kNN For Whisper And Its Effect On Bias And Speaker Adaptation

Maya K. Nachesa **Vlad Niculae**
Language Technology Lab, University of Amsterdam
{ m.k.nachesa, v.niculae }@uva.nl

Abstract

Speech recognition performance varies by language, domain, and speaker characteristics such as accent, but fine-tuning a model on any of these categories may lead to catastrophic forgetting. Token-level k nearest neighbor search (kNN), first proposed for neural sequence decoders for natural language generation (NLG) and machine translation (MT), is a non-parametric method that instead adapts using inference-time search in an external datastore, without training the underlying model. We show that Whisper, a transformer end-to-end speech recognition model, benefits from kNN. We investigate the differences between the speech and text setups. We discuss implications for speaker adaptation, and analyze improvements by gender, accent, and age.

1 Introduction

Automatic speech recognition (ASR) has improved significantly over the years. A recent success has been the end-to-end transformer encoder-decoder Whisper model (Radford et al., 2023). Besides its architecture, in the landscape of data scarcity that plagues speech recognition, Whisper stands out in having been trained on over 680,000 hours of transcribed audio on a wide variety of languages. Most of this data was in English, and both the amount of training data and performance for each language varied considerably.

Since fine-tuning models may lead to catastrophic forgetting, research has looked toward non-parametric methods to improve performance. One such method is token-level kNN search, first introduced by Khandelwal et al. (2020) for language generation, and then applied to machine translation (MT) by Khandelwal et al. (2021). This kNN method, described in detail in §2.2, involves storing the hidden states together with each token in a sequence as key-value pairs in an optimized structure called a *datastore*. At inference time, at each step, the model's hidden state is used to search the datastore for the k nearest tokens, and the output probability of the found tokens is changed. One benefit of kNN is that one can create separate datastores depending on any category that one wishes to adapt a model's performance to, without needing to fine-tune the model. Although the datastores require space on the disk, they tend to be smaller than the weights that would otherwise need to be stored for a fine-tuned model, especially for very big models. Additionally, we find kNN promising as it is a rare successful departure from the dominant paradigm of parametric linear classification heads.

Compared to text-to-text language generation models, speech introduces a new variable, namely pronunciation variability, which influences ASR performance both for individual speakers, as well as for speaker groups, which has the potential to introduce new forms of bias.

Our contributions are extending kNN at a token level to the ASR task, assessing the viability of kNN for speaker adaptation, as well as assessing Whisper's bias in Dutch for gender, accent, and age, and how kNN impacts it.

2 Background and Related Work

2.1 Automatic Speech Recognition

Whisper (Radford et al., 2023) is a multilingual transformer end-to-end ASR model. Initially, all Whisper sizes were trained on over 680,000 hours of supervised speech data. Subsequent versions of the large model (large-v2 and large-v3) are trained on more data, including weakly and pseudo-labeled audio. This sets it apart from other models that are partially trained in an unsupervised manner, such as Wav2Vec 2.0 (Baevski et al., 2020), opening up the possibility of transferring techniques previously applied to transformer encoder-decoder text-to-text language generation models.

Findings of the Association for Computational Linguistics: NAACL 2025, pages 6636–6642

2.2 kNN

Both speech and language models tend to fall short outside of the domains that they were trained on Fine-tuning them on these tasks, however, is expensive, and may lead to (catastrophic) forgetting (Dingliwal et al., 2022). Khandelwal et al. (2020, 2021) propose a non-parametric method for adjusting the model output, namely kNN at a token-level for natural language generation (NLG) and machine translation (MT). They create a datastore with hidden states generated from the input and decoded output so far as keys and reference output tokens as values. At inference time, the hidden state for an input and output generated so far is used as a query q to search the datastore for the k nearest neighbors. The probability for each neighbor is obtained by

$$p(k_i) \propto \exp(-d_i), \tag{1}$$

where d represents the distance between the query q and each neighbor. The probabilities of all non-unique tokens are then summed over:

$$p_{k\mathrm{NN}}(y) = \sum_i \mathbb{1}_{y=v_i} p(k_i). \tag{2}$$

Finally, this vocabulary distribution is interpolated with the original model's distribution:

$$p(y) = \lambda p_{k\mathrm{NN}}(y) + (1-\lambda) p_{\mathrm{model}}(y). \tag{3}$$

In NLG and MT, Khandelwal et al. (2020, 2021) show that kNN improves over no kNN, and sometimes even over a fine-tuned model in both in-domain and out-of-domain settings, while needing fewer resources to train the datastore. A downside, however, is the increase in decoding time, as a kNN search needs to be done at each decoding step. This computational expense depends on the size of the datastore and several mitigating strategies have been studied, including the use of smaller datastores (Dai et al., 2023), optimized data structures for approximate neighbor lookups (Johnson et al., 2021), or chunked lookups (Martins et al., 2022).

2.3 Augmenting ASR

Both nonretrieval- as well as retrieval-based approaches have been studied in the context of ASR.

Dingliwal et al. (2022) adapt a large language model (LLM) that reranks ASR hypotheses. They do so by adding embedding parameters ("domain prompts") to the embedding layer of the LLM that they train for a specific domain.

Chan et al. (2023) use a combination of kNN and attention fusion. Their datastore contains audio embeddings as keys, and text embeddings as values. These are then used directly in a cross-attention fusion layer to get the final vocabulary distribution per token. Mittal et al. (2023) create a hidden-state trie that they search over with kNN. Their method ensures that only words that exist in the trie can be output. Wang et al. (2024) use in-context learning by providing the k nearest audio and transcript as prompts to Whisper. The approach by Sarı et al. (2020) attends over speaker i-vectors (identity vectors) in the memory for speaker adaptation.

2.4 Bias in ASR

Bias is found across various categories in ASR. It has been found that in some cases, speech models perform better for women (Koenecke et al., 2020), while in others, they perform worse (Garnerin et al., 2019). Performance also varies by age and accent (Feng et al., 2021, 2024; Fuckner et al., 2023), as well as race (Koenecke et al., 2020), where children, older adults, speakers with a "non-standard" accent, and Black speakers tend to be impacted negatively. In consequence, these groups have less access to accessibility tools, such as voice assistants, and other services that use ASR in the pipeline.

3 Methods

Datasets We use four datasets for our experiments: VoxPopuli (Wang et al., 2021), LibriSpeech (Panayotov et al., 2015), CommonVoice (Ardila et al., 2020) and RixVox (Rekathati, 2023). LibriSpeech is an English-spoken dataset and RixVox is a Swedish dataset. For VoxPopuli we use the English portion of the dataset, and for CommonVoice we use the Dutch portion of version 18.0.

Models We perform all experiments on OpenAI's Whisper speech model (Radford et al., 2023). To assess the effect of kNN on different model sizes and to find the optimal settings, we run a hyperparameter search on the VoxPopuli dataset using the Whisper tiny, medium, and large-v3 models. For the remainder of the datasets, we only tune λ on Whisper large-v3. We used the FAISS library (Johnson et al., 2021) to build the datastores. We used the IVFPQ index with 2048 centroids, code size 64, and 32 probes or partitions with ℓ_2^2 distance.

Tuning the kNN Hyperparameters For our choice of hyperparameters, we follow the setup

Whisper	Dataset	WER	
		Vanilla	kNN
tiny	voxpopuli.en	12.28	12.19
medium		7.48	7.96
large-v3		8.13	7.30
large-v3	LS-clean	1.89	1.85
	LS-other	3.79	3.65
	CV	4.58	4.43
	RixVox	16.71	14.41

Table 1: Overall WER without and with kNN.

(...) de voornaamste punten willen aanstippen.
(...) belangrijke punten willen aanstippen. (...) kwestie die ik zou willen aanstippen, (...) Ik zou graag drie punten willen aanhalen. Zou u mijn excuses willen aanvaarden.

Table 2: Example neighbors for CommonVoice. The top row is the decoded sentence, the bottom four are the four neighbors found inside their original contexts. Red marks the current token being decoded.

by Martins et al. (2023) for λ, T, and k. Thus: $\lambda \in \{0.3, 0.4, 0.5, 0.6\}$, $T \in \{1, 10, 100\}$, and $k \in \{4, 8, 16\}$. For the three different model sizes and the VoxPopuli dataset, we tune on the full range of these parameters. For the remaining datasets, we choose the T and k found to be optimal for Whisper large-v3 and only tune λ. The results are based on the best hyperparameter configuration for each dataset and model combination described above. Any ties are broken at random.

Speaker Adaptation To assess the usefulness of kNN for speaker adaptation, we build personalized datastores for a random subset of 33 speakers in the RixVox datasets. Each datastore only included the hidden states and tokens obtained for each respective speaker. We compare this to building a datastore of the same size for each speaker, but instead filled with random tokens uniformly sampled from the full datastore. For both speaker-level conditions, λ was tuned for each speaker.

Bias We use the CommonVoice dataset to test for bias related to gender, accent, and age before and after applying kNN. Each of these characteristics are self-reported by the speakers.[1] Since not all rows were marked with this information, for each category only a subset of the data was examined.

4 Results

4.1 Main Results

Table 1 shows the main results, while hyperparameter tuning results are deferred to appendix B. We report the word error rate (WER, Vogel et al., 2000), *i.e.*, the number of mistakes divided by the total number of words in the reference, calculated using JiWER.[2] On VoxPopuli, kNN improves the results for the tiny and large-v3 models, while having a negative effect on the medium model. We observe an increase for LibriSpeech, CommonVoice and RixVox. Manual inspection of the errors reveals they are mostly difficult or ambiguous cases, including names, written out accents, old words, or other minor differences in spelling, interjections, or compounds. The other datasets may be noisier or Whisper may be less familiar with them, which could mean there is more room for improvement.

Table 2 shows an example of the nearest neighbors found for a sentence in CommonVoice. Some of the prior contexts of the neighbors include the current decoded context. We also see that some of the following words are the same as the word that was decoded next, and otherwise includes the beginning of the next word. This suggests that the hidden states do not only encode information about the current token, but also about the surrounding tokens.

4.1.1 The Effect of k, T, and λ

Figure 1 shows how k, T, and λ affect the WER on the VoxPopuli dev set with Whisper large-v3.

We observe the following patterns: First, results generally improve for lower λ as we reduce the number of neighbors and for higher λ with more neighbors. Second, setting T higher tends to give better results, but this is also up to variability depending on the other hyperparameters. Finally, we find that, generally, we get the best results at $\lambda = 0.5$.

These results contrast with those obtained by Khandelwal et al. (2020, 2021). They find that a higher k tends to improve performance, whereas we find a mixed trend. It could be that for ASR with a transformer encoder-decoder, adding more neighbors may increase the chance of retrieving irrelevant neighbors. Second, we find that the optimal λ is also different from theirs. They find a lower λ for in-domain kNN, and a higher one for out-of-domain kNN. In this work, Whisper

[1] Gender is not binary, but our analysis is limited by the dataset only including the qualifiers `female_feminine`, `male_masculine`, or no qualifier. Additionally, some recordings were associated with multiple accents. We always picked the first one.

[2] https://jitsi.github.io/jiwer

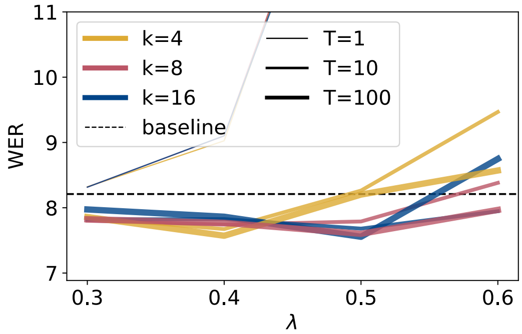

Figure 1: WERs on VoxPopuli.en dev with Whisper large-v3 for different ks (line color), Ts (line width), and λ (x-axis).

has been trained on a large variety of data, very likely including (some of) the datasets used in this work. Thus, applying kNN with VoxPopuli is somewhere between in-domain and out-of-domain tuning. This suggests that the more out-of-domain the dataset used for kNN, the higher λ should be, as the speech (or language) model might be more unfamiliar with the patterns in the target text. This is further supported by needing $\lambda = 0.6$ for RixVox, whereas $\lambda = 0.5$ was best in most other cases. As RixVox was released in 2023, it is less likely, but not impossible, for Whisper large-v3 to be trained on it.

4.2 Speaker Adaptation

In this section we analyze the results for the speaker adaptation experiments. The results can be found in Table 3. Using the complete datastore results in a larger average improvement per speaker compared to any of the other methods. Using a personal datastore leads to smaller improvements and is comparable to using the random datastore of equal size (Section §3). What the smaller datastores lose in accuracy they make up for in efficiency: with the full datastore, a single file takes on average ~1 minute and 15 seconds to transcribe, while for the smaller datastores it only takes roughly 7 seconds. From a private-user perspective, this win in time could be the deciding factor, if it means being able to use some transcription technology or voice assistant live.

4.3 Bias

In this section we analyze the transcription performance with and without kNN on various speaker groups in the CommonVoice dataset. We cover three categories: gender, accent, and age. See Table 4 and Figure 2 for the results.

Setting	Van.	Rand.	Pers.	Gen.
Mean	16.10	15.95	15.97	13.93
Std.	2.74	2.49	2.69	2.67

Table 3: WERs for speaker adaptation on RixVox with different settings. "Van." refers to no kNN, "Rand." and "Pers." are random and personal datastores respectively with a tuned λ per speaker, and "Gen." indicates the full datastore.

Category	Vanilla	kNN	#Test recordings
Overall	4.58	4.43	11309
Women	4.72	4.16	1048
Men	4.70	4.67	3618
Netherlands	4.45	4.33	3546
Belgium	5.29	4.98	1078

Table 4: WER across different categories in CommonVoice 18.0 (NL) with Whisper large-v3.

First, for gender, we see that Whisper performs comparably for both, but with kNN leads to a larger improvement for women than for men.

Second, Dutch speakers from the Netherlands are recognized better than those from Belgium. The WER improves for speakers from both groups, with Belgian speakers benefiting somewhat more. It is difficult to ascertain whether this is due to accent or vocabulary, as the dataset contains Dutch vocabulary from both Belgium and the Netherlands.

When it comes to age, Whisper performs the worst on people in their teens and seventies, and best on those in their fifties.[3] No other clear patterns can be observed. Adding kNN results in an improvement for some age groups.

The accent result cannot be explained by the number of recordings in the training data for kNN, as there are approximately twenty times more recordings of Dutch than Belgian speakers. For gender and age, this is inconclusive, as approximately half the train recordings contain no information on this.

5 Conclusion

In this paper, we find that kNN can improve the ASR performance for Whisper, a transformer end-to-end speech model. Additionally, we observe that using a smaller datastore for individual speakers can still lead to an improvement, trading in the performance from using a full datastore for speed. Finally, we find that Whisper's performance in

[3]The eighties and nineties group are each only represented by one speaker, and have been included for completeness.

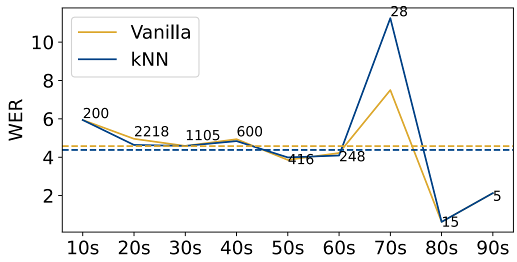

Figure 2: WER per age group for CommonVoice 18.0 NL using Whisper large-v3. The horizontal dashed lines represent the overall results without and with kNN. The numbers in the graph indicate the bin count.

Dutch is similar for women and men, is worse for Belgian speakers of Dutch than for Dutch speakers, and varies by age. Applying kNN shows larger improvements for women and Belgian speakers, and leads to improvements for some age groups, except for teens. Our analysis suggests some improvements from kNN seem to stem from Whisper's decoder's predictive nature, as the context of some of the retrieved neighbors also includes the same decoded continuation.

6 Limitations

In this study, we used a subset of speakers for the speaker adaptation experiment. It is possible that, when using the full dataset, we could have seen a different pattern. Furthermore, due to time-constraints, we fine-tune λ, and fix k and T for the other datasets besides VoxPopuli, which may have led to not observing the full picture for the effect of kNN in different languages and settings. It is also difficult to tell to what extent Whisper is or is not familiar with the data used in this study, as its training data has not been made public. Additionally, all of the languages included in this study are Indo-European of origin, resulting in certain similarities, such as overlapping vocabulary due to cognates, as well as all having a synthetic morphology. They also (mostly) use the same alphabet. More work is needed to see the impact of kNN on languages with other typological features and/or from different language families. For the other datasets besides CommonVoice, more work is needed to assess whether kNN affects all speaker groups equally. Finally, in this work, we took bias to mean unequal performance across the given categories and labels in the dataset. However, for each of the categories a more complete analysis is needed, as gender was only analyzed as binary, there are more accents of Dutch than the ones described, and there was no data for children's speech.

Acknowledgments

The authors thank the members of the TAIM lab and the UvA LTL for helpful suggestions. This publication is part of the project *ROBUST: Trustworthy AI-based Systems for Sustainable Growth* with project number KICH3.LTP.20.006, which is (partly) financed by the Dutch Research Council (NWO), RTL, and the Dutch Ministry of Economic Affairs and Climate Policy (EZK) under the program LTP KIC 2020-2023. VN is partly funded by the Dutch Research Council (NWO) via VI.Veni.212.228 and by European Union's Horizon Europe research and innovation programme via UTTER 101070631.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

David M. Chan, Shalini Ghosh, Ariya Rastrow, and Björn Hoffmeister. 2023. Domain Adaptation with External Off-Policy Acoustic Catalogs for Scalable Contextual End-to-End Automated Speech Recognition. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Yuhan Dai, Zhirui Zhang, Qiuzhi Liu, Qu Cui, Weihua Li, Yichao Du, and Tong Xu. 2023. Simple and scalable nearest neighbor machine translation. In *The Eleventh International Conference on Learning Representations*.

Saket Dingliwal, Ashish Shenoy, Sravan Bodapati, Ankur Gandhe, Ravi Teja Gadde, and Katrin Kirchhoff. 2022. Domain Prompts: Towards memory and compute efficient domain adaptation of ASR systems. In *Interspeech 2022*, pages 684–688.

Siyuan Feng, Bence Mark Halpern, Olya Kudina, and Odette Scharenborg. 2024. Towards inclusive automatic speech recognition. *Computer Speech & Language*, 84:101567.

Siyuan Feng, Olya Kudina, Bence Mark Halpern, and Odette Scharenborg. 2021. Quantifying Bias in Automatic Speech Recognition. *arXiv preprint arXiv:2103.15122v2*.

Marcio Fuckner, Sophie Horsman, Pascal Wiggers, and Iskaj Janssen. 2023. Uncovering Bias in ASR Systems: Evaluating Wav2vec2 and Whisper for Dutch speakers. In *2023 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 146–151.

Mahault Garnerin, Solange Rossato, and Laurent Besacier. 2019. Gender Representation in French Broadcast Corpora and Its Impact on ASR Performance. In *Proceedings of the 1st International Workshop on AI for Smart TV Content Production, Access and Delivery*, AI4TV '19, page 3–9, New York, NY, USA. Association for Computing Machinery.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest Neighbor Machine Translation. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations*.

Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R. Rickford, Dan Jurafsky, and Sharad Goel. 2020. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 117(14):7684–7689.

Pedro Henrique Martins, João Alves, Tânia Vaz, Madalena Gonçalves, Beatriz Silva, Marianna Buchicchio, José G. C. de Souza, and André F. T. Martins. 2023. Empirical Assessment of kNN-MT for Real-World Translation Scenarios. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 115–124, Tampere, Finland. European Association for Machine Translation.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2022. Chunk-based nearest neighbor machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ashish Mittal, Sunita Sarawagi, Preethi Jyothi, George Saon, and Gakuto Kurata. 2023. Speech-enriched Memory for Inference-time Adaptation of ASR Models to Word Dictionaries. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Faton Rekathati. 2023. The KBLab Blog: RixVox: A Swedish Speech Corpus with 5500 Hours of Speech from Parliamentary Debates.

Leda Sarı, Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2020. Unsupervised Speaker Adaptation Using Attention-Based Speaker Memory for End-to-End ASR. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7384–7388.

Stephan Vogel, Sonja Nießen, and Hermann Ney. 2000. Automatic extrapolation of human assessment of translation quality. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens, 31.5-2.6.2000 / LREC '00 / Gavrilidou, M. [u.a.] Hrsg. - Vol. 1*, pages 35–39, Paris. 2. International Conference on Language Resources and Evaluation, Athens (Greece), 31 May 2000 - 2 Jun 2000, ELRA.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.

Siyin Wang, Chao-Han Yang, Ji Wu, and Chao Zhang. 2024. Can Whisper Perform Speech-Based In-Context Learning? In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13421–13425.

A RixVox Splits

We created a new RixVox split for the speaker adaptation experiments. This split was also used for the general RixVox experiments. Table 5 shows the new data distribution.

B Optimal Hyperparameters

Table 6 reports the hyperparameters for our main experiments.

Split	train	dev	test
Hours	1436	30	30
N. speeches	212673	4428	4500
Min speeches per speaker	964	20	21
Max speeches per speaker	5419	113	113

Table 5: RixVox split information.

Model name	Dataset	k	T	λ
tiny		16	100	0.3
medium	VoxPopuli.en	16	100	0.5
large		16	100	0.5
	LibriSpeech	16	100	0.4
large-v3	CommonVoice 18.0 NL	16	100	0.5
	RixVox	16	100	0.6

Table 6: Optimal hyperparameters for different Whisper sizes and datasets.

C Use of AI Assistant

We used GitHub Copilot[4] for auto-completing single lines of code that were repetitive with respect to the rest of our code.

D Compute

For this project, we used approximately 5000 GPU hours. This includes restarts, experiments that were not included, as well as runs leading to the final output. These hours are spread out over the following GPUs: NVIDIA GeForce GTX TITAN X, NVIDIA L40, NVIDIA RTX A6000, NVIDIA TITAN X (Pascal), and Tesla P40.

E Code

The code for this project can be found here: `https://github.com/MKNachesa/kNN`.

[4]https://github.com/features/copilot