary models may still have issues with language-specific generation; fine-tuning of open models makes sense even for large and mid-resource languages.

5 Conclusion

In this work, we use large language models fine-tuned on definition modeling to generate labels for novel senses. One can think of this task as updating a dictionary based on a set of new texts.

The participants of the AXOLOTL'24 shared task mostly tackled the problem by using the proprietary GPT 3.5 model. We show that one can instead fine-tune *open-weights* LLMs and achieve a better performance in this subtask with contextualized definitions generated by them. We also publicly share the models.

Our comparison of different base models showed that instruction-tuned encoder-decoder (T5-like) models perform on par with their decoder-only (Llama-like) counterparts. According to automatic metrics, larger models always outperform smaller ones, as well as larger fine-tuning datasets often do. However, not only data quantity, but also its relevance to the domain matters. Our experiments also show that few thousands of fine-tuning data instances may not be enough for a complex semantic task such as definition generation.

Limitations

An obvious limitation of this paper is its focus on the second subtask of AXOLOTL'24 only. This means we are dealing with 'given' novel senses, without the need to actually identify them, and the evaluation is focused on the ability of the systems to produce a sensible sense definition from a set of usages. We leave the challenge of developing an end-to-end approach that solves both subtasks jointly for future work.

Language-specific hyperparameter choice and data preprocessing such as dealing with too long texts and removing special dictionary labels is beyond the scope of this paper. Instead, we try to make language-independent observations.

It is likely that all three base models have been exposed to the Russian test set, which is taken from Wiktionary. However, they produce predictions that are very different from the ground truth, which is not surprising, since definition modeling task was not among their instructions.

It is also important to note that fine-tuned definition generators inherit the biases and peculiarities of the lexical resources they were trained on, which can become a potential risk.

References

Duarte Miguel Alves, José Pombal, Nuno M Guerreiro, Pedro Henrique Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and Andre Martins. 2024. Tower: An Open Multilingual Large Language Model for Translation-Related Tasks. In *First Conference on Language Modeling*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLORA: efficient finetuning of quantized LLMs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Mariia Fedorova, Andrey Kutuzov, and Yves Scherrer. 2024a. Definition generation for lexical semantic change detection. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5712–5724, Bangkok, Thailand. Association for Computational Linguistics.

Mariia Fedorova, Timothee Mickus, Niko Partanen, Janine Siewert, Elena Spaziani, and Andrey Kutuzov. 2024b. AXOLOTL'24 shared task on multilingual explainable semantic change modeling. In *Proceedings of the 5th Workshop on Computational Approaches to Historical Language Change*, pages 72–91, Bangkok, Thailand. Association for Computational Linguistics.

Noah Gardner, Hafiz Khan, and Chih-Cheng Hung. 2022. Definition modeling: literature review and dataset analysis. *Applied Computing and Intelligence*, 2(1):83–98.

Mario Giulianelli, Iris Luden, Raquel Fernandez, and Andrey Kutuzov. 2023. Interpretable word sense representations via definition generation: The case of semantic change analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3130–3148, Toronto, Canada. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Andrey Kutuzov, Mariia Fedorova, Dominik Schlechtweg, and Nikolay Arefyev. 2024. Enriching word usage graphs with cluster definitions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6189–6198, Torino, Italia. ELRA and ICCL.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Francesco Periti, David Alfter, and Nina Tahmasebi. 2024. Automatically generated definitions and their utility for modeling word meaning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14008–14026, Miami, Florida, USA. Association for Computational Linguistics.

Francesco Periti and Stefano Montanelli. 2024. Lexical semantic change through large language models: a survey. *ACM Comput. Surv.*, 56(11).

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Vincent Segonne and Timothee Mickus. 2023. Definition modeling : To model definitions. generating definitions with little to no semantics. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 258–266, Nancy, France. Association for Computational Linguistics.

Gilles Sérasset. 2012. Dbnary: Wiktionary as a LMF based multilingual RDF network. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2466–2472, Istanbul, Turkey. European Language Resources Association (ELRA).

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *Preprint*, arXiv:2307.09288.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

A Instruction tuning hyperparameters

All models were fine-tuned for 1 epoch with the settings and hyper-parameters shown in Table 4.

Weight decay	Tower 0.001, mt0 and Aya 0
Learning rate	Tower 1e-4, mt0 and Aya 5e-5
Warmup ratio	0.05
Batch size	16
Optimizer	paged_adamw_8bit
LoRA rank	256
LoRA alpha	512
LoRA dropout	0.1

Table 4: Settings and hyper-parameters for model fine-tuning.

We compared performance of models trained with `adafactor`[10] and `paged_adamw_8bit`[11] optimizers on the development set and found no significant difference in metrics according to t-test, while the latter required less runtime.

All models were trained on a single AMD MI250x (64 GB) GPU, except for Finnish Tower-Dictionary models, which were trained on a single NVIDIA A100-SXM4-80GB. The training time varied from hours to two days for the largest combination of model parameters and training data, Aya-101 fine-tuned on German Dbnary.

Using weight decay allowed to pass the whole data, while without it the training early stopped after several hundred steps.

B Generation hyperparameters

Table 5 specifies generation parameters used to obtain results presented in the table 2.

[10] https://huggingface.co/docs/transformers/main/en/main_classes/optimizer_schedules#transformers.Adafactor

[11] https://huggingface.co/docs/bitsandbytes/reference/optim/adamw#bitsandbytes.optim.PagedAdamW8bit

num_beams	5
do_sample	False
length_penalty	1.1
early_stopping	True
repetition_penalty	1.1

Table 5: Settings and hyper-parameters for text generation.

We compared different generation strategies[12] both for an encoder-decoder and a decoder-only model on the Russian development split. The exact parameters of various generation strategies that we tried are available in our Github repository[13]. The results can be found in Table 6. The models are "sensitive" to generation parameters. The best results obtained from beam search combination with multinomial sampling and contrastive search[14] may be explained by the known fact (Welleck et al., 2019) that non-deterministic decoding "suffers" less from repetitions. However, to ensure reproducibility of our results, we have chosen the standard beam search for our experiments (also the same paper argues that "being prone" to repetitions may still depend more on how a model was trained rather than on decoding strategies).

Strategy	mT0	Tower
greedy search	-	63.52
multinomial sampling	-	62.8
beam search	-	64.15
beam search multinomial sampling	**67.11**	**64.39**
contrastive search, repetition penalty 1.1	66.87	64.33
contrastive search, repetition penalty 1.2	61.66	64.05
dola decoding	-	63.83

Table 6: Different decoding strategies, Russian development set.

[12] https://huggingface.co/docs/transformers/en/main_classes/text_generation#transformers.GenerationConfig

[13] https://github.com/ltgoslo/MultilingualDefGen/blob/dff165051166b3bdf2a6dedd07904c99868f47ad/src/modeling/decoder_only_predict.py#L25

[14] https://huggingface.co/blog/introducing-csearch

C Prompts

Following Giulianelli et al. (2023), the model input was formatted like a usage example followed by a prompt that can be roughly translated to English as 'What is the meaning of <target word>?'. The prompts were suggested by human speakers of the corresponding languages and are reported in the Table 7. These prompts were used both for fine-tuning and for inference.

Language	Prompt
Russian	Что такое <target word>?
Finnish	. Mitä tarkoittaa <target word>?
German	. Was ist die Definition von <target word>?

Table 7: Prompts for the definition generation models. We also experimented with 'Mikä on <target word>?' for Finnish, but it caused model to generate noun-like definitions for the target words in other parts of speech. This caused lower scores, so we do not report it in the main text.

D Example of an AXOLOTL'24 instance

Target:	cell
Sense:	CELL_3
Period:	new
Usage:	In multicellular organisms, groups of *cells* form tissues and tissues come together to form organs
Definition:	A unit of a living organism

Table 8: An example instance of the AXOLOTL'24 training set in English. In the test set, the *definition* field for the usages from the 'new' time period is blank. For subtask 2, the participants have to submit *(target, sense, definition)* triplets for each novel sense of the target words.

For subtask 2, the test submission must consist of 'target-sense-definition' triplets (bold in the table 8).

E Peculiarities of the German dataset

The lower metrics in Table 2 for German might be explained by a large share of test instances consisting of many sentences, while all data splits for other languages and German Dbnary mostly feature usage examples not longer than one sentence. In order to avoid truncating usage examples before the target word occurrence, we splitted the text into sentences and selected only those containing

the target word lemmas. We used SpaCy[15] model de_core_news_sm for that.

[15] https://spacy.io/usage/models