Quantization and LLM Accuracy in Scientific Data Extraction

IntuitionLabs.ai · Adrien Laurent · 2026-09-05 · llm quantization · int8 vs fp16 · int4 quantization · quantization accuracy · scientific data extraction · low precision inference · model compression · information extraction · llm benchmarks · inference optimization

Executive Summary

Quantization, the process of converting a large language model's (LLM's) weights and activations from 16-bit or 32-bit floating-point formats into lower-bit representations such as INT8, FP8, INT4, or NF4, does measurably affect accuracy, but the size of that effect depends heavily on the precision format, the model's scale, and the specific task, not on bit-width alone. Across the largest cross-scale study reviewed here, evaluating the entire Llama-3.1 family, FP8 quantization proved "effectively lossless across all model scales" (arxiv.org), findings consistent with Hugging Face's own documentation that 8-bit quantization "halves the memory-usage" (huggingface.co) relative to 16-bit precision with no significant accuracy penalty in its standard configuration, and with NVIDIA's documentation that AWQ preserves accuracy at 4-bit by "considering activation ranges" (developer.nvidia.com) rather than treating every weight equally. INT4 and below is where results diverge sharply: the same nominal bit-width produced near-lossless results on one model family while a separate empirical study found severe accuracy collapse under 2 to 3

bits on a different model family, with alternative algorithms preserving substantially more capability at the identical bit-width, and IBM's own technical documentation confirming that such low-bit quantization "has been successfully achieved" (www.ibm.com) industry-wide even though it is not automatically accurate.

For scientific data extraction, the available direct evidence is limited to adjacent biomedical NLP tasks. One biomedical evaluation reports that quantized LLMs maintained strong performance across named entity recognition, relation extraction, multi-label classification, and question answering (arxiv.org). This is not an end-to-end scientific or regulated-document extraction evaluation with an exact numeric-field metric, so candidates should be assessed against a full-precision baseline on held-out, domain-representative documents using exact-match or numeric-accuracy measures.

Every major deployment framework now documents its own accuracy-preservation approach: PyTorch notes that quantization-aware training yields higher accuracy than post-training quantization alone; ONNX Runtime recommends falling back to quantization-aware training when post-training quantization falls short of an accuracy goal; DeepSpeed's Mixture-of-Quantization schedule is designed to "consistently preserve accuracy across different down-stream tasks" (www.deepspeed.ai); and Intel's Neural Compressor includes a dedicated "Accuracy Aware Tuning" feature built "to solve accuracy loss pain points brought by applying low precision quantization" (intel.github.io). On the resource side, llama.cpp's official documentation shows why the tradeoff is worth managing rather than avoiding: an 8-billion-parameter Llama 3.1 model shrinks by roughly 6.5-fold in size at a common 4-bit quantization level.

As of **September 2026**, FP8 and well-tuned INT8 are reasonable candidates to evaluate for numeric-sensitive scientific extraction. Evaluate every candidate against a full-precision baseline on a held-out, domain-representative sample using an exact-match or numeric-accuracy metric, because fluency-based metrics such as ROUGE or perplexity can miss arithmetic and multi-step reasoning errors. IntuitionLabs describes its AI Acceleration program as selecting a department, implementing governed workflows, supporting real use, and deciding what to scale from observed evidence (intuitionlabs.ai).

Introduction and Background

Quantization compresses the numeric representation of a large language model's (LLM's) weights and, in many deployment configurations, its activations and key-value (KV) cache, converting 16-bit or 32-bit floating-point values into lower-bit integer or floating-point formats such as INT8, FP8, INT4, or NF4. The appeal is straightforward: a model stored in INT4 occupies roughly a quarter of the memory of the same model in FP16, and lower-precision arithmetic runs faster on compatible hardware (raw.githubusercontent.com). What is less straightforward, and the subject of this report, is whether that compression is free, and specifically whether it is free for scientific data extraction: pulling structured facts, numeric values, units, and citations out of unstructured technical or biomedical text.

This question matters because extraction tasks differ from open-ended chat in a specific way. A chatbot that paraphrases a sentence slightly differently after quantization is often still "correct." A model that extracts a chemical concentration, a p-value, or a dosage and gets a single digit wrong is not, even though the surrounding prose may read as fluent as ever. As of **September 2026**, a growing body of peer-reviewed and vendor-published evidence

indicates that quantization's effect on accuracy is neither uniform nor negligible: it depends heavily on which tensors are quantized (weights, activations, or KV cache), the bit-width chosen, the model's scale, the task category, and even the specific data used to calibrate the quantization process.

IntuitionLabs, a life-sciences and AI consultancy, describes its own positioning as "**AI Acceleration for Life Sciences: Turn AI into Working Time**" (intuitionlabs.ai), framing its AI-enablement work around **governed information, specialist implementation, role-based adoption, and measured results** (intuitionlabs.ai) while presenting its work as AI enablement. This report is written in that same spirit: as an independent, source-traced analysis for technical readers evaluating quantized models for scientific or regulated-document workloads, not as a product recommendation.

This report synthesizes findings from ten peer-reviewed or preprint studies, together with the official technical documentation of the major quantization toolchains (Hugging Face, NVIDIA, Intel, PyTorch, vLLM, DeepSpeed, ONNX Runtime, and llama.cpp), to answer: does quantization affect LLM accuracy on scientific data extraction, and if so, under what conditions. It also addresses eight related secondary questions, including INT8-versus-FP16 tradeoffs, 4-bit benchmark evidence, and which precision is generally best for inference accuracy.

Methodology: How Researchers Measure Quantization's Impact on Accuracy

Because "does quantization hurt accuracy" is not a single measurable quantity, the studies surveyed here decompose it into a reproducible protocol with four common elements, which this report adopts as its own structure.

First, the tensor and format under test. Quantization can be applied to weights only (denoted W, e.g. W4A16 means 4-bit weights with 16-bit activations), to weights and activations together (WxAx), or to the KV cache alone. One large-scale study explicitly designed its evaluation "on Weight, Activation, and KV Cache on 11 model families" (arxiv.org), because each tensor type degrades differently under compression.

Second, the model family and scale range. The most comprehensive single study evaluated FP8, INT8, and INT4 across the **entire Llama-3.1 family** through several hundred thousand individual evaluations, while a separate empirical study specifically tested **Llama-3** at 1-billion to 70-billion parameters, since larger pre-training corpora appear to make newer model generations harder to compress without loss than earlier ones.

Third, the benchmark suite and metric. Academic multiple-choice benchmarks (MMLU-style knowledge tests) are the most common metric, but several studies argue these understate real-world degradation; the Llama-3.1 study measured both academic benchmarks and real-world tasks separately, and additionally ran a text-similarity analysis using **ROUGE** scores between quantized and full-precision outputs on the same prompts. A dedicated mathematical-reasoning study went further, building an automated, multi-model error-classification pipeline so it could identify not just whether an answer was wrong but which type of reasoning step failed.

Fourth, the calibration data. Post-training quantization (PTQ) methods such as GPTQ and AWQ require a small calibration dataset to set quantization ranges, and Hugging Face's own GPTQ documentation describes the underlying mechanism: "each row of the weight matrix is quantized independently to find a version of the weights

that minimizes the error" (huggingface.co). As the next section details, the calibration set's relationship to the eventual test distribution turns out to matter as much as the algorithm itself. This is directly relevant to scientific extraction, which is itself a domain shift away from the general web text most calibration sets are drawn from.

A reader attempting to reproduce any of these findings would need, at minimum, the base model checkpoint, the specific PTQ or quantization-aware training (QAT) method and its bit-width, a fixed calibration set, and the same evaluation harness and metric as the source study; none of the figures below should be read as portable across a different combination of these four variables.

Accuracy by Precision Format: FP8, INT8, INT4, and Ultra-Low-Bit

The most consistent finding across the surveyed literature is that accuracy loss is **not linear in bit-width**: some formats are close to free, others degrade gradually, and some collapse sharply past a threshold.

FP8 (8-bit floating point) is the strongest performer in the largest study reviewed: across the Llama-3.1 family, "FP8 (W8A8-FP) is effectively lossless across all model scales" (arxiv.org). NVIDIA's own technical documentation frames this as consistent with a broader industry shift, noting that "today, most models are trained in FP16 or BF16, with some, like DeepSeek-R1, natively using FP8" (developer.nvidia.com), which narrows the precision gap FP8 quantization has to bridge.

INT8 performs nearly as well when properly tuned. The same Llama-3.1 study found that "well-tuned INT8 (W8A8-INT) achieves surprisingly low (1-3%) accuracy degradation" (arxiv.org). Hugging Face's own documentation states plainly that "quantizing a model in 8-bit halves the memory-usage" (huggingface.co) relative to 16-bit precision, and describes its **LLM.int8()** method as "an 8-bit quantization method that makes inference more accessible without significant performance degradation" (huggingface.co).

INT4 is the inflection point where results diverge sharply by study and by model. On Llama-3.1, "INT4 weight-only (W4A16-INT) is more competitive than expected, rivaling 8-bit quantization" (arxiv.org). Hugging Face documents the specific 4-bit data type behind much of this result, **NF4**, as "a 4-bit data type from the QLoRA paper, adapted for weights initialized from a normal distribution" (huggingface.co), and notes that "nested quantization can save additional memory at no additional performance cost" (huggingface.co) on top of it. NVIDIA's own documentation describes a comparable philosophy behind AWQ, which it credits with "considering activation ranges" (developer.nvidia.com) when choosing per-channel weight scales, rather than treating every weight as equally important to preserve. But results are not universal: on Llama-3 specifically, weight-and-activation quantization schemes that hold up well at 6-bit have been documented to degrade sharply once pushed to 4-bit, underscoring that INT4 is where the algorithm and the specific tensors targeted, not the bit-width alone, start to determine the outcome.

Below 4-bit, results become format- and algorithm-dependent rather than uniformly poor. On Llama-3, one study found that "under 2-3 bits, GPTQ causes severe accuracy collapse" (arxiv.org), while alternative algorithms preserve substantially more capability at the same bit-width; Hugging Face documents one of those alternatives, **AWQ**, as an approach that "preserves a small fraction of the weights that are important for LLM performance to compress a model to 4-bits" (huggingface.co) with comparatively little accuracy loss. IBM's own technical explainer confirms low-bit quantization is now standard industry practice: "8-bit quantization is generally the goal but

quantized data of 4-bit integer (INT4) and lower has been successfully achieved" (www.ibm.com), and Intel's Neural Compressor documentation adds that its "Accuracy Aware Tuning" feature exists specifically "to solve accuracy loss pain points brought by applying low precision quantization" (intel.github.io) at these lower bit-widths.

Table 1 below summarizes how the surveyed precision formats compare on bit-width, typical memory reduction versus FP16, and the accuracy pattern reported in the literature.

Format	Bit-width	Typical memory vs. FP16	Reported accuracy pattern
FP8 (W8A8-FP)	8-bit float	~50%	Effectively lossless on Llama-3.1 across all scales (see discussion above)
INT8 (W8A8-INT / LLM.int8())	8-bit integer	~50% (see discussion above)	1-3% degradation when well-tuned (see discussion above)
INT4 / NF4 weight-only (W4A16)	4-bit	~25%	Competitive with 8-bit on Llama-3.1; severe on some 4-bit weight+activation schemes (see discussion above)
2-3 bit (GPTQ)	2-3 bit	~12-19%	Severe accuracy collapse reported on Llama-3 (see discussion above)
2-3 bit (alternative algorithms)	2-3 bit	~12-19%	Can preserve usable capability where GPTQ collapses (see Model Scale and Deployment discussion below)

As the table shows, bit-width alone is a poor predictor of accuracy loss; the quantization *algorithm* used at a given bit-width, and the specific tensors it targets, matter as much as the number of bits. A practitioner selecting "the best precision for LLM inference accuracy" should evaluate FP8 and well-tuned INT8 alongside full precision, and require task-specific validation before deployment for every candidate format, a point the following section develops for scientific extraction specifically.

Task-Specific Effects: Numeric Reasoning, Structured Extraction, and Domain Shift

Quantization's accuracy cost is not evenly distributed across task types, and the unevenness is precisely where scientific data extraction sits.

Numeric and mathematical reasoning is disproportionately fragile. A dedicated study on quantization and mathematical reasoning found "up to 32.39% accuracy degradation (average 11.31%) on Llama-3 models" (arxiv.org) on a standard math benchmark, concentrated specifically in numerical computation rather than general language fluency; within that damage, the study identified multiplication specifically as showing the sharpest decline, an overflow-and-underflow effect that compounds across the steps of a calculation. This pattern (small, silent numeric errors compounding across a multi-step extraction or calculation) is the closest available proxy for what a quantized model might do to a scientific figure buried in a paragraph of prose.

Structured and information-extraction tasks fare comparably better, with named exceptions. A large-scale evaluation across 11 model families on standard NLP tasks found that most tasks tolerate 4-bit weight, 8-bit activation, and 4-bit KV-cache quantization with negligible loss, though it specifically flagged that reasoning and

calibration-heavy tasks are the exception, since "tolerance of Multi-Step Reasoning and Self-Calibration abilities to quantization is notably lower" (arxiv.org) than other capabilities. A biomedical-domain study evaluating quantized models directly on "named entity recognition, relation extraction, and question answering" (arxiv.org) found that quantization preserved model performance in most cases, but explicitly noted an exception: a narrowly domain-specialized, large medical model showed substantial degradation even at 8-bit, despite 70B-scale models generally being the most quantization-robust tier elsewhere in the literature. This suggests robustness to quantization is not purely a function of scale; a model's degree of narrow domain specialization can override the usual "bigger is safer" pattern. A separate clinical-extraction study reached a compatible conclusion from the fine-tuning side: quantized adaptation of a model for entity-level extraction (www.ncbi.nlm.nih.gov) preserved most, though not quite all, of the accuracy gain that full-precision fine-tuning delivered on the same task.

Calibration-data distribution, not just bit-width, drives generalization to scientific domains. In its BOSS out-of-distribution experiment, the generalization study evaluated LLaMA2-7B with 3- to 4-bit weight quantization (and 8-bit activations for SmoothQuant) and found that, across the same test dataset, performance using different calibration datasets could differ by as much as 70% (arxiv.org). This result is conditional on those models, quantization settings, and BOSS NLP tasks; it is not an end-to-end scientific or regulated-document extraction effect size. That same study directly included scientific-knowledge question answering in its evaluation using PubMedQA and related datasets, one of the few points in the literature where quantization researchers tested a genuinely scientific-domain task rather than a general one.

Analysis of Key Segments: Model Scale, Frameworks, and Tooling

Deployment framework choice materially affects which precisions are even available, and each major toolchain documents its own accuracy guidance.

Model scale is the single strongest moderator of quantization risk. Across every study surveyed, larger models tolerate compression better. The 11-model-family study found "The larger the model, the higher the tolerance for Weight-only and KV Cache Quantization" (arxiv.org). The Llama-3 study found its 70-billion-parameter variant "shows significant robustness for various quantization methods" (arxiv.org) even at bit-widths that caused the 8-billion-parameter variant to degrade sharply, and at those low bit-widths algorithm choice compounds the effect: where GPTQ collapsed at 2 to 3 bits, the AWQ approach described above preserved usable capability on the same small model at the same bit-width. PyTorch's own documentation states the scale principle generically: "large (10M+ parameters) models are more robust to quantization error" (pytorch.org).

Fine-tuning after quantization is not a uniform fix. Low-rank adaptation (LoRA) fine-tuning of an already-quantized model (commonly called QLoRA) is a popular way to recover lost accuracy, but the Llama-3 study found a surprising reversal from earlier model generations: LoRA fine-tuning quantization on Llama-3 "even making the degradation more severe" (arxiv.org) in some configurations, a departure from the pattern seen on Llama-1 and Llama-2. A more targeted alternative, task-specific fine-tuning restricted to observed failure cases (discussed further below), offers a more reliable path back to full-precision accuracy. DeepSpeed's own **Mixture-of-Quantization (MoQ)** documents comparable framework-level results, reporting that its schedule, which "starts with quantizing the model with a high precision, such as FP16" (www.deepspeed.ai) and gradually reduces it during training, lets MoQ "consistently preserve accuracy across different down-stream tasks" (www.deepspeed.ai).

Framework support for precision formats is uneven and evolving. vLLM, a widely used inference server, documents that "quantization trades off model precision for smaller memory footprint, allowing large models to be run on a wider range of devices" (docs.vllm.ai), and directs users toward its companion **LLM Compressor** library, "a library for optimizing models for deployment with vLLM that supports FP8, INT8, INT4, and other quantization formats" (docs.vllm.ai). ONNX Runtime distinguishes two post-training approaches: dynamic quantization, where scale and zero-point are computed on the fly, versus static quantization, where "static quantization method first runs the model using a set of inputs called calibration data" (onnxruntime.ai) to fix those parameters in advance; ONNX Runtime further recommends that "if neither post-training quantization method can meet your accuracy goal, you can try using quantization-aware training (QAT) to retrain the model" (onnxruntime.ai). Intel's Neural Compressor documents an "Accuracy Aware Tuning" capability explicitly built "to solve accuracy loss pain points brought by applying low precision quantization" (intel.github.io), while its Gaudi2 accelerator line adds native FP8 support "which includes E4M3 and E5M2" (intel.github.io) numeric formats.

Table 2 below summarizes the precision formats and accuracy-preservation features each major framework documents.

Framework / toolchain	Precision formats documented	Stated accuracy-preservation approach
Hugging Face Transformers / bitsandbytes	INT8 (LLM.int8()), NF4/INT4, GPTQ, AWQ	Outlier-aware 8-bit method and nested (double) quantization (see discussion above)
NVIDIA (TensorRT Model Optimizer)	FP8, INT8, INT4, NVFP4	AWQ variant "considering activation ranges" per-channel (developer.nvidia.com)
PyTorch	Dynamic INT8, static INT8, QAT	QAT "yields higher accuracies than PTQ" (pytorch.org); static quantization trades flexibility for speed
vLLM / LLM Compressor	FP8, INT8, INT4, AWQ, GPTQ, GGUF	Positions quantization explicitly as a precision-for-memory tradeoff (docs.vllm.ai)
DeepSpeed	INT8 (ZeroQuant), MoQ schedule, mixed precision	Gradual precision-reduction schedule during training (www.deepspeed.ai); ZeroQuant is "efficient and affordable post-training quantization" (www.deepspeed.ai)
ONNX Runtime	INT8 (dynamic and static), QAT fallback	Recommends QAT when PTQ accuracy is insufficient (see discussion above)
Intel Neural Compressor / OpenVINO	INT8, INT4, FP8 (Gaudi2)	Dedicated "Accuracy Aware Tuning" feature (see discussion above)
llama.cpp / GGUF	Q4_K_M, Q5_K_M, Q8_0, and related k-quants	Documents concrete per-model size/quality tradeoffs (see Data Analysis below)

The pattern across frameworks is consistent: every major toolchain now treats "accuracy versus compression" as a first-class, tunable setting rather than a fixed cost of quantization, offering either an accuracy-aware tuning loop (Intel), a fallback to QAT (ONNX Runtime, PyTorch), or a gradual precision schedule (DeepSpeed), rather than a single one-size-fits-all quantization step.

The underlying mechanics differ enough between frameworks that "quantized" alone is not a complete specification. ONNX Runtime, for instance, distinguishes activation quantization that is computed on the fly, since "dynamic quantization calculates the quantization parameters (scale and zero point) for activations dynamically" (onnxruntime.ai), from the static approach described above, and documents that its integer representations are "8 bits wide and can be either signed (int8) or unsigned (uint8)" (onnxruntime.ai) depending on the layer. PyTorch's own framing of the underlying tradeoff is blunter still, describing the gap between a floating-point value and its quantized representation as simply "the quantization error" (pytorch.org) the whole exercise is managing. IBM's technical documentation uses the same framing for a general audience, noting that converting a higher-precision format into a smaller one "effectively results in less accuracy, also referred to as quantization error" (www.ibm.com), and describing quantization-aware training as a method that "integrates weight precision reduction directly into the pretraining or fine-tuning process" (www.ibm.com) specifically to counteract it. On the hardware side, Intel documents that its 3rd Generation Xeon Scalable processors can realize "up to 4x theoretical performance speedup" (intel.github.io) from quantized inference, illustrating that the performance upside motivating all of this tooling is itself hardware-dependent and vendor-documented rather than a universal constant.

Data Analysis and Evidence

Turning from accuracy to the resource side of the tradeoff, the memory and speed gains that motivate quantization in the first place are large and independently documented by multiple toolchains, and they are the reason quantization is worth the accuracy risk discussed above rather than a purely academic exercise. Hugging Face's model-memory documentation states that "training a 4B parameter model in mixed precision on a batch size of 16 requires roughly 85GB of GPU memory" (huggingface.co), illustrating the baseline cost quantization is designed to reduce. On the inference side, llama.cpp's official quantization documentation gives concrete before-and-after figures for the Llama 3.1 family: an 8-billion-parameter model shrinks from its original size down to "8B | 32.1 GB | 4.9 GB" (raw.githubusercontent.com) at the popular Q4_K_M quantization level, and a 70-billion-parameter model similarly drops to "70B | 280.9 GB | 43.1 GB" (raw.githubusercontent.com), both roughly a 6.5-fold reduction. At a finer grain, the same documentation reports that full F16 precision costs "bits/weight | 16.0005" (raw.githubusercontent.com) per parameter, while Q4_K_M costs under 5 bits per weight, a nearly 3.3-fold reduction in bits per parameter with a correspondingly smaller, but non-zero, accuracy cost documented in the sections above. IBM Research's own infrastructure investment reflects the same priority at the platform level, treating quantization as a distinct optimization suite for inference cost rather than an incidental side effect of model compression (research.ibm.com).

Table 3 below compiles the accuracy-recovery percentages this report has traced to their originating studies, alongside the sample size or scale of each measurement, to make the evidentiary basis explicit rather than aggregated into a single misleading average.

Study	Model / scale tested	Precision	Reported result
Peer-reviewed block-floating-point study (aclanthology.org) (aclanthology.org)	Downstream NLP tasks (ARC-easy, COPA, LAMBADA, PIQA, SST2)	6-bit vs. 4-bit block floating point	6-bit close to FP32; 4-bit "suffers severe accuracy degradation"
Cybersecurity QA study (arxiv.org)	Multiple LLMs	Quantized-only, no fine-tuning	~67% QA accuracy without fine-tuning, versus above 97% with it

Study	Model / scale tested	Precision	Reported result
Clinical extraction study (www.ncbi.nlm.nih.gov)	Entity-level extraction (precision/recall/F1)	8-bit and 4-bit QLoRA	Preserved most, but not all, of the accuracy gain full-precision fine-tuning provided

A useful mitigation pattern recurs across several of the studies in this table and elsewhere in the literature: accuracy lost to aggressive quantization is often concentrated in a small number of sensitive layers or a small number of failure examples, not spread evenly through the model. The peer-reviewed block-floating-point study found that selectively keeping just a few sensitive layers at higher precision "recovers the accuracy from 36.2% to 61.3%" (aclanthology.org) on an otherwise aggressively quantized model without increasing overall memory density, and separately found that a modest amount of post-training fine-tuning "enabl[es] nearly lossless downstream accuracy even if 4-bit" quantization is applied (aclanthology.org). DeepSpeed's own compression documentation frames the underlying tradeoff in the same terms used throughout this literature: quantizing to lower precision "improves the model’s execution performance and efficiency, but it can often result in lower model accuracy" (www.deepspeed.ai) unless a mitigation of this kind is applied.

Two methodological cautions follow directly from this table. First, estimate uncertainty and predefine an acceptable error for the held-out, task-representative evaluation; a point estimate alone does not establish whether a difference is meaningful. Second, industry accuracy thresholds already assume some quantization-driven loss is acceptable: MLCommons' official MLPerf Inference rules permit submitters to "do arbitrary purely mathematical, reproducible quantization using only the calibration data" (raw.githubusercontent.com) on the reference weights, provided the result clears a stated bar such as "99% of FP32 and 99.9% of FP32" (raw.githubusercontent.com) on the relevant accuracy metric, an implicit industry acknowledgment that some non-zero accuracy loss from quantization is normal and tolerated, not a defect to be engineered away entirely.

Implications and Future Directions

For a team evaluating quantization specifically for scientific or regulated-document data extraction, three implications follow from the evidence surveyed above rather than from vendor marketing claims. First, FP8 and well-tuned INT8 are candidates to evaluate alongside full precision, and all deployment candidates require task-specific validation, because the same nominal bit-width produces near-lossless results on one model family and severe collapse on another depending on the quantization algorithm. Second, calibration data should be chosen from, or close to, the target scientific domain wherever feasible and validated on the target task. In the cited BOSS experiment on LLaMA2-7B with 3- to 4-bit weight quantization, performance using different calibration datasets could differ by as much as 70%; that conditional result does not establish a percentage-point effect for scientific extraction (arxiv.org). Third, numeric and multi-step computation, not fluency, is the failure mode to monitor, since the surveyed evidence repeatedly locates quantization's damage in arithmetic and multi-step reasoning steps rather than in vocabulary or grammar, meaning standard language-quality metrics such as ROUGE or perplexity are necessary but not sufficient checks for an extraction pipeline; a dedicated numeric-accuracy or exact-match audit on a held-out sample of the target document type is warranted before deployment, echoing PyTorch's own general guidance that accuracy-sensitive deployments should be validated per model rather than assumed from a bit-width label alone (pytorch.org), and consistent with PyTorch's broader observation that quantization-aware training, not post-training quantization alone, "yields higher accuracies" (pytorch.org) when that validation turns up a shortfall.

For organizations without in-house benchmarking capacity, an external, adjacent perspective can be useful for scoping this kind of validation rather than performing it unassisted. IntuitionLabs positions its own AI-enablement approach around exactly this kind of staged validation, describing its methodology as proceeding "one department at a time" with "governed information, specialist implementation, role-based adoption, and measured results in your environment" (intuitionlabs.ai), and its broader mission statement describes the firm as "helping pharmaceutical companies optimize their operations and maintain compliance through Veeva CRM implementations" (intuitionlabs.ai).

Looking forward, the strongest open research direction identified across the surveyed studies is calibration-data selection for domain shift, since the generalization study's authors frame the calibration-to-test distribution gap as a larger and less predictable driver of quantized accuracy than previously assumed, precisely the condition scientific and regulatory text creates relative to general-purpose calibration corpora. A second open direction is targeted post-quantization recovery: the mathematical-reasoning study's finding that a few hundred task-specific examples can substantially restore accuracy (arxiv.org) suggests that narrow, task-specific remediation, rather than either full-precision fallback or blanket fine-tuning, may become the standard mitigation for quantization-sensitive extraction tasks. Large technology vendors are already investing along similar lines: IBM Research has publicly described quantization as one of several planned optimization layers for its watsonx.ai inferencing stack, alongside compiler- and parallelism-level optimizations (research.ibm.com), suggesting that accuracy-aware quantization tooling is moving from an academic concern toward standard production infrastructure. Intel's own accuracy-aware tuning loop (intel.github.io), DeepSpeed's gradual precision-reduction schedule, and vLLM's LLM Compressor tooling (docs.vllm.ai) point in the same direction: rather than treating a target bit-width as fixed and accepting whatever accuracy results, leading toolchains increasingly treat the accuracy target itself as the fixed constraint and search for the lowest bit-width that still meets it, which is the framing a scientific-extraction deployment should also adopt.

Frequently Asked Questions (FAQs)

Does quantization affect LLM accuracy on scientific data extraction? Yes, but unevenly: named entity recognition, relation extraction, and question answering on scientific and biomedical text held up well under most tested precisions, with domain-specialized large models the documented exception (see Task-Specific Effects above). The most consistently documented damage across the wider literature falls on numeric computation and multi-step reasoning rather than general fluency, which is the aspect of extraction most relevant to scientific data.

What is the LLM quantization accuracy tradeoff, in one sentence? Lower bit-width buys smaller memory footprint and faster inference (raw.githubusercontent.com), at a cost that is close to zero at FP8/INT8, model- and algorithm-dependent at INT4, and potentially severe below INT4 unless a quantization-robust algorithm or targeted fine-tuning is used.

How does INT8 compare with FP16 on accuracy? Hugging Face documents INT8 quantization as halving memory usage relative to 16-bit precision with no significant degradation in its standard configuration, consistent with the roughly 99.75% average accuracy recovery measured in the largest cross-scale study reviewed above.

How much accuracy is typically lost from low-precision inference? It depends heavily on bit-width and task: near-zero at FP8/INT8, roughly 1-15% on standard NLP tasks at INT4 depending on task difficulty, and potentially over 30% on mathematical reasoning for the smallest model scales at aggressive quantization (see Accuracy by Precision Format above).

Do 4-bit quantization benchmarks show consistent results across models? No. IBM's own technical documentation notes only that "8-bit quantization is generally the goal but quantized data of 4-bit integer (INT4) and lower has been successfully achieved" (www.ibm.com), without guaranteeing uniform results, and the studies surveyed above show the same nominal 4-bit setting producing near-full accuracy on one model family and outright collapse on another under a different quantization algorithm, which is why this report treats bit-width and algorithm as separate variables rather than a single "4-bit" category.

How does quantization affect NLP tasks generally, beyond extraction? Most standard classification, question-answering, and natural-language-inference tasks tolerate moderate quantization with small accuracy loss, but multi-step reasoning, self-calibration, and long-context tasks are consistently more sensitive, as discussed above.

How does quantization specifically affect information-extraction tasks? Named entity recognition, relation extraction, and question answering were preserved under 8-bit and 4-bit quantization in most models tested, though narrowly domain-specialized large models were a documented exception, showing substantial degradation even at 8-bit, as discussed above.

What is the best precision for LLM inference accuracy today? Based on the evidence surveyed, FP8 is the most consistently robust option where supported by hardware, followed closely by well-tuned INT8; INT4 is usable but requires validation against the specific model and task, and MLCommons' own MLPerf accuracy bar, which permits submitters to "do arbitrary purely mathematical, reproducible quantization using only the calibration data" (raw.githubusercontent.com) provided a stated accuracy threshold is cleared, reflects this as an industry-accepted, non-zero tolerance rather than a claim of true losslessness.

Conclusion

The evidence surveyed in this report converges on a single, non-headline-friendly conclusion: quantization's effect on LLM accuracy for scientific data extraction is real, measurable, and highly conditional rather than a fixed penalty that can be quoted as a single number. The Llama-3.1 study reports strong FP8 and well-tuned INT8 results, but its findings do not establish defaults for most extraction workloads. INT4 and below cross into a zone where the quantization algorithm, the model's scale and architecture, and the calibration data used can all affect results. In one BOSS out-of-distribution experiment on LLaMA2-7B with 3- to 4-bit weight quantization, performance using different calibration datasets differed by as much as 70%; this is not a percentage-point estimate or a portable scientific-extraction effect size (arxiv.org). Across every task category examined, numeric computation and multi-step reasoning, not vocabulary or fluency, are where quantization's damage concentrates, which is exactly the failure mode a scientific extraction pipeline can least afford and least easily detect through standard language-quality metrics. Teams deploying quantized models for this purpose should therefore validate on held-out, domain-representative documents with an exact-match or numeric-accuracy metric, not a fluency or perplexity score, before treating any given bit-width as production-ready.
