servicenow

ServiceNow Small Language Model (SLM) Model Card (v2.0)

ServiceNow Small Language Model (SLM) Model Card (v2.0)

Intended Use and Functionality

Purpose of the Model

The Model is designed to support enterprise AI applications by enhancing text-based automation and content generation within ServiceNow workflows. It is optimized for text-to-text processing, code generation, workflow generation (text-to-flow), summarization, question answering, query handling, and agentic workflows—ensuring alignment with ServiceNow-specific enterprise use cases.

This model is engineered to deliver advanced reasoning and automation capabilities while remaining deployable in constrained environments such as on-prem or air-gapped enterprise settings.

- **Enterprise Optimization:** Tailored for business process automation, ensuring generated content meets ServiceNow platform requirements.
- **Workflow Integration:** Supports AI-driven text generation for enterprise applications, knowledge retrieval, and automated reporting.
- **Middle-Tier Model Design:** Fits within 40–80 GB of memory, enabling deployment on a single high-end GPU while maintaining performance parity with larger baselines.
- **Latency-Performance Balance:** Optimized for real-time use cases like retrieval-augmented generation (RAG) and coding tasks.
- **Advanced Reasoning:** Enhanced support for complex tasks such as agentic workflows, multi-step reasoning, and domain-specific tool invocation.

Model Name

Apriel 13B

Model Version

v2.0

Model Release Date

September 10th, 2025

Union Market Release

Not enough information

Model Distribution Method

ServiceNow Platform

Model License

Apache 2.0

Products Using this Model

Now Assist skills, agents, and agentic workflows

Model Dependencies

Based on/derived from **Mistral-Nemo-Base-2407 (12B)**; upscaled to 13B via layer duplication

User Benefits

The Model is fine-tuned for **enterprise natural language processing (NLP) tasks**, enabling **efficient automation and text generation** within **ServiceNow workflows**. The model supports various AI-driven capabilities, designed to enhance **user productivity and enterprise automation.**

- **Instruction Adherence and Response Generation:** Helps ensure that AI-generated outputs align with business logic, enterprise standards, and workflow requirements.
- **Summarization and Knowledge Retrieval:** Optimized for content summarization, enterprise search, and knowledge base (KB) generation, enabling faster access to critical information.
- **Question Answering (Q&A):** Enhances enterprise query resolution, allowing users to retrieve relevant information based on structured prompts.
- **Intent Recognition and Classification:** Supports text understanding and automation, improving response accuracy in AI-driven workflows.
- **Code and Query Generation:** Fine-tuned for Text-to-Code and Text-to-Cypher tasks, assisting users in automating programming and database interactions.
- **Workflow Generation (Text-to-Flow):** Automatically generate multi-step flows in Workflow Studio with configured triggers, actions, data pill values, and text instructions, following standard design patterns.
- **Agentic workflows:** Fine-tuned for single-turn agentic scenarios that require the invocation of one or more tools.

By leveraging these capabilities, users benefit from **improved efficiency, streamlined workflows, and AI-assisted content creation while maintaining control over AI-generated outputs.**

Risks

As with any language model, the potential outputs cannot be predicted in advance due to the probabilistic nature of generative AI. The model can produce inaccurate, biased, or otherwise objectionable responses to user prompts.

Factors and limitations

Optimization Scope:

- The Model is a general-purpose AI system optimized for a broad range of ServiceNow applications, including Text-to-Code, Text-to-Cypher, Content Moderation, Workflow Generation (Text-to-Flow), and agent-related use cases.
- It consolidates multiple functionalities into a singular high-performance architecture, reducing complexity while improving efficiency.

Input Requirements:

- The Model relies on **structured, well-defined prompts** to generate high-quality outputs.
- **Ambiguous, incomplete, or overly generic prompts may lead to misinterpretations, incorrect results, or suboptimal performance** in text generation and code-related use cases.
- For **code generation,** the Model performs best when provided **clear problem statements, sufficient context, and examples of expected output.**

Data Scope:

- The Model is primarily trained & fine-tuned on a combination of **open-source data, ServiceNow platform-specific datasets, and a curated selection of mathematical, multilingual, reasoning, and instruction-following tokens.**

Domain-Specific Challenges:

- The Model is **not explicitly designed for niche domains requiring specialized technical knowledge.**

Ethical considerations

The Model has been fine-tuned with the intention of reducing bias, toxicity, and hallucinations, although such limitations may still exist due to the probabilistic nature of generative AI.

Text LLMs can produce harmful text based on how they are prompted, and the Model is not free from such limitations, consistent with other industry LLMs. When using the model customers should follow ServiceNow's guidelines on intended use available on docs.servicenow.com as well as ServiceNow's AI Acceptable Use Policy.

Please report instances of unintended hallucinations, harmful text (e.g. toxicity, profanity, etc.), or unexpected data occurrences in the model output so that we can evaluate for remediation.

Supported Languages

The Model was trained on a diverse multilingual dataset to help ensure strong language proficiency, text generation, and enterprise automation support. The dataset includes extensive linguistic coverage, with a focus on enterprise-relevant languages.

Multilingual Coverage:

- Supports a broad range of P1 and P2 languages (English, French, Dutch, German, Spanish, Italian, Brazilian Portuguese, Portuguese, French Canadian, Japanese), helping to ensure effective AI-driven text generation across multiple linguistic contexts)
- Includes specialized datasets to enhance fluency and contextual accuracy, particularly in business and enterprise interactions.

English Proficiency:

- Optimized for enterprise text processing tasks, including summarization, question answering, content generation, and instruction-following
- Trained on text datasets to help ensure coherence, factual accuracy, and response generation.

Supported Coding Languages:

- **JavaScript (GlideScript):** Optimized for ServiceNow Platform development and code generation tasks.
- **Python:** Supported for general-purpose scripting and automation, leveraging task-specific coding datasets.
- **SQL (Cypher Query Language):** Integrated for Text-to-Cypher tasks, enabling query generation and database interaction.

Model Architecture

The Model is a **transformer-based dense language model** with 13 billion parameters, structured with the following key attributes:

- **Number of Layers:** 44
- **Model Dimension:** 5,120
- **Head Dimension:** 128
- **Hidden Dimension:** 14,336
- **Activation Function:** SwiGLU
- **Attention Heads:** 32 (Grouped Query Attention)
- **Key-Value Heads:** 8
- **RoPE Theta:** 1,000,000
- **Vocabulary Size:** ~131,000 tokens
- **Tokenizer:** The model utilizes the Tekken tokenizer, an advanced iteration of Tiktoken, optimized for over 100 languages. It achieves 30% greater compression for source code and underrepresented languages, with notable efficiency gains in Korean and Arabic.

It is designed as a general-purpose AI system optimized for a broad range of ServiceNow applications, including Text-to-Text, Text-to-Code, Workflow Generation (Text-to-Flow), Text-to-Cypher, and Content Moderation tasks. The model consolidates multiple functionalities into a single, high-performance architecture, reducing system complexity while enhancing efficiency

Training Process Key Components

- **Model Upscaling:** Start from an open-source backbone and expand capacity via **depth-upscaling** (layer duplication) to reach a sweet spot for enterprise reasoning, while keeping single-GPU deployability in mind. Width-upscaling was explored but deferred due to training instability.
- **Continual Pretraining (CPT) Engine:** Strengthens reasoning with a mix of reasoning, chain-of-thought, and replay data. Uses no chat template; sequences concatenate input → intermediate steps → target with full-sequence loss. Training uses packed long-context sequences with cross-document attention masked and checkpoint averaging to stabilize the handoff. Effectiveness is checked with standard LM benchmarks and a lightweight downstream SFT probe.
- **Supervised Fine-Tuning (SFT) Aligner:** Aligns the model to target behaviors across enterprise and generic tasks, sharpening prompt interpretation, structured output discipline, conversational quality, and instruction following, balancing both reasoning and non-reasoning modes.
- **Reinforcement Learning (RL) with GRPO:** Improves robustness and compliance using rule/LLM-as-judge rewards. Enforces a tagged output structure (reasoning trace + final answer), covers single-turn tool use, agent-assist tasks judged by an external LLM, instruction-following with verifiable constraints, and coding scored by test success. Includes reasoning-mode controls that deliberately mix reasoning and no-reasoning rollouts.

Number of Parameters

13 billion

Maximum Input and Output Size

- **Maximum Input Size + Output Size:** 64k
- **Shared Context Window:** 32,768 tokens

Input and Output Modalities

Modality Type

Single-modality: The Model processes and generates **text-to-text** outputs.

Inputs

Input Type: The Model accepts **plain text** input, typically structured as questions, commands, or prompts in natural language (e.g., "Summarize this article" or "Translate this sentence to French").

Input Constraints:

- The Model is designed primarily for natural language inputs and may perform sub-optimally with **non-text inputs** such as raw numerical data or unstructured code.
- Some **long-form inputs** may be truncated based on token limits, affecting response completeness.
- The Model may struggle with **highly ambiguous or domain-specific jargon** if not adequately trained in those areas.

Outputs

Output Type: The Model generates **plain text responses** based on the input prompt. This can include answers to questions, summarizations, translations, code snippets, or structured text.
Output Constraints: Outputs are constrained by **token length limits**, which may truncate longer responses.

Input and Output Formats

- **Input Format:** Inputs must be formatted as **plain text** with no additional structuring required.
- **Output Format:** Outputs are generated as **plain text** with no additional structuring unless explicitly prompted.

Training & Fine-Tuning Data

The training of the Model followed a staged approach, with each phase using carefully designed data mixtures to progressively enhance reasoning, coding, and enterprise alignment capabilities.

Upscaling Dataset

During the model upscaling phase, the 12B backbone was expanded to 13B parameters and trained on 100B tokens from a balanced open-source mix. This corpus included high-quality web content, scientific and technical literature, reference works, programming code, and mathematical problem sets (e.g., coding data across multiple languages, StackExchange, and math datasets). This broad replay corpus ensured the model gained general reasoning and domain coverage.

Continual Pre-Training (CPT) Data

To strengthen the base model's reasoning, we continually pre-trained on a diverse mix spanning math, science, coding, and instruction-following, supplemented with Chain-of-Thought (CoT) and replay-style pretraining data.

- **Mixture:** 60% reasoning, 25% CoT, 15% pretraining-style.
- **Formatting:** No chat template; reasoning and CoT samples concatenated as newline-delimited input → intermediate steps → target; loss on all tokens.
- **Tokens & Batching:** 68B tokens, batch size 768; sequences packed to 16k with cross-document attention masked.
- **Optimization:** AdamW (weight decay 0.1); base LR 5e-5 cosine-decayed to 5e-6; 10% linear warmup.
- **Checkpointing:** Average 3 equally spaced CPT checkpoints for the handoff to the next stage.
- **Evaluation:** (1) LM Evaluation Harness benchmarks; (2) downstream reasoning after SFT on 15k reasoning samples (3 epochs, batch 128, max seq 16k, LR 1e-5 cosine→0, 10% warmup, no packing).

Supervised Fine-Tuning (SFT) Data

Following upscaling and continual pre-training (CPT), we fine-tuned the model on ~2.7M high-quality samples to align it as a full-fledged reasoner; ~1.6M of these include explicit reasoning traces to balance performance across reasoning and non-reasoning tasks.

- **Tool Use:** Open-source + ServiceNow-synthetic data to strengthen single-turn tool invocation within the agentic framework.
- **Flow Generation:** Custom datasets for ServiceNow Flow Designer covering (i) flow outline generation and (ii) input generation for flow components from natural-language instructions.
- **Coding:** Curated Python, JavaScript, and ServiceNow scripting data; pipelines seeded from OSS repos/docs to improve code generation, editing, and autocomplete.
- **Complex Instruction Following:** High-quality synthetic tasks (e.g., word problems, code gen) to enforce multi-step, compositional adherence.
- **Complex JSON Schema:** Specialized corpora for precise structured output (JSON) for enterprise use cases.
- **Cypher (Neo4j):** Synthetic data spanning schema variants (JSON, YAML, plain text, Neo4j) and diverse prompts to increase robustness and instruction-following accuracy.
- **Agent Assist:** Synthetic ServiceNow-targeted data (cases, chats, resolution notes, knowledge articles) to build deep comprehension and multilingual capability (P1/P2 priority languages), augmented with M2Lingual research outputs.

This SFT stage sharpens prompt interpretation, conversational quality, structured output reliability, and instruction-following across ServiceNow-targeted and generic domains.

Reinforcement Learning (RL) Data

For GRPO-based reinforcement learning, we used curated task categories with rule-based/LLM-as-judge, verifiable rewards:

- **Output Format:** Responses must include a reasoning trace and a final answer within predefined tags; tag compliance is verified first.
- **Agentic Ability:** Single-turn agentic scenarios (generic and ServiceNow-targeted) requiring correct invocation of one or more tools.
- **Agent Assist:** Case/chat summarization, resolution-note generation, and knowledge-article authoring scored by an external LLM for fidelity and usefulness.
- **Instruction Following:** Verifiable compositional instructions constraining content, format, length, and structure.
- **Coding:** Python and JavaScript tasks evaluated by multiple test cases; reward scales with the percentage of tests passed.

Summary of Training Content

The model was trained and evaluated exclusively on text-based data derived from publicly available and synthetic sources.

Publicly Available Datasets:

Text-only datasets obtained from open-source and internal collections, including materials distributed under open or permissive licenses and proprietary components utilized in accordance with the governing terms of each respective source.

Synthetic Data:

- Text-based synthetic data were generated and used for model training and fine-tuning.
- No commercially licensed datasets, other private datasets, crawled or scraped data, or user data were used in connection with model training.
- All training and evaluation data are text-only, with a multilingual focus. The dataset includes multiple languages as detailed in the section titled “Supported Languages.” No underrepresented or low-resource languages are included.
- The most recent data incorporated into the training corpus was acquired on **August 1, 2025**. The model does not undergo continuous or ongoing training.

As part of our Data Processing and Compliance Measures, we adhere to the text and data mining opt-out frameworks, comply with Robots.txt protocols and do not engage in web crawling or automated data collection for training. During dataset preparation, LLM “Judge” models assess content quality and policy compliance, automatically removing any material flagged as illegal or prohibited to ensure all training data meets rigorous ethical and legal standards. ServiceNow is a signatory to the EU’s GPAI Code of Practice.

Evaluation Data

In this section, we provide a comprehensive evaluation of the Model using a variety of academic, enterprise and ServiceNow specific use-case benchmarks. We describe the metrics that each benchmark measures across language understanding, reasoning, task-specific performance in enterprise scenarios, and agentic benchmarks.

We present comparative results that highlight the strengths and weaknesses of our models relative to state-of-the-art alternatives across different model sizes.

Metrics

We evaluate across instruction following, multi-turn conversation, ServiceNow agent-assist tasks, Japanese fluency, and Text2Code.

Core Metrics Tracked

IFEval

- Evaluate LLMs' ability to follow instructions using verifiable prompts
- IF Eval scores range from 0-1, where higher scores indicate better performance.

MultiChallenge

- Complex multi-turn reasoning

MT-Bench

- Evaluates LLMs' ability to engage in coherent, informative, and engaging conversations.
- MT-Bench scores range from 1 to 10, where higher scores indicate better performance.

ServiceNow Agent-Assist (Overall, Case/Chat Summarization, KB Gen, Resolution Notes)

- Scores range from 0-1, where higher scores indicate better performance.
- The final score reflects both faithfulness and completeness of the model's response. It is calculated as the harmonic mean of hallucination and completeness scores, so high performance requires strength in both areas.

Text2Code

MBPP (Python/JS)

- MBPP (Mostly Basic Programming Problems) is a benchmark dataset used to evaluate the code-generation capabilities of large language models (LLMs).
- The metric derived from this benchmark, often called Pass@k, measures the model's ability to produce functionally correct code from a natural language prompt.

Glide Coding Benchmarks

- Code Autocomplete
- Code Edit
- Glide JS Code Gen pass@1
- Generic JS Code Gen pass@1

Text2Flow

- Quantify alignment between a generated workflow and a designated ground truth by computing the complement of their tree edit distance.
- Evaluation dataset is a set of 108 records, created by subject matter experts and developers
- Scores represent accuracy and range from 0-100, where higher scores indicate better performance.

Text2Cypher

- Evaluates LLM performance on a real-world test HR dataset (124 relevant questions, 70 irrelevant).
- Scores represent accuracy and range from 0-100, where higher scores indicate better performance.

Benchmarks

Academic & Instruction-Following

Instruction Following

We report results on standard instruction-following benchmarks that assess model compliance with explicit constraints and user directives.

IFEval: A benchmark of~500 prompts with automatically verifiable constraints (e.g., word counts, formatting requirements, keyword usage) that measures whether models follow explicit instructions, emphasizing strict compliance rather than just semantic plausibility.

MultiChallenge: A multi-turn conversational benchmark (up to 10 turns) combining instruction retention, inference memory, versioned editing, and self- coherence. It evaluates whether models can maintain context, satisfy evolving instructions, and remain consistent across dialogue turns.

Model	IFEval	MultiChallenge	Average
SLM May 2025	76.52	13.92	45.22
Apriel 13B	***80.41***	***16.12***	***48.26***
OpenAI GPT-4.1 Mini	82.07	33.70	57.88

MT-Bench (Overall and by Language)

MT-Bench tests the ability of LLMs to engage in coherent, informative, and engaging conversations. The results in the table below show an improvement across all languages between the May2025 SLM and the Apriel 13B model.

Model	Overall	English	German	French	Can. French	Italian	Dutch	Portuguese	Japanese	Spanish
SLM May 2025	7.24	7.42	7.38	7.37	7.47	7.41	6.89	7.18	6.69	7.38
Apriel 13B	***7.93***	***7.84***	***8.13***	***7.80***	***8.06***	***7.91***	***7.93***	***8.13***	***7.58***	***7.98***

ServiceNow Use-Case Benchmarks

Agent Assist

ServiceNow Agent Assist leverages generative AI to enhance the efficiency of customer support agents by automating several key tasks. For example, Case Summarization condenses detailed customer support cases into clear summaries that help agents quickly grasp the issue, while Chat Summarization distills conversations between agents and customers—or virtual agents and customers—into concise overviews. Additionally, Knowledge Base Generation aids agents in converting case details into comprehensive knowledge articles, and Resolution Notes Generation assists in drafting final resolution steps from support interactions. Benchmarks for these use cases focus on evaluating the models' ability to generate outputs that are both faithful to the source information and complete in capturing all critical details.

Model	Overall	Case Summarization	Chat Summarization	KB Gen	Resolution Notes
Apriel 13B	***0.942***	***0.961***	***0.964***	***0.922***	***0.922***
SLM May 2025	0.902	0.903	0.928	0.880	0.897
OpenAI/gpt-oss 20B (Low)	0.931	0.923	0.969	0.923	0.908
OpenAI/gpt-oss 120B (Low)	0.951	0.962	0.962	0.930	0.952
OpenAI GPT-4.1 Mini	0.963	0.961	0.989	0.939	0.963

Japanese Fluency

Japanese fluency benchmark results highlight performance in following aspects of Japanese language:

- **Acceptable Rate:** The sentences are good across all the metrics for Japanese
- **Complete Sentences:** The sentences are complete and formal
- **Acceptable Loanwords:** Casual terms from English, but still part of Japanese language. For example, (chiketto) - This comes from the English word "ticket" and is commonly used for event tickets, concert tickets, and travel tickets
- **Acceptable Noun Phrase:** Noun phrases should be before verbs
- **Correct Name Address:** Surnames should be followed by honorifics. Also, surnames should be used and not first names.

Model	Acceptable Rate	Acceptable Loanword	Complete Sentence	Acceptable Noun Phrase	Correct Name Address
Apriel 13B	***0.255***	***0.335***	***0.745***	***0.925***	***0.740***
SLM May 2025	0.250	0.370	0.715	0.935	0.510
OpenAI/gpt-oss 20B (Low)	0.090	0.385	0.380	0.735	0.483
OpenAI/gpt-oss 120B (Low)	0.085	0.400	0.360	0.860	0.353
OpenAI GPT-4.1 Mini	0.205	0.405	0.575	0.890	0.408

Text2Code

In the Text2Code Benchmarks, we evaluate the Apriel-13B's model's proficiency in both Glide scripting and general JavaScript tasks, focusing on code autocompletion, code editing and code generation capabilities. The scores for code autocompletion and code editing are computed using a comprehensive LLM-as-a Judge. For evaluating Glide code generation quality, where the task is to complete code based on a user instruction in natural language, explicit unit tests are constructed using the ground truth code and used for computing an average pass@1 score for generated snippets.

Average Pass@1 scores for measuring the code generation quality for generic JavaScript (JS) code, the HumalEval benchmark is utilized.

Model	MBPP Python	MBPP JavaScript
SLM May 2025	54.80	53.15
Apriel 13B	***67.60***	***69.02***
GPT-4.1 Mini	80.60	71.54

Glide Coding Benchmarks

Model	Code Autocomplete	Code Edit	Glide JS Code Gen (pass@1)	Generic JS Code Gen (pass@1)
SLM May 2025	43.01	65.52	46.29	48.17
Apriel 13B	***45.41***	***74.10***	***50.56***	***67.07***
GPT-5	43.35	82.93	53.93	87.80
Claude Sonnet 3.7	53.16	80.62	50.34	83.54
GPT-4.1 Mini	45.79	82.42	54.15	87.20
GPT-4.1	47.18	82.68	58.65	87.20

Text2Flow

In the Text2Flow Benchmarks, we quantify alignment between a generated workflow and a designated ground truth by computing the complement of their tree edit distance.

- **Flow Outline:** A similarity metric that compares workflows based only on their overall structure (the sequence of triggers and components). Checks the correct steps are present and ordered properly, without considering the parameters inside each step.
- **Flow Outline with Inputs:** A similarity metric that compares workflows based on both their overall structure and the parameters (inputs) of each step. It checks not only whether the right steps are present and ordered, but also whether each step has the correct details (e.g., table name, field values, recipients).
- **Subflow Inputs and SubFlow Outputs:** A similarity metric based on the subflows within the end-to-end flow. For the subflows, we check correctness by comparing the expected subflow inputs and outputs. They measure the similarity between the actual and expected values.

Model	Outline	Outline + Inputs	Subflow Inputs	Subflow Outputs
GPT-5-chat	74.0	65.3	53.2	64.8
GPT-4.1	73.2	64.5	53.9	62.7
GPT-4.1 Mini	59.3	57.1	56.4	65.0
Gemini-2.5-Flash	63.5	58.8	42.5	**66.0**
Claude Sonnet 3.7	62.1	54.5	55.8	64.1
SLM May 2025	71.3	64.9	68.6	60.2
Apriel 13B	***79.3***	***70.8***	***70.3***	*63.3*

Text2Cypher

The evaluation of Cypher (Neo4J Graph DB query language) is conducted using a benchmark derived from a real-world dataset. This dataset comprises 194 test records, including 124 relevant questions that can be answered using a Knowledge Graph (KG) and 70 irrelevant questions that are either unrelated or not answerable from the KG.

The benchmark assesses the model's ability to accurately generate Cypher queries for retrieving correct information from structured graph data while distinguishing between answerable and unanswerable queries.

Model	Relevant (pass/fail/total)	Acc. %	Irrelevant (pass/fail/total)	Acc. %	Overall Avg. Acc. %
Apriel 13B (new prompt)	102 / 27 / 129	**79.07**	66 / 4 / 70	94.29	86.68
Apriel 13B (prod prompt)	98 / 31 / 129	75.97	68 / 2 / 70	97.14	86.56
May2025-SLM	99 / 30 / 129	76.74	**69 / 1 / 70**	**98.57**	**87.66**

Technical Means for Integration

All interactions and processing occur within the platform's secure architecture.