

# Blazing Kiwi: a 70 MB model that looks things up in its own library

Technical report on Blazing Kiwi, a 122M-parameter 4-bit English model whose recall head searches an offline library of 49,690 Wikipedia summaries in the browser.

31 October 2026 · Yunus Emre Vurgun · [PDF](#) · [Hugging Face ↗](#)

## Abstract

A model that runs in the browser is paid for twice: once in download time and once in memory. Flyingkiwi's English model, a fine-tune of Qwen3-0.6B stored in 8 bits, is a 571 MB download. Blazing Kiwi is a 122-million-parameter derivative of SmoLM2-135M-Instruct that downloads 69.5 MB of weights and 10.6 MB of reference library, 71 times less than the 0.6B model. Three changes make it small and keep it useful. First, its vocabulary is cut from 49,152 to 27,799 tokens and every matrix, the embedding included, is stored in 4 bits after quantisation-aware distillation from a fine-tuned 1.7B model of the same family. Second, the model addresses an offline library of 49,690 Wikipedia summaries with its own representations: a recall head turns its hidden state, read while it processes the user's message, into a 256-bit key, and the library's passages are stored in the model's own token alphabet, so a BM25 index over those token ids is built in the browser at load time. One scan scores every entry with BM25, the keys' Hamming agreement and a title bonus in under a millisecond, without a separate embedding model or a downloaded index. Third, the retrieved notes are given to the model as a separate conversation turn, a small automaton in the decoder lets it cite only pages that were actually retrieved, and a logit bonus on the notes' own tokens keeps its wording close to them. On 200 questions about held-out library passages, graded blind, version 1.1 answered 120 correctly against 25 for Flyingkiwi's 0.6B model; it downloads 80.9 MB in all, writes about 3.5 times faster in the browser and peaks at about half the memory. It is much weaker at what a library cannot supply: on Flyingkiwi's frozen set of reasoning and conversation cases it scores 25/100 against 64 for unchanged Qwen3-0.6B. A second training run aimed at those cases did not move the score, while larger models of the same family, unchanged, score 30 (360M) and 73 (1.7B): at 70 MB the number of weights sets the limit, and more training of this kind did not raise it.

## 1. Why a smaller model

Every Flyingkiwi model runs entirely on the visitor's device: the weights are downloaded once, verified by hash and cached, and a C engine compiled to WebAssembly computes each token in a background worker. Nothing the visitor types leaves the page. The cost of that design is the download. The English v3.1 model is 571 MB in 17 parts; on a phone connection the first reply can take minutes, and the model needs about 1.3 GB of tab memory while it runs.

Blazing Kiwi's target was set by the owner of the project: at most 70 MB of model, fast enough to feel instant, still useful across many topics, able to answer in the site's structured formats (cards, flowcharts, diagrams, tables, pros and cons), and able to point readers to the right Wikipedia page. A 135M-parameter model cannot hold many facts in its weights. Instead of hoping it remembers, Blazing Kiwi carries a compact library and learns to read it.

The design follows a line of work that separates knowledge from the network that uses it: retrieval-augmented generation [Lewis et al., 2020], nearest-neighbour language models [Khandelwal et al., 2020] and retrieval-enhanced transformers [Borgeaud et al., 2022]. What is specific here is the budget. The retriever may not be a second model (there is no room for one), the index has to fit next to the weights, and the search has to run in WebAssembly on a phone. The answer has two parts that both reuse what the model already has: its own layers produce a binary search key, and its own tokenizer's ids index the library's text.

## 2. Architecture

### 2.1 Starting point

The base is SmoLM2-135M-Instruct [Allal et al., 2025] at revision `12fd25f7`, an Apache-2.0 model trained on about 2 trillion tokens and then instruction-tuned. It is a Llama-style decoder: 30 layers, width  $D = 576$ , 9 query heads and 3 key-value heads of 64 dimensions, a SwiGLU feed-forward block of 1,536, RMSNorm, rotary position embeddings with  $\theta = 100,000$  and an embedding matrix tied to the output layer. Its tokenizer is byte-level BPE with 49,152 tokens, shared with the larger SmoLM2 models, which makes token-level distillation from SmoLM2-1.7B possible (section 3.2).

### 2.2 Vocabulary surgery

Most of SmoLM2's vocabulary serves code and other languages. We counted token use over everything Blazing Kiwi reads and writes (19.2 million tokens: the training conversations and the library), kept every token used at least 37 times, every single-byte token (SmoLM2's vocabulary has 235 of the 256 byte values; the other 21 are control bytes and bytes that never occur in UTF-8 text), Latin letters and punctuation, and closed the set over BPE merge ancestors, so that any text made only of kept tokens is segmented exactly as before. The result has 27,799 tokens. On 3,000 held-out texts it needs 1.97% more tokens than the full vocabulary. The threshold was not chosen by hand: it is the smallest that keeps the whole 4-bit file under 70,000,000 bytes (section 2.3).

SmoLM2's pre-tokenizer (`Digits` then `ByteLevel` with its built-in regular expression) was rewritten as two `Split` steps with the same patterns, the form the site's JavaScript tokenizer implements. On 108,921 texts both produce identical ids.

## 2.3 Four bits for every weight

Each matrix is split along its input dimension into groups of 32 values. A group stores one fp16 scale  $s = \max_i |w_i| / 7$  and 32 four-bit codes  $q_i = \text{clip}(\text{round}(w_i / s), -8, 7)$ ; the engine multiplies  $q_i s$  back on the fly. That is  $4 + 16 / 32 = 4.5$  bits per weight. Unlike Flyingkiwi's earlier models, the tied embedding is stored the same way, because at this size it is a large share of the file. The file size is therefore a function of the vocabulary  $V$ :

$$S(V) = \frac{9}{16} (VD + LP) + c, \quad P = 4D^2 + 2Dd_{kv} + 3DF$$

with  $L = 30$  layers,  $d_{kv} = 192$ ,  $F = 1, 536$ , so  $P = 3, 538, 944$  weights per layer, and  $c$  for the header, norms and the recall head. At  $V = 27, 799$  the file is 69,497,600 bytes. Keeping all 30 layers and shrinking the vocabulary was preferred to dropping layers: the dropped tokens are rare, while every layer is used for every token.

Four-bit rounding of a model trained in 16 bits loses accuracy; Flyingkiwi measured 0.17 nats of extra answer loss for plain 4-bit rounding of its 0.6B model, which is why that model ships in 8 bits. Blazing Kiwi is instead trained for 4 bits (section 3.3), so the file the browser runs is the model that was trained.

## 2.4 The recall head

While the engine reads the user's message, it averages the residual stream after layer  $R$  over the message's tokens  $U$ , after normalising each token's state:

$$h = \frac{1}{|U|} \sum_{t \in U} \frac{x_t^{(R)}}{\sqrt{\frac{1}{D} \|x_t^{(R)}\|^2 + \epsilon}}$$

A query projection maps  $h$  to 256 numbers whose signs form the key:

$$k_q = 1[W_q h + b_q > 0] \in \{0, 1\}^{256}, \quad W_q \in \mathbb{R}^{256 \times 576}$$

Nothing else runs: the state  $x_t^{(R)}$  is computed anyway on the way to the next token, and the projection is 147,712 numbers (0.6 MB in float32, stored exactly). Library entries are encoded the same way ahead of time, but with their own document projection  $(W_d, b_d)$ , so queries and documents may use different directions. Only  $W_q$  ships. The layer was chosen by measurement (section 3.4):

$R = \$24$ . *Binary keys go back to locality – sensitive hashing by random hyperplanes* [Charikar, 2002]; *here the hyperplanes are learned* [Cao et al., 2017]. *###2.5 The library*  $s(q, d) = \text{BM25}(q, d) + \lambda \cdot \frac{256 - 21 \cdot \text{popcount}(k_q \oplus k_d)}{256} + b(q, d)$

*The middle term is the keys' agreement: read as vector of  $\pm 1$ s, the dot product of two keys is the number of agreeing bits minus the number of differing bits*

$\text{BM25}(q, d) = \sum_{t \in q} \text{idf}(t) \cdot \frac{f_{t,d} \cdot (k_1 + 1)}{f_{t,d} + k_1 + 1} \cdot \frac{1}{\text{len}(d)}$   
 $\text{idf}(t) = \ln \left( \frac{N}{n_t + 0.5} \right) / \ln \left( \frac{N}{n_t + 0.5} + 0.5 \right)$  with  $k_1 = 1.2$ ,  $b = 0.75$  and  $N = 49,690$ .  
The inverse document frequencies and the length normalisers are stored in the library file (0.3 MB); the postings themselves (3,280,991 entries) are built by the library controller in the browser from the stored token ids in about 120 ms, so no index is downloaded. Question words such as "what", "where" or "explain" are rare in encyclopedia text and would otherwise weigh a lot; their tokens (440 forms) get zero weight in queries. When the reader picks an answer format, the request the page adds to the message ("Answer as 3 to 5 fact cards...") is left out of both the key and the search, which would otherwise find pages about cards or tables.

Matching ids instead of words has one trap. Byte-level BPE gives a word different ids depending on its case and on whether a space precedes it: "Chlorophyll" at the start of a message, " Chlorophyll" inside one and " chlorophyll" in lower case are three different token sequences, and users rarely type a title the way the article starts. Each entry's indexed title therefore holds all three forms (the passage is indexed as written). With the earlier weights this alone raised the share of simple title questions whose entry was among the two notes from 93.7% to 96.8% (half B, section 4.2).

The last term,  $b(q, d)$ , is a bonus when the entry's title appears word for word in the message (two or more words:  $w_{\text{multi}}$ ; one word:  $w_{\text{single}}$  or  $w_{\text{weak}}$  for titles that are also everyday words such as "Time"). The weights were fitted on half of two held-out question sets (section 4.2), for each version's keys:  $\lambda = 32$ ,  $w_{\text{multi}} = 8$ ,  $w_{\text{single}} = 4$ ,  $w_{\text{weak}} = 4$  for version 1.1 (1.0: 64, 16, 10 and 4). The whole scan over 49,690 entries takes 0.27 ms (median; 0.35 ms at the 90th percentile) in single-threaded WebAssembly (Node 24 on an AMD Ryzen 5 7530U laptop processor).

Why not the key alone? Because measured on held-out questions it is clearly the weakest of the three (section 4.2): a 256-bit projection of a 122M model's state does not separate 49,690 encyclopedia topics on its own. Combined, the three terms beat word-level BM25 over the same library, without its word index.

The two best entries become notes; the next three are offered by title and description only. If the best score is below a gate  $\tau$  (31.88 in 1.1, 47.36 in 1.0), no notes are added.

The library is one file (FKL1): a header, the keys, a fixed-size record per entry, a string table, the passages as 16-bit token ids and an open-addressing hash table of normalised titles. Stored as token ids, the passages take 6.0 MB after gzip against 7.5 MB as gzip-compressed text, and the worker never has to tokenise them again. The file is 17.0 MB uncompressed and 10.6 MB as served.

## 2.6 Reading the notes

Notes enter the conversation as a turn of their own, after the user's message and before the reply. This is the real turn the shipped library (version 1.1) produces for one message, followed by the model's greedy reply (passages shortened; their text is from Wikipedia, CC BY-SA 4.0):

```

<|im_start|>user
What does chlorophyll do?<|im_end|>
<|im_start|>library
[[Chlorophyll]] Green pigments found in plants, algae and bacteria
Chlorophyll is any of several related photochemical catalysts and green pigments found in cyanobacteria and in
the
chloroplasts of algae and plants. ... Chlorophyll allows plants to absorb energy from light.

[[Lutein]] Yellow organic pigment created by plants
Lutein (from Latin luteus meaning "yellow") is a xanthophyll and one of 600 known naturally occurring
carotenoids. ...

Also in the library: [[Yellow-green algae]] Class of algae; [[Plant]] Kingdom of organisms; [[Chloroplast]]
Plant
organelle that conducts photosynthesis<|im_end|>
<|im_start|>assistant
Chlorophyll is a green pigment found in plants, algae, and bacteria that absorbs sunlight to power the process
of
photosynthesis. It helps plants convert carbon dioxide and water into glucose and oxygen.

Read more: [[Chlorophyll]]<|im_end|>

```

A separate role keeps the user's words and the library's words apart, which matters because one is a request and the other is evidence. The model is trained (section 3.1) to answer from a note that answers the question, to ignore notes that do not, and to end with `Read more: [[Title]]` when it used one. The turn is kept in the conversation, so a follow-up is read with the same notes; a follow-up of eight words or fewer also borrows half of the previous message's state  $h$  before its key is formed, so "When did she die?" still points at the person asked about.

## 2.7 A bonus for the notes' own words

A model this small paraphrases loosely: given a note that answers the question, it often writes something close to the note but wrong in a detail. While a reply is written, the decoder therefore adds a constant  $\beta$  to the logit of every token that occurs in the two notes (their titles and passages), the set  $C$ :

$$z'_v = z_v + \beta \cdot 1[v \in C], \quad \beta = 2$$

The bonus does not force copying; it tilts each choice towards the notes' vocabulary, which multiplies a note token's odds by  $e^2 \approx 7.4$  against tokens outside the notes. It is applied together with the answer-format and citation masks and costs one pass over the set per token. With no library turn there is no bonus. On the held-out library questions it is the largest single decoding effect measured (section 4.3).

## 2.8 Citations that cannot be invented

A small model asked to cite will sometimes cite a page that does not exist. Blazing Kiwi's decoder makes that impossible. Before each token, the library controller checks every token whose bytes contain a bracket or backtick against a byte-level automaton: outside a citation every token is allowed; after `[[`, a token is allowed only if the bytes written so far remain a prefix of one of the offered targets  $T$  (the five titles shown in the library turn and their Wikivoyage pages), and `]]` is allowed only after a complete target. With no library turn, `[[` cannot be written at all. Code blocks are exempt. The page then renders `[[Title]]` as a link to the article, so every link a reply contains points to a page that the model was shown. This is the same mechanism Flyingkiwi already uses for its answer formats [Flyingkiwi, 2026], applied to a different grammar.

## 2.9 Engine changes

The C engine reads a new version of its file format. Version 2 adds a header flag for Llama layers (no QK-norm), a 4-bit embedding row decoder and the recall head: `fk_recall_begin`/`fk_recall_end` mark the tokens to pool, `fk_recall_vector` returns  $h$  and `fk_recall_key` returns the key. Version-1 files are read exactly as before: the v3.1 English model produces bitwise-identical logits and generations with the old and the new native builds. The library search and the citation automaton are a separate freestanding C module (`kiwi-recall.c`, 8,232 bytes of WebAssembly, no imports). Both run in the same worker as before. The answer-format controller needed one change: it required a single-byte token for every byte value, but SmoLLM2's vocabulary has 235 of the 256, so it now requires only the three bytes it writes itself (`:`, `|` and newline); the 0.6B models, which have all 256, behave exactly as before.

# 3. Training

## 3.1 Data

Answers were written by Qwen3.5-35B-A3B in FP8 (Apache-2.0), the teacher Flyingkiwi already used, run with vLLM 0.32 on one RTX PRO 6000 (96 GB) in Google Colab. The teacher was told everything Blazing Kiwi is and is not (its size, its library, that it cannot browse) and how to use notes: answer from a note that answers the question, ignore notes that do not, never mention them, and cite with exact titles only. The training set has 30,210 conversations (614 more for validation; one conversation longer than 3,072 tokens was skipped when training, so both models saw 30,209):

| Kind                 | Conversations | What the teacher was asked                                                                                                                                                                                                       |
|----------------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Library questions    | 9,269         | Answer a question written from one passage, with that passage and a BM25 neighbour as notes (in 12%, two neighbours only)                                                                                                        |
| General, re-answered | 6,626         | Answer a Flyingkiwi v3.1 training prompt again with its top two BM25 notes attached, relevant or not (half of the 15,834 v3.1 prompts were drawn for this; 58 others were re-answered only to replace an old identity statement) |
| General, kept        | 7,866         | The other v3.1 training conversations, unchanged apart from the new system prompt                                                                                                                                                |
| Answer formats       | 3,157         | Answer in one of five layouts (cards, flowchart, diagram, table, pros and cons); half with notes                                                                                                                                 |
| Page routing         | 2,133         | "Where can I read more about X?" with notes                                                                                                                                                                                      |
| Follow-ups           | 764           | A second question about the same subject, with new notes                                                                                                                                                                         |
| Identity             | 395           | Questions about Blazing Kiwi itself                                                                                                                                                                                              |

20,317 conversations (67%) contain a library turn and 15,756 answers end with a citation.

Every answer was then filtered. It had to finish, contain no chat markup, cite only titles present in its own library turn and not talk about "the notes" or "the library". The filter rejected 392 library answers, 1,009 general answers and 535 follow-ups, almost all for talking about the mechanism and the rest for citing a page that was not offered, usually in the 12% of library questions where the right passage had been left out on purpose. Every format answer was parsed by the site's own controller and kept only if it formed a complete, valid template (3,223 of 3,800). 600 library passages were held out completely; their questions are the test set of section 4.

### 3.2 A teacher assistant and the distillation objective

A 122M-parameter student learns more from a full next-token distribution than from one sampled answer [Hinton et al., 2015], but the 35B teacher uses a different tokenizer. SmoLM2-1.7B-Instruct shares the student's tokenizer, so it was first fine-tuned on the same conversations (one epoch, learning rate  $10^{-5}$ ) and then used as a teacher assistant [Mirzadeh et al., 2020]. The student's sequences are mapped back to the original token ids, so both models read identical tokens, and the assistant's distribution is renormalised over the kept vocabulary  $\mathcal{V}$ . On every answer position the student minimises

$$\mathcal{L} = \alpha \cdot (-\log p_S(y_t)) + (1 - \alpha) \sum_{v \in \mathcal{V}} p_A(v) \log \frac{p_A(v)}{p_S(v)}, \quad \alpha = 0.5$$

where  $y_t$  is the teacher-written token,  $p_A$  the assistant's distribution and  $p_S$  the student's. The first term is sequence-level distillation from the 35B teacher [Kim and Rush, 2016]; the second passes on how the assistant spreads probability over the alternatives. The student trained for three epochs at learning rate  $10^{-4}$  with cosine decay. The assistant's validation loss on the teacher's answers fell from 1.517 to 1.083 nats per token. The student's cross-entropy on the same answers fell from 1.985 (SmoLM2-135M-Instruct with the new vocabulary, untrained) to 1.464 at step 489, where quantisation-aware training began, and its KL divergence to the assistant from 0.933 to 0.533.

### 3.3 Quantisation-aware training

For the last 30% of steps every matrix, the embedding included, runs forward through the exact 4-bit quantiser of section 2.3 while gradients pass straight through the rounding to the full-precision weights [Bengio et al., 2013; Jacob et al., 2018]:

$$\hat{W} = W + \text{sg}[Q(W) - W]$$

where sg stops the gradient. The exported codes are  $Q(W)$  of the final weights, the numbers the engine multiplies. Rounding to 4 bits is not free at this size. 111 steps after QAT began (step 600), the 4-bit model's validation cross-entropy was 1.591, against 1.464 for the full-precision model at step 489; at the end of the epoch (step 699) it was 1.578, and a second QAT epoch at a lower learning rate (233 steps,  $3 \times 10^{-5}$ ) brought it to 1.549 (KL 0.618). That is about 0.085 nats above the full-precision model at the point QAT began; the released file is this 4-bit model. An 8-bit embedding alone would add about 8 MB and break the 70 MB budget.

### 3.4 Training the recall head

The head was trained after the language model, with the 4-bit student frozen, on 17,096 pairs of a question and the passage it was written from (plus routing questions). Pooled states  $h$  were computed for every question and every library entry at eight candidate layers. For each layer a query head and a document head were trained with a symmetric InfoNCE loss [van den Oord et al., 2018] on relaxed codes  $u = \tanh(\beta z)$ , with  $\beta$  rising from 1 to 6 so that the codes approach their signs [Cao et al., 2017]:

$$\mathcal{L}_{\text{recall}} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(\gamma u_{q_i}^\top u_{d_i} / 256)}{\sum_j \exp(\gamma u_{q_i}^\top u_{d_j} / 256)} + (\text{the same with queries and documents swapped})$$

with  $\gamma = 20$ , the other questions' passages and 1,024 random library entries as negatives. Each head was then scored with real binary keys against all 49,690 entries on the held-out questions. Binary keys only (no BM25, no title bonus), 589 held-out questions against all 49,690 entries:

| Layer     | Correct entry first | In the top 5 | In the top 10 |
|-----------|---------------------|--------------|---------------|
| 6         | 22.6%               | 44.0%        | 54.0%         |
| 9         | 21.6%               | 39.9%        | 49.1%         |
| 12        | 18.8%               | 33.3%        | 42.6%         |
| 15        | 16.1%               | 35.5%        | 45.8%         |
| 18        | 16.0%               | 35.5%        | 44.5%         |
| 21        | 13.6%               | 28.7%        | 37.5%         |
| <b>24</b> | <b>25.6%</b>        | <b>53.0%</b> | <b>63.2%</b>  |
| 27        | 21.7%               | 45.2%        | 55.0%         |

Layer 24 was kept. On its own the key is a weak retriever, which is why the search combines it with BM25 and the title bonus (section 2.5).

### 3.5 Version 1.1: more behaviour data

Version 1.0 was weak at what the library cannot supply (section 4.4), and its data held few examples of those behaviours: 249 false premises and 112 ambiguous requests among 30,210 conversations, inherited from the v3.1 data. Version 1.1 keeps everything else (vocabulary, architecture, library format, browser code) and adds 9,418 teacher conversations for those behaviours, written with the generic prompt generator of the v3.1 data and a new seed: the teacher first writes user messages by category, then answers them as Blazing Kiwi, and about half of them carry library notes, as in 1.0.

| Kind (1.1 additions)                                                   | Conversations |
|------------------------------------------------------------------------|---------------|
| Short reasoning and arithmetic word problems                           | 1,763         |
| False premises                                                         | 1,369         |
| Exact output constraints (length, format, audience)                    | 907           |
| Ambiguous requests                                                     | 783           |
| Friendly chat                                                          | 643           |
| Invented entities (only the teacher is told the entity does not exist) | 561           |
| Private, live, future, exact or obscure facts                          | 1,294         |
| Requests for things it cannot do (browse, send, remember)              | 358           |
| Memory tests: a personal detail, then a message that needs it          | 764           |
| Follow-ups to the conversations above                                  | 976           |

These counts are after the filters of section 3.1 and two more: messages that match an evaluation prompt (the frozen set or the development set; none did) and answers that talk about the notes (606 dropped). The 1.0 conversations are unchanged, so 1.1 trained on 39,627 conversations, with 805 for validation. The assistant was retrained on them (299 steps; validation loss  $1.493 \rightarrow 1.040$ ), and the 1.0 student continued with 4-bit QAT from the first step for two epochs at  $5 \times 10^{-5}$  (598 steps, 18 minutes): on the 805 validation conversations its cross-entropy fell from 1.534 to 1.471 and its KL divergence to the new assistant from 0.621 to 0.579. A new recall head at layer 24 (keys alone: 28.4% first, 53.5% in the top five, 61.6% in the top ten) gave new library keys, and the weights of section 2.5 were refitted on the new keys ( $\lambda = 32$ ,  $w_{\text{multi}} = 8$ ,  $w_{\text{single}} = 4$ ,  $w_{\text{weak}} = 4$ , gate 31.88). The run took 1 h 29 min of one G4 GPU (14.82 compute units). Section 4 reports both versions.

## 4. Evaluation

Everything below was measured on the released files: the 4-bit `.fkm` file, the library exactly as it is served and the site's own browser modules, run in Node or in Chrome. Replies are greedy (temperature 0, repetition penalty 1.1 over 256 tokens, at most 512 new tokens), the protocol of Flyingkiwi's earlier reports; the site itself samples at temperature 0.5. Version 1.1 is the released model; version 1.0 is shown next to it where both were measured. Graded results come from blind packets that hold every answer to a case at once, graded by Claude Haiku subagents from a written criterion; the same answers graded in different packets moved by up to four points, so versions are only compared within one grading.

### 4.1 Size, speed and memory

|            | Blazing Kiwi 1.1                  | Flyingkiwi 0.6B (English v3.1) |
|------------|-----------------------------------|--------------------------------|
| Parameters | 122.2M (plus a 0.15M recall head) | 0.6B                           |

|                                                   | Blazing Kiwi 1.1                                   | Flyingkiwi 0.6B (English v3.1) |
|---------------------------------------------------|----------------------------------------------------|--------------------------------|
| Weights                                           | 4-bit, 69,497,600 bytes in 3 parts                 | 8-bit, 571 MB in 17 parts      |
| Library                                           | 10,612,260 bytes (gzip; 17,027,688 bytes unpacked) | none                           |
| Downloaded on the first visit                     | 80,899,482 bytes                                   | 571,378,688 bytes              |
| Reply speed in Chrome (six threads, relaxed SIMD) | 98–134 tokens/s                                    | 29–35 tokens/s                 |
| Peak memory of the largest Chrome process         | 509–601 MB                                         | 1,237 MB                       |

The download is 7.1 times smaller. Speed and memory were measured in headless Chrome on a laptop processor (AMD Ryzen 5 7530U) with the same three prompts for every model (tools/e2e\_browser.mjs), once for 1.0, once for 1.1 and once for the 0.6B model, which had the same session as 1.0: Blazing Kiwi wrote about 3.5 times as many tokens per second, and the largest Chrome process peaked at about half the memory (resident set high-water mark, the method of Flyingkiwi's earlier reports; the two Blazing Kiwi runs differ by 92 MB with the same file sizes, a measure of run-to-run spread). The library loads in about 0.2 s (74 ms to unpack, 119 ms to read and index in WebAssembly) and occupies 34.2 MB of WebAssembly memory; one search over all 49,690 entries takes 0.27 ms (median of 200; 0.35 ms at the 90th percentile).

## 4.2 Retrieval

Two held-out question sets, each split in two by entry id; the weights were chosen on half A only and every number below is half B. The first set is the 589 questions written by the teacher from the 600 held-out passages (320 in half B); no question or passage of this set was used in training. The second is 600 short questions that name an entry's title ("What is X?", "Tell me about X.", "What do you know about X?") for a fixed random sample of entries (315 in half B), the commonest way people ask and the hardest case for token-level BM25. Hit@k is the share of questions whose entry is among the first k results; hit@2 is the share that get it as a note.

| Retriever (same library)                            | Held-out questions: hit@1 | hit@2        | hit@5        | Title questions: hit@1 | hit@2        | hit@5        |
|-----------------------------------------------------|---------------------------|--------------|--------------|------------------------|--------------|--------------|
| Keys alone (1.1)                                    | 30.3%                     | 40.0%        | 56.6%        | 25.4%                  | 34.0%        | 47.3%        |
| Token BM25 alone                                    | 79.7%                     | 85.3%        | 91.9%        | 89.5%                  | 93.7%        | 96.2%        |
| Word BM25 (a word index, not shipped)               | 79.7%                     | 87.8%        | 92.5%        | 82.9%                  | 87.3%        | 94.9%        |
| <b>Shipped 1.1: token BM25 + keys + title bonus</b> | <b>87.2%</b>              | <b>94.1%</b> | <b>96.9%</b> | <b>96.2%</b>           | <b>97.5%</b> | <b>98.4%</b> |
| Version 1.0 (its own keys and weights)              | 86.2%                     | 92.5%        | 96.9%        | 95.6%                  | 98.4%        | 98.7%        |

The keys alone are weak, as section 3.4 found, but they add to BM25: the combination finds the right passage among the two notes for 94.1% of held-out questions against 85.3% for token BM25 alone. Title questions whose subject is a single word are the hardest (197 of the 600; over all 600, hit@2 96.5%); before the title was indexed in three surface forms and the weights were refitted on both sets, the earlier pack found only 81.2% of them among the two notes. With 1.1's gate, 97.5% of all held-out questions, 95.8% of all title questions and 91.5% of 400 general chat prompts get notes; general prompts mostly receive irrelevant notes, which the model was trained to ignore.

## 4.3 Answering questions about the library

The 200 test questions were drawn from the held-out questions (their passages were never trained on), and every answer was graded against the passage the question was written from, with labels shuffled per case and citation lines removed so that they could not reveal the model. Two gradings were run. The second compared the released versions with the 0.6B model, which has no library:

| Answerer (200 held-out library questions) | Correct    | Flagged for invented details | Reply words found in the passage |
|-------------------------------------------|------------|------------------------------|----------------------------------|
| <b>Blazing Kiwi 1.1</b>                   | <b>120</b> | <b>60</b>                    | <b>67.0%</b>                     |
| Blazing Kiwi 1.0                          | 103        | 76                           | 66.8%                            |
| Flyingkiwi 0.6B (v3.1), no library        | 25         | 134                          | 26.7%                            |

Case by case, 1.1 was right where 1.0 was wrong 40 times and the reverse 23 times (two-sided sign test,  $p = 0.043$ ). The first grading, at 1.0's release, isolated the mechanisms:

| Answerer (first grading)                 | Correct | Flagged for invented details | Reply words found in the passage |
|------------------------------------------|---------|------------------------------|----------------------------------|
| Blazing Kiwi 1.0 (library, copy bonus 2) | 100     | 74                           | 66.8%                            |
| Blazing Kiwi 1.0, copy bonus off         | 58      | 104                          | 44.3%                            |
| Blazing Kiwi 1.0, no library turn        | 8       | 152                          | 24.7%                            |
| Flyingkiwi 0.6B (v3.1), no library       | 23      | 133                          | 26.7%                            |

The library is what makes the model useful on these questions (8 without it), and the copy bonus is the largest single decoding effect (58 → 100). The grading is strict: an answer whose core is right but which adds one false detail counts as wrong, which is why even the best version has 60 flagged answers. Most of these are a wrong work, date or place added to an otherwise correct summary; one reply gives George Romero's dates and role correctly and then credits him with **The Shining**. The last column is the share of a reply's content words (at least four letters, common words excluded, citations removed) that occur in the right passage, a proxy for answering from the note (`tools/blazing/libqa_stats.py`).

Retrieval bounds the result. For 1.1 the right passage was among the two notes for 188 of the 200 questions and among the five offered pages for 193; 198 replies cited a page, 183 of them the right one (1.0: 180, 190, 196 and 176).

#### 4.4 The frozen set

Flyingkiwi's frozen English set has 100 cases in six categories: answerable questions (short arithmetic, logic, reading and instruction-following tasks), questions nobody could answer from the conversation, false premises, ambiguous requests, friendly or personal messages and multi-turn conversations. It tests reasoning and conversational judgement much more than knowledge. The two 0.6B models' answers are the stored outputs of their releases, graded again here with the others in five-answer packets.

| Category                             | Cases      | Blazing Kiwi 1.1 | Blazing Kiwi 1.0 | SmolLM2-135M-Instruct | Flyingkiwi 0.6B (v3.1) | Qwen3-0.6B |
|--------------------------------------|------------|------------------|------------------|-----------------------|------------------------|------------|
| Answerable                           | 30         | 1                | 3                | 1                     | 20                     | 23         |
| Unknowable                           | 20         | 16               | 15               | 5                     | 19                     | 17         |
| False premise                        | 10         | 0                | 0                | 0                     | 6                      | 7          |
| Ambiguity                            | 10         | 1                | 0                | 0                     | 3                      | 0          |
| Friendly                             | 10         | 1                | 2                | 0                     | 8                      | 7          |
| Multi-turn                           | 20         | 6                | 5                | 4                     | 10                     | 10         |
| <b>All</b>                           | <b>100</b> | <b>25</b>        | <b>25</b>        | <b>10</b>             | <b>66</b>              | <b>64</b>  |
| Answers flagged for invented details |            | 20               | 19               | 36                    | 4                      | 4          |

Blazing Kiwi is two and a half times its base model, but far below the 0.6B models: 25 against 64 for the reference. Version 1.1's 9,418 behaviour conversations did not move it (25 and 25; 6 cases won and 6 lost against 1.0), although they were aimed at exactly these categories. It still solves almost none of the answerable cases and corrects none of the false premises; its few attempts start well ("That premise is incorrect because a fictional town does not exist") and then add invented explanations. It does say when it cannot know something (16 of 20). By the project's own rule a release must not score below the reference; Blazing Kiwi is released as the default English model nonetheless, at the owner's explicit request for a model of at most 70 MB, with the 0.6B model one click away and this result shown next to it. Version 1.1 replaced 1.0 although the rule set before its training, a higher frozen score, was not met: it ties on the frozen set and is better on library questions with fewer invented details.

#### 4.5 Engine fidelity

The reference for the engine is PyTorch running the dequantised 4-bit codes in float32, since those codes are what quantisation-aware training trained. On 30 validation conversations (all with a library turn), the KL divergence of the 1.1 engine's distribution at the first reply token from the reference was 0.0014 on average for the native build (largest 0.0072), 0.0007 for WebAssembly SIMD (largest 0.0027) and 0.0008 for relaxed SIMD (largest 0.0066), and the most likely token was the same in all 30 cases for all three builds (1.0: 0.0014, 0.0010 and 0.0012). The remaining differences come from the engine's 8-bit activations and KV cache. The engine's recall keys agree with the reference's on 254.0 of 256 bits on average (at least 252). Version-1 files are unaffected: the v3.1 English model gives bitwise-identical logits with the old and new builds. TruthfulQA MC1 (817 questions, each option scored as the reply under Blazing Kiwi's system prompt) is 24.7% for 1.1 and 25.5% for 1.0, against 25.1% for Flyingkiwi 0.6B and 25.8% for Qwen3-0.6B: at this level the test does not separate the models.

#### 4.6 How much general ability fits in 70 MB?

The goal for Blazing Kiwi included general intelligence across topics, and on that it falls short. To see whether more training could close the gap, the same frozen set was given to the larger models of its own family, unchanged and in 8 bits with Flyingkiwi's English system prompt, and graded in four-answer packets with Blazing Kiwi 1.1 and Qwen3-0.6B as anchors (they scored 22 and 63 there):

| Model                              | Parameters | Weights at 4 bits                 | Frozen set | Flagged for invented details |
|------------------------------------|------------|-----------------------------------|------------|------------------------------|
| SmolLM2-135M-Instruct              | 135M       | about 76 MB                       | 10         | 36                           |
| Blazing Kiwi 1.1 (fine-tuned 135M) | 122M       | 69.5 MB                           | 22–25      | 20–25                        |
| SmolLM2-360M-Instruct              | 362M       | about 204 MB                      | 30         | 28                           |
| Qwen3-0.6B                         | 0.6B       | 341 MB (measured, cut vocabulary) | 63–64      | 4                            |
| SmolLM2-1.7B-Instruct              | 1.71B      | about 963 MB                      | 73         | 4                            |

In this family the score follows the parameter count: the untuned 360M model is ahead of both trained versions of Blazing Kiwi, and the 1.7B model, the base of the teacher assistant, is ahead of Qwen3-0.6B. A 70 MB file holds about 124 million weights at 4.5 bits per weight; reaching the reference's 63 inside it would need a model about four times more parameter-efficient than the best open models we could use, and fewer bits per weight cost accuracy that the extra weights would first have to win back (section 2.3). Two training runs at this size, the second aimed squarely at the failing categories, both stayed at 25. What the library adds is knowledge, and there Blazing Kiwi beats the 0.6B model by a wide margin (section 4.3); what it cannot add is reasoning.

## 5. Limitations

- **General ability.** Blazing Kiwi is much weaker than the 0.6B models at everything the library cannot supply: arithmetic, short logic, false premises, ambiguity and conversation (25 against 64 on the frozen set), and it invents details more often (20 flagged answers against 4). Section 4.6 explains why more training at this size is unlikely to change that. It should not be used for sums or for advice that matters. The 0.6B model remains available in the same menu.
- **What the library holds.** 49,690 lead sections of English Wikipedia's vital articles as of 30 October 2026, at most 110 tokens each (89 on average). Details beyond an article's opening, recent events and subjects outside the list are not there; such questions get notes that do not answer them, or none, and the model then knows little.
- **Retrieval misses.** For 5.9% of held-out questions (half B) neither note is the right passage. The usual misses describe a subject without naming it ("this big plateau in India"; "who runs the modern Olympic Games"). A follow-up borrows the previous message's state only when it has eight words or fewer, a heuristic.
- **Citations are checkable, not proof.** The decoder guarantees that a cited page was offered to the model, not that the sentence before it says what the page says. The copy bonus favours the notes' wording, including notes that do not answer the question.
- **Answer formats.** The format controller keeps every reply inside its template (cards, flowchart, diagram, table, pros and cons), but the text is only as good as the model's: in tests a card about vaccines was titled "Early Vaccination Works as a Precurement of the Infection".
- **Four bits cost something.** Quantisation-aware training left the 4-bit model about 0.085 nats per token above the full-precision model at the point QAT began.
- **English only.** The vocabulary was cut for English text; the Turkish model is unchanged.
- **How it was measured.** Each answer was graded once, by a Claude Haiku subagent working from a written criterion, not by people, and the same answers moved by up to four points between gradings; the frozen set has 100 cases and the library questions were written by the same teacher that wrote the training data, from passages held out of training. The title questions are templates. Speed and memory were measured on one laptop processor (AMD Ryzen 5 7530U) in headless Chrome, not on phones.

## 6. Reproducibility

Code: `training/blazing/` (data, training), `engine/` (C engine, library controller), `tools/blazing/` (library fetch, packing, retrieval tuning, evaluation harness), `public/assets/` (browser modules). The Hugging Face repository holds the BF16 checkpoint, the exact 4-bit codes, the recall head, the browser file, the library with its source text, the WebAssembly builds and the evaluation outputs. Compute: one NVIDIA RTX PRO 6000 (96 GB) in Google Colab, 2 h 6 min in total (teacher data 37 min, assistant SFT 19 min, student 21 min plus a 7 min second QAT epoch, recall heads and exports), 20.92 compute units of an existing Colab Pro+ plan.

## References

- Allal, L. B., et al. (2025). SmolLM2: When Smol Goes Big — Data-Centric Training of a Small Language Model. arXiv:2502.02737.
- Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. arXiv:1308.3432.
- Borgeaud, S., et al. (2022). Improving Language Models by Retrieving from Trillions of Tokens. ICML.
- Cao, Z., Long, M., Wang, J., and Yu, P. S. (2017). HashNet: Deep Learning to Hash by Continuation. ICCV.
- Charikar, M. S. (2002). Similarity Estimation Techniques from Rounding Algorithms. STOC.
- Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the Knowledge in a Neural Network. arXiv:1503.02531.
- Jacob, B., et al. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. CVPR.
- Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., and Lewis, M. (2020). Generalization through Memorization: Nearest Neighbor Language Models. ICLR.
- Kim, Y., and Rush, A. M. (2016). Sequence-Level Knowledge Distillation. EMNLP.
- Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS.
- Mirzadeh, S. I., et al. (2020). Improved Knowledge Distillation via Teacher Assistant. AAAI.
- Robertson, S., and Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval, 3(4).
- van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748.
- Flyingkiwi (2026). Flyingkiwi v3.1 technical report and answer-format controller. <https://flyingkiwi.yunusemrevurgun.com/technical/report/>