

Package: Rbebelm (via r-universe)

June 7, 2026

Title 'R' Bindings for 'BebeLM' 'CPU' Inference

Version 0.2.0-0.1.0

Description Provides experimental 'R' bindings for the pure-Rust 'BebeLM' 'CPU'-only language-model inference library <<https://github.com/maximecb/bebelm>>. The native layer is structured as runtime-selected Rust backends so portable 'R' binaries can avoid build-host-specific SIMD assumptions while still using optimized backends when available.

License GPL (>= 3)

URL <https://github.com/soukoku-bioinfo/Rbebelm>,
<https://soukoku-bioinfo.github.io/Rbebelm>

BugReports <https://github.com/soukoku-bioinfo/Rbebelm/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

SystemRequirements Rust toolchain with cargo, GNU make

Suggests tinytest, roxygen2, rmarkdown, knitr, pkgdown, jsonlite,
nanonext

VignetteBuilder knitr

Config/pak/sysreqs make libclang-dev

Repository <https://soukoku-bioinfo.r-universe.dev>

Date/Publication 2026-06-07 18:54:14 UTC

RemoteUrl <https://github.com/soukoku-bioinfo/Rbebelm>

RemoteRef HEAD

RemoteSha 81c19a82cfaebb5beabdad6c878b7b3c54e1f769

Contents

bebel_agent	3
bebel_agent_configure	3
bebel_agent_generate	4
bebel_agent_info	5
bebel_agent_run	5
bebel_agent_tool	6
bebel_agent_tool_catalog	7
bebel_append	7
bebel_append_system	8
bebel_append_tokens	8
bebel_append_tool_result	9
bebel_append_user	9
bebel_assistant_turn	10
bebel_chat	10
bebel_clear	11
bebel_console_event	12
bebel_default_r_tools	12
bebel_detokenize	13
bebel_event_handler	13
bebel_event_types	14
bebel_generate	14
bebel_history	15
bebel_live_console	16
bebel_model_load	17
bebel_parse_tool_call	17
bebel_r_agent	18
bebel_r_agent_clear	19
bebel_r_agent_console	19
bebel_r_agent_rpc_server	20
bebel_r_agent_start	20
bebel_r_agent_turn	21
bebel_token_ids	22
bebel_tokenize	23
bebel_tool	23
bebel_transcript	24
BebelAgent	24
BebelModel	25
print.bebelGeneration	25
rbebelm_backend_features	26
rbebelm_backend_info	26
rbebelm_cpuid_info	26
rbebelm_set_backend	27

bebel_agent *Create a persistent BebeLM agent*

Description

A BebeAgent owns an independent token transcript and decode cache while sharing the loaded model weights. This mirrors upstream bebelm::agent::Agent.

Usage

```
bebel_agent(  
  model,  
  greedy = FALSE,  
  max_gen = NULL,  
  max_context = NULL,  
  max_think = NULL,  
  temperature = NULL,  
  top_k = NULL,  
  repeat_penalty = NULL  
)
```

Arguments

model A BebeLM object.
greedy Use deterministic greedy decoding.
max_gen, max_context, max_think
 Optional generation limits.
temperature, top_k, repeat_penalty
 Optional sampling settings.

Value

A BebeAgent object.

bebel_agent_configure *Configure a BebeLM agent*

Description

Configure a BebeLM agent

Usage

```

bebel_agent_configure(
    agent,
    greedy = NULL,
    max_gen = NULL,
    max_context = NULL,
    max_think = NULL,
    temperature = NULL,
    top_k = NULL,
    repeat_penalty = NULL
)

```

Arguments

agent	A BebelAgent object.
greedy	Use deterministic greedy decoding.
max_gen, max_context, max_think	Optional generation limits.
temperature, top_k, repeat_penalty	Optional sampling settings.

Value

Updated agent info.

bebel_agent_generate *Generate a raw continuation from a BebeLM agent transcript*

Description

Generate a raw continuation from a BebeLM agent transcript

Usage

```

bebel_agent_generate(
    agent,
    on_event = bebel_console_event(),
    check_interrupt = TRUE
)

```

Arguments

agent	A BebelAgent object.
on_event	Event callback, named list of event-specific handlers, or NULL. Event types are bebel_event_types(). Delta events contain delta, id, and index; final events contain accumulated content or text. Use bebel_console_event() for live console output.

check_interrupt

Check for Ctrl-C during prefill and before every decoded token.

Value

A classed generation result.

bebel_agent_info *Inspect a BebeLM agent*

Description

Inspect a BebeLM agent

Usage

```
bebel_agent_info(agent)
```

Arguments

agent A BebeAgent object.

Value

Named list of state and configuration.

bebel_agent_run *Run a BebeLM agent with R tool dispatch*

Description

This is an Agent-first orchestration loop. It observes `tool_call_end` events, parses tool calls, invokes matching R tools with private context, appends tool results to the agent transcript, and continues generation.

Usage

```
bebel_agent_run(
  agent,
  tools = list(),
  context = new.env(parent = emptyenv()),
  hooks = list(),
  parse_tool_call = bebel_parse_tool_call,
  max_steps = 4,
  on_event = NULL,
  check_interrupt = TRUE
)
```

Arguments

agent	A BebelAgent object.
tools	A list of bebel_tool() objects or named functions.
context	Private run context passed to tools and hooks but not appended to the model transcript.
hooks	Optional named list of hooks: turn_start, event, tool_request, tool_result, tool_error, turn_end.
parse_tool_call	Function converting tool-call content to list(name, arguments, raw).
max_steps	Maximum assistant/tool iterations.
on_event	Optional event callback or handler list for model events.
check_interrupt	Check for Ctrl-C during generation.

Value

A bebelAgentRun list with turns, tool calls, and final agent info.

bebel_agent_tool	<i>Create an Rbebelm agent tool specification</i>
------------------	---

Description

This is a small metadata layer on top of [bebel_tool\(\)](#). It keeps a JSON-schema-like parameter specification next to the executable R function so the same tool catalog can be used by the console agent and the RPC surface.

Usage

```
bebel_agent_tool(name, description, params = list(), fun)
```

Arguments

name	Tool name.
description	Short description shown to the model and clients.
params	Named list of parameter specifications. Each entry may contain type, description, required, and enum.
fun	Function called as fun(args, context, call) or any subset of those names, following bebel_tool() conventions.

Value

A bebelAgentTool object.

`bebel_agent_tool_catalog`*Describe an Rbebelm agent tool catalog*

Description

Describe an Rbebelm agent tool catalog

Usage

```
bebel_agent_tool_catalog(tools)
```

Arguments

`tools` A tool list accepted by `bebel_r_agent()`.

Value

A data frame with tool names and descriptions.

`bebel_append`*Append raw text to a BebeLM agent transcript*

Description

Append raw text to a BebeLM agent transcript

Usage

```
bebel_append(agent, text)
```

Arguments

`agent` A BebelAgent object.

`text` Raw text to append.

Value

Invisibly returns agent.

bebel_append_system *Append a ChatML system turn to a BebeLM agent transcript*

Description

Appends `<|im_start|>system\n...<|im_end|>` framing. BebeLM upstream does not expose a separate system-prompt channel; this helper provides the ChatML system-role form for users who want to place an instruction before user turns.

Usage

```
bebel_append_system(agent, message)
```

Arguments

agent	A BebeLM agent object.
message	System instruction text.

Value

Invisibly returns agent.

bebel_append_tokens *Append token ids to a BebeLM agent transcript*

Description

Append token ids to a BebeLM agent transcript

Usage

```
bebel_append_tokens(agent, ids)
```

Arguments

agent	A BebeLM agent object.
ids	Integer token ids.

Value

Invisibly returns agent.

bebel_append_tool_result

Append a ChatML tool result turn to a BebeLM agent transcript

Description

Append a ChatML tool result turn to a BebeLM agent transcript

Usage

```
bebel_append_tool_result(agent, content)
```

Arguments

agent	A BebeLM agent object.
content	Tool result content to append.

Value

Invisibly returns agent.

bebel_append_user

Append a ChatML user turn to a BebeLM agent transcript

Description

Append a ChatML user turn to a BebeLM agent transcript

Usage

```
bebel_append_user(agent, message)
```

Arguments

agent	A BebeLM agent object.
message	User message.

Value

Invisibly returns agent.

bebel_assistant_turn *Generate and close an assistant ChatML turn from a BebeLM agent*

Description

Generate and close an assistant ChatML turn from a BebeLM agent

Usage

```
bebel_assistant_turn(  
    agent,  
    on_event = bebel_console_event(),  
    check_interrupt = TRUE  
)
```

Arguments

agent	A BebeAgent object.
on_event	Event callback, named list of event-specific handlers, or NULL. Event types are bebel_event_types(). Delta events contain delta, id, and index; final events contain accumulated content or text. Use bebel_console_event() for live console output.
check_interrupt	Check for Ctrl-C during prefill and before every decoded token.

Value

A classed generation result.

bebel_chat *Generate a single ChatML assistant reply*

Description

Generate a single ChatML assistant reply

Usage

```
bebel_chat(  
    model,  
    message,  
    greedy = FALSE,  
    on_event = bebel_console_event(),  
    check_interrupt = TRUE,  
    max_gen = NULL,
```

```

    max_context = NULL,
    max_think = NULL,
    temperature = NULL,
    top_k = NULL,
    repeat_penalty = NULL
)

```

Arguments

model	A BebelModel object.
message	User message.
greedy	Use deterministic greedy decoding.
on_event	Event callback, named list of event-specific handlers, or NULL. Event types are bebel_event_types(). Delta events contain delta, id, and index; final events contain accumulated content or text. Use bebel_console_event() for live console output.
check_interrupt	Check for Ctrl-C during prefill and before every decoded token.
max_gen, max_context, max_think	Optional generation limits.
temperature, top_k, repeat_penalty	Optional sampling settings.

bebel_clear	<i>Clear a BebeLM agent transcript and caches</i>
-------------	---

Description

Clears the conversation state while keeping the loaded model weights and the agent's generation configuration. This is the helper form of agent\$clear().

Usage

```
bebel_clear(agent)
```

Arguments

agent	A BebelAgent object.
-------	----------------------

Value

Updated agent info.

bebel_console_event *Console event handler for generated text and thinking*

Description

Returns an event handler suitable for `on_event`. Thinking blocks are printed with `<think>` markers, text deltas are printed as they arrive, and done events add a trailing newline.

Usage

```
bebel_console_event()
```

Value

A function accepting one generation event list.

bebel_default_r_tools *Built-in R session tools for the Rbebelm agent layer*

Description

The default catalog is intentionally small. It exposes read-only file and R session inspection tools plus optional R evaluation and plot rendering. These are ordinary R functions and run in the current R process.

Usage

```
bebel_default_r_tools(
  env = .GlobalEnv,
  cwd = getwd(),
  allow_eval = FALSE,
  max_chars = 4000L
)
```

Arguments

<code>env</code>	Environment used by <code>r_objects</code> , <code>r_eval</code> , and <code>r_plot</code> .
<code>cwd</code>	Working directory for file and plot tools.
<code>allow_eval</code>	Whether to include the <code>r_eval</code> and <code>r_plot</code> tools. If <code>FALSE</code> , code-evaluation tools are not advertised to the model.
<code>max_chars</code>	Maximum characters returned from a single tool.

Value

A named list of `bebelAgentTool` objects.

bebel_detokenize	<i>Decode BebeLM token ids</i>
------------------	--------------------------------

Description

Decode BebeLM token ids

Usage

```
bebel_detokenize(model, ids)
```

Arguments

model	A BebeLM object.
ids	Integer token ids.

Value

Decoded text.

bebel_event_handler	<i>Build a BebeLM generation event handler</i>
---------------------	--

Description

bebel_event_handler() creates a single on_event callback from handlers for individual event types. Current event types are returned by bebel_event_types().

Usage

```
bebel_event_handler(
  start = NULL,
  thinking_start = NULL,
  thinking_delta = NULL,
  thinking_end = NULL,
  text_start = NULL,
  text_delta = NULL,
  text_end = NULL,
  tool_list_start = NULL,
  tool_list_delta = NULL,
  tool_list_end = NULL,
  tool_call_start = NULL,
  tool_call_delta = NULL,
  tool_call_end = NULL,
  done = NULL,
  default = NULL
)
```

Arguments

start, thinking_start, thinking_delta, thinking_end, text_start, text_delta, text_end	Optional functions called for the corresponding stream event.
tool_list_start, tool_list_delta, tool_list_end	Optional handlers for BebeLM tool-list delimiter blocks.
tool_call_start, tool_call_delta, tool_call_end	Optional handlers for BebeLM tool-call delimiter blocks.
done	Function called for the final done event, or NULL.
default	Function called for events without a type-specific handler, or NULL.

Value

A function accepting one generation event list.

bebel_event_types	<i>Return BebeLM stream event types.</i>
-------------------	--

Description

Return BebeLM stream event types.

Usage

```
bebel_event_types()
```

bebel_generate	<i>Generate a raw continuation from a prompt</i>
----------------	--

Description

Generate a raw continuation from a prompt

Usage

```
bebel_generate(  
  model,  
  prompt,  
  greedy = FALSE,  
  on_event = bebel_console_event(),  
  check_interrupt = TRUE,  
  max_gen = NULL,  
  max_context = NULL,  
  max_think = NULL,
```

```

    temperature = NULL,
    top_k = NULL,
    repeat_penalty = NULL
)

```

Arguments

model A BebelModel object.

prompt Prompt text.

greedy Use deterministic greedy decoding.

on_event Event callback, named list of event-specific handlers, or NULL. Event types are bebel_event_types(). Delta events contain delta, id, and index; final events contain accumulated content or text. Use bebel_console_event() for live console output.

check_interrupt Check for Ctrl-C during prefill and before every decoded token.

max_gen, max_context, max_think Optional generation limits.

temperature, top_k, repeat_penalty Optional sampling settings.

Value

A classed list with generated text, token ids, stop reason, and timing statistics.

bebel_history	<i>Return a BebeLM agent token transcript</i>
---------------	---

Description

Returns the full token transcript currently held by the agent. This is the helper form of agent\$history().

Usage

```
bebel_history(agent)
```

Arguments

agent A BebelAgent object.

Value

Integer token ids.

bebel_live_console *Live terminal console for BebeLM chats*

Description

Start an interactive terminal chat loop. If `x` is a `BebeLMModel`, a new `BebeLMAgent` is created. If `x` is a `BebeLMAgent`, its existing transcript and caches are reused. Type `/quit` or `/exit` to leave the loop.

Usage

```
bebel_live_console(  
  x,  
  prompt = ">>> ",  
  exit_commands = c("/quit", "/exit"),  
  on_event = bebel_console_event(),  
  check_interrupt = TRUE,  
  greedy = FALSE,  
  max_gen = NULL,  
  max_context = NULL,  
  max_think = NULL,  
  temperature = NULL,  
  top_k = NULL,  
  repeat_penalty = NULL  
)
```

Arguments

<code>x</code>	A <code>BebeLMModel</code> or <code>BebeLMAgent</code> .
<code>prompt</code>	Prompt displayed before reading each user message.
<code>exit_commands</code>	Character vector of commands that exit the console.
<code>on_event</code>	Event handler used for assistant output.
<code>check_interrupt</code>	Check for Ctrl-C during generation.
<code>greedy</code>	Use deterministic greedy decoding.
<code>max_gen, max_context, max_think</code>	Optional generation limits.
<code>temperature, top_k, repeat_penalty</code>	Optional sampling settings.

Value

Invisibly returns the `BebeLMAgent` used by the console.

bebel_model_load	<i>Load a BebeLM GGUF model</i>
------------------	---------------------------------

Description

Load a BebeLM GGUF model

Usage

```
bebel_model_load(path, num_threads = NULL)
```

Arguments

path	Path to the GGUF weights file.
num_threads	Optional Rayon global thread-pool size. This can only be set once per R process.

Value

A BebelModel object.

bebel_parse_tool_call	<i>Parse a BebeLM tool call block</i>
-----------------------	---------------------------------------

Description

The default parser accepts JSON objects such as `{"name": "tool", "arguments": {...}`, simple `name({...})` calls, and bracketed BebeLM calls such as `[name(key="value")]`. Pass a custom parser to `bebel_agent_run()` for model- or prompt-specific formats.

Usage

```
bebel_parse_tool_call(content)
```

Arguments

content	Accumulated content between BebeLM tool-call delimiters.
---------	--

Value

A list with name, arguments, and raw.

bebel_r_agent	<i>Create an R-native Rbebelm agent session</i>
---------------	---

Description

`bebel_r_agent()` is a higher-level layer inspired by R console agent harnesses. It keeps one BebelLM agent, a private tool context, and a small R tool catalog together so the same object can be driven by a console loop or by the JSON-RPC server.

Usage

```
bebel_r_agent(
  model,
  system_prompt = NULL,
  tools = NULL,
  env = .GlobalEnv,
  cwd = getwd(),
  allow_eval = FALSE,
  prompt_style = c("compact", "full"),
  greedy = FALSE,
  max_gen = 512,
  max_context = 4096,
  max_think = 64,
  temperature = 0.8,
  top_k = 50,
  repeat_penalty = 1.1
)
```

Arguments

<code>model</code>	A BebelModel object.
<code>system_prompt</code>	System prompt. NULL builds a default prompt including the tool catalog.
<code>tools</code>	Tool catalog. Defaults to <code>bebel_default_r_tools()</code> .
<code>env</code>	Environment exposed to R tools.
<code>cwd</code>	Working directory for file tools.
<code>allow_eval</code>	Whether to include an <code>r_eval</code> tool that executes code.
<code>prompt_style</code>	Tool prompt verbosity. "compact" is faster for console use; "full" includes descriptions for every argument.
<code>greedy</code> , <code>max_gen</code> , <code>max_context</code> , <code>max_think</code> , <code>temperature</code> , <code>top_k</code> , <code>repeat_penalty</code>	Generation options passed to <code>bebel_agent()</code> .

Value

A `bebelRAgent` environment.

bebel_r_agent_clear *Clear an Rbebelm R agent session*

Description

Clear an Rbebelm R agent session

Usage

```
bebel_r_agent_clear(session)
```

Arguments

session A bebelRAgent.

Value

Invisibly returns session.

bebel_r_agent_console *Start an interactive Rbebelm console agent*

Description

Start an interactive Rbebelm console agent

Usage

```
bebel_r_agent_console(
  session,
  prompt = "bebel> ",
  max_steps = 4L,
  show_stats = TRUE,
  blank_limit = 10L
)
```

Arguments

session A bebelRAgent.
prompt Prompt string.
max_steps Maximum assistant/tool iterations per user prompt.
show_stats Whether to print token/timing stats after each turn.
blank_limit Number of consecutive blank inputs before exiting. Set to Inf to never auto-exit on blanks.

Value

Invisibly returns session.

bebel_r_agent_rpc_server

Serve an Rbebelm R agent over JSON-RPC

Description

This optional SDK surface uses nanonext to expose the same bebelRAgent object used by the console. It is intentionally small and not an OpenAI API: clients call JSON-RPC methods such as `turn`, `tools/list`, and `session/transcript`.

Usage

```
bebel_r_agent_rpc_server(session, url = "http://127.0.0.1:8080")
```

Arguments

session	A bebelRAgent.
url	URL to listen on, e.g. "http://127.0.0.1:8080".

Value

A nanoServer object from nanonext.

bebel_r_agent_start

Launch an R-native Rbebelm console from weights

Description

Convenience wrapper for loading a model, creating a `bebel_r_agent()`, and entering `bebel_r_agent_console()`. This keeps the loaded model object local to the launcher while the agent tools, `/r`, and `/rplot` commands share env.

Usage

```
bebel_r_agent_start(
  weights = Sys.getenv("BEBELM_WEIGHTS_FILE", "LFM2.5-8B-A1B-Q4_K_M.gguf"),
  num_threads = as.numeric(Sys.getenv("BEBELM_NUM_THREADS", "2")),
  env = .GlobalEnv,
  cwd = getwd(),
  allow_eval = TRUE,
  greedy = TRUE,
  max_gen = as.numeric(Sys.getenv("BEBELM_AGENT_MAX_GEN", "256")),
```

```

max_context = 4096,
max_think = as.numeric(Sys.getenv("BEBELM_AGENT_MAX_THINK", "48")),
temperature = 0.8,
top_k = 50,
repeat_penalty = 1.1,
prompt = "bebel> ",
max_steps = 4L,
show_stats = TRUE,
blank_limit = 10L,
prompt_style = c("compact", "full")
)

```

Arguments

weights	GGUF weights file. Defaults to BEBELM_WEIGHTS_FILE, then "LFM2.5-8B-A1B-Q4_K_M.gguf" in the working directory.
num_threads	Optional Rayon thread count passed to <code>bebel_model_load()</code> .
env	Environment shared by <code>/r</code> , <code>/rplot</code> , <code>r_objects</code> , and optional code-evaluation tools.
cwd	Working directory for file tools and <code>/rplot</code> output.
allow_eval	Whether to include <code>r_eval</code> and <code>r_plot</code> tools that the model can call.
greedy, max_gen, max_context, max_think, temperature, top_k, repeat_penalty	Generation options passed to <code>bebel_r_agent()</code> .
prompt	Prompt string for <code>bebel_r_agent_console()</code> .
max_steps	Maximum assistant/tool iterations per user prompt.
show_stats	Whether to print token/timing stats after each turn.
blank_limit	Number of consecutive blank inputs before exiting the console. Set to Inf to never auto-exit on blanks.
prompt_style	Tool prompt verbosity passed to <code>bebel_r_agent()</code> .

Value

Invisibly returns the bebelRAgent session after the console exits.

bebel_r_agent_turn	<i>Run one user turn through an Rbebelm R agent</i>
--------------------	---

Description

Run one user turn through an Rbebelm R agent

Usage

```

bebel_r_agent_turn(
    session,
    prompt,
    max_steps = 4L,
    on_event = NULL,
    hooks = list(),
    check_interrupt = TRUE
)

```

Arguments

session	A bebelRAgent from <code>bebel_r_agent()</code> .
prompt	User prompt.
max_steps	Maximum assistant/tool iterations.
on_event	Optional BebeLM event callback.
hooks	Optional hooks passed to <code>bebel_agent_run()</code> .
check_interrupt	Check for Ctrl-C during generation.

Value

A bebelRAgentTurn list.

bebel_token_ids	<i>Return BebeLM tokenizer special token ids.</i>
-----------------	---

Description

Return BebeLM tokenizer special token ids.

Usage

```
bebel_token_ids()
```

bebel_tokenize	<i>Tokenize text with a BebeLM model tokenizer</i>
----------------	--

Description

Tokenize text with a BebeLM model tokenizer

Usage

```
bebel_tokenize(model, text, add_bos = TRUE)
```

Arguments

model	A BebeLM object.
text	Text to encode.
add_bos	Whether to prepend the BOS token.

Value

Integer token ids.

bebel_tool	<i>Define a BebeLM R tool</i>
------------	-------------------------------

Description

Define a BebeLM R tool

Usage

```
bebel_tool(name, fun, description = NULL, schema = NULL)
```

Arguments

name	Tool name exposed to the tool dispatcher.
fun	Function to run. It is called as <code>fun(args = ..., context = ..., call = ...)</code> when it accepts those names, otherwise with progressively simpler fallbacks.
description	Optional human-readable description.
schema	Optional schema/metadata object for prompts or adapters.

Value

A `bebelTool` object.

bebel_transcript	<i>Decode a BebeLM agent transcript</i>
------------------	---

Description

Decodes the agent's full token transcript. This is the helper form of `agent$transcript()`.

Usage

```
bebel_transcript(agent)
```

Arguments

agent	A BebelAgent object.
-------	----------------------

Value

Transcript text.

BebelAgent	<i>Persistent BebeLM conversation agent with transcript and decode caches.</i>
------------	--

Description

Persistent BebeLM conversation agent with transcript and decode caches.

Usage

```
BebelAgent
```

Format

An object of class `Rbebelm::BebelAgent__bundle` (inherits from `savvy_Rbebelm__sealed`) of length 1.

BebelModel	<i>Loaded BebeLM GGUF model.</i>
------------	----------------------------------

Description

Loaded BebeLM GGUF model.

Usage

BebelModel

Format

An object of class `Rbebelm::BebelModel__bundle` (inherits from `savvy_Rbebelm__sealed`) of length 1.

`print.bebelGeneration` *Print a BebeLM generation result*

Description

Print a BebeLM generation result

Usage

```
## S3 method for class 'bebelGeneration'  
print(x, ...)
```

Arguments

<code>x</code>	A result returned by <code>bebel_generate()</code> or <code>bebel_chat()</code> .
<code>...</code>	Unused.

Value

Invisibly returns `x`.

rbebelm_backend_features

Return feature information reported by the loaded Rust backend.

Description

Return feature information reported by the loaded Rust backend.

Usage

rbebelm_backend_features()

rbebelm_backend_info *Inspect Rbebelm backend dispatch state*

Description

Inspect Rbebelm backend dispatch state

Usage

rbebelm_backend_info()

Value

A named list describing installed, supported, requested, and selected backends, with class rbebelmBackendInfo.

rbebelm_cpuid_info *Inspect CPU SIMD support used by backend dispatch*

Description

Inspect CPU SIMD support used by backend dispatch

Usage

rbebelm_cpuid_info()

Value

A named list of logical CPU feature checks with class rbebelmCpuidInfo.

rbebelm_set_backend *Select the Rbebelm native backend*

Description

Must be called before loading a model or querying backend features.

Usage

```
rbebelm_set_backend(backend = "auto")
```

Arguments

backend One of "auto", "scalar", "avx2", "avx512", or "neon".

Value

The requested backend name.

Index

* datasets

- BebelAgent, 24
- BebelModel, 25

bebel_agent, 3

bebel_agent(), 18

bebel_agent_configure, 3

bebel_agent_generate, 4

bebel_agent_info, 5

bebel_agent_run, 5

bebel_agent_run(), 22

bebel_agent_tool, 6

bebel_agent_tool_catalog, 7

bebel_append, 7

bebel_append_system, 8

bebel_append_tokens, 8

bebel_append_tool_result, 9

bebel_append_user, 9

bebel_assistant_turn, 10

bebel_chat, 10

bebel_chat(), 25

bebel_clear, 11

bebel_console_event, 12

bebel_default_r_tools, 12

bebel_default_r_tools(), 18

bebel_detokenize, 13

bebel_event_handler, 13

bebel_event_types, 14

bebel_generate, 14

bebel_generate(), 25

bebel_history, 15

bebel_live_console, 16

bebel_model_load, 17

bebel_model_load(), 21

bebel_parse_tool_call, 17

bebel_r_agent, 18

bebel_r_agent(), 7, 20–22

bebel_r_agent_clear, 19

bebel_r_agent_console, 19

bebel_r_agent_console(), 20, 21

bebel_r_agent_rpc_server, 20

bebel_r_agent_start, 20

bebel_r_agent_turn, 21

bebel_token_ids, 22

bebel_tokenize, 23

bebel_tool, 23

bebel_tool(), 6

bebel_transcript, 24

BebelAgent, 24

BebelModel, 25

print.bebelGeneration, 25

rbebelm_backend_features, 26

rbebelm_backend_info, 26

rbebelm_cpuid_info, 26

rbebelm_set_backend, 27