May 2025

Celebrating 40 Years

Benchmarking Generative AI for CSA Clause Extraction and CDM Representation

CONTENTS

EXECUTIVE SUMMARY

This whitepaper explores recent advances in generative artificial intelligence (AI) and its potential to automate the extraction, interpretation and digitization of legal clauses from ISDA's credit support annexes (CSAs) into standardized, machine-readable code using the Common Domain Model[1] (CDM). Specifically, the paper benchmarks and reviews the ability of generative AI to accurately extract five CSA clauses and digitize them into CDM format. The findings suggest that integrating industry-specific data significantly boosts generative AI accuracy, larger generative AI models typically handle nuanced legal language better, and generative AI can be utilized within a modular framework (eg, agentic AI) to extract legal clauses more accurately and efficiently.

This study addresses a fundamental question: how can generative AI independently and accurately extract clauses from CSAs and digitize them into CDM format

Generative AI is a subset of artificial intelligence that can understand and create new content. In general, it has been trained on a vast and wide range of publicly available books, websites and other sources, giving it a broad understanding of human knowledge and enabling it to learn language patterns to generate responses by predicting what comes next in a text sequence. The first example of generative AI to gain mainstream attention was ChatGPT, which was made public in late 2022 and fueled explosive interest in and growth of the generative AI sector.

AI has a long history going back decades. While traditional AI[2] solutions already exist to parse and digitize CSAs into CDM representation with impressive accuracy, there are compelling reasons for exploring generative AI as an alternative approach for this purpose. Current solutions[3] that rely on traditional AI require extensive programming, large training datasets and regular maintenance to handle new or non-standard document variations. In contrast, generative AI can adapt to these variations in document structure and language without the need for extensive retraining, making it potentially more flexible and efficient for processing new versions of documents or non-standard, highly customized documents. This is especially beneficial for financial institutions with large portfolios of non-digitized or partially digitized CSAs or financial institutions with older or heavily customized CSAs.

This study addresses a fundamental question: how can generative AI independently and accurately extract clauses from CSAs and digitize them into CDM format?

Financial institutions must respond to growing regulatory demands for real-time reporting of margin requirements and collateral valuations. However, legal documents like CSAs, which are integral to the ISDA Master Agreement, contain clauses with complex terminology, numeric references and cross-references, making manual clause extraction from these types of documents time-consuming and prone to errors. Although generative AI offers promising capabilities, it typically struggles with this[4] because it does not contain sufficient expertise in derivatives to accurately capture the data. To enable generative AI to produce more accurate interpretations and structured outputs from CSAs, high-quality domain-specific data, such as the ISDA Clause Library[5] and the ISDA Documentation Taxonomy[6], needs to be incorporated. When backed by

[1] Common Domain Model (CDM), www.finos.org/common-domain-model

[2] In the context of the paper, traditional artificial intelligence (AI) refers to earlier language models and traditional rule-based and machine-learning systems

[3] Although most solutions rely on traditional AI, generative AI is a rapidly evolving area and new vendor solutions are being tested and implemented that take advantage of generative AI

[4] Towards Robust Legal Reasoning: Harnessing Logical LLMs in Law, Kant et al, February 2025, arxiv.org/html/2502.17638v1

[5] What is the ISDA Clause Library?, www.isda.org/a/u6TgE/ISDA-Clause-Library-factsheet-2021.pdf

[6] The ISDA Clause Library was developed based on the ISDA Documentation Taxonomy: www.isda.org/2020/06/23/isda-launches-clause-library/

relevant domain data, generative AI can better understand subtle variations in provision wording and uncover interdependencies among clauses.

This study benchmarks multiple large language models (LLMs), which are a subset of the broader generative AI category that specifically processes and generates human language, on a sample of 60 CSAs of various sizes and complexities, measuring their ability to accurately extract and digitize them into CDM format.

Three key findings from the benchmarking study reveal opportunities as well as constraints:

- **Industry-specific data enhances accuracy:** Regardless of the LLMs used, augmenting them with industry-specific taxonomies, standard clause references and legal definitions greatly improves the accuracy of extraction, interpretation and digitization.
- **Larger LLMs typically demonstrate advanced nuance handling:** Proprietary[7] and larger open-source[8] LLMs consistently recognize variations in clauses, conditional language and references to different sections.
- **Agentic frameworks offer best results:** In parallel with this study, ISDA collaborated with the AWS Innovation Center from Arizona State University on the development of an agentic (multi-agent)[9] framework that allocates tasks across specialized agents, each focused on extracting specific clauses. This multi-agent and modular approach, which integrates CSA-specific domain knowledge within each agent, reduces management overheads and increases accuracy.

While this study suggests LLMs can greatly improve the digitization and standardization of CSA clauses into CDM representation, it also emphasizes the importance of leveraging existing ISDA standards, documentation and taxonomies in enabling effective use of AI. Further research on how ISDA documentation could improve AI accuracy and efficiency will be valuable.

While this study suggests LLMs can greatly improve the digitization and standardization of CSA clauses into CDM representation, it also emphasizes the importance of leveraging existing ISDA standards, documentation and taxonomies in enabling effective use of AI

This whitepaper is structured in five parts. Section 1 introduces the topic and describes the benefits of digitizing CSAs and the opportunity for generative AI. Section 2 sets out the design of the benchmarking study and the LLMs tested. Section 3 reports the detailed results, including a comparison of proprietary and open-source LLMs. Section 4 describes the multi-agent (agentic) framework employed to use generative AI for CSA-to-CDM digitization. Section 5 concludes with key takeaways and a forward-looking perspective on how generative AI can streamline document standardization.

[7] Proprietary large language models (LLMs) refer to LLMs like OpenAI's GPT models, which are privately developed and owned

[8] Open-source LLMs are those that are publicly available and can be deployed within an organization's own private infrastructure

[9] An agentic (or multi-agent) framework is a system architecture where multiple autonomous AI components, called agents, work together to solve complex problems

1. INTRODUCTION

1.1 Background and Challenges

A vast array of legal documents related to derivatives transactions are managed by financial institutions. CSAs are especially important as they define the terms for collateral exchange to mitigate counterparty credit risk. However, many CSAs remain non-digitized or only partially digitized, meaning they are not represented in a standardized way that is machine-readable (ie, in CDM format). This lack of standardization and machine-readable formatting across large contract portfolios makes data processing difficult, can lead to data management inefficiencies and increases the risk of errors.

> Non-digitized or partially digitized legacy contracts complicate compliance checks, risk assessments and necessary changes, increasing both regulatory and operational risks

Firms need to update their contract portfolios to stay compliant with regulations as they continue to evolve. However, non-digitized or partially digitized legacy contracts complicate compliance checks, risk assessments and necessary changes, increasing both regulatory and operational risks. Large volumes of non-standard agreements also hinder digital transformation because their varied formats and content can require firms to store and manage them in different systems or processes, leading to data silos and inconsistent workflows. These inconsistencies make it more challenging to consolidate exposures and adjust collateral terms quickly, increasing operational risk and reducing efficiency.

When collateral processes remain manual or partially digitized, institutions encounter additional inefficiencies and operational challenges. Many still depend on bespoke, labor-intensive methods that become even more problematic during market stress. Without end-to-end automation, tasks such as onboarding new counterparty relationships, communicating margin calls, assessing collateral eligibility, processing settlements and coordinating substitutions are prone to delays and errors – risks that can escalate in volatile conditions. Inconsistent or non-digitized CSA data also complicates reconciliation and compliance, potentially leading to disputes or issues with margin requirements. Ultimately, non-digitized CSAs require more resources, increase errors and complicate regulatory compliance, highlighting the need for digital transformation in collateral management.

This study assesses whether supplying LLMs with a small set of relevant CSA-specific examples and domain knowledge can significantly improve performance. Providing LLMs with CSA-specific information through targeted examples can determine if this helps LLMs better understand and process specialized legal terminology and structures in CSAs.

Given the complexity and specialized nature of derivatives documentation, it is uncertain whether current ISDA documentation structures and taxonomies fully support the optimal use of generative AI. Preliminary results suggest that further exploring how ISDA standards, clause libraries and taxonomies can be explicitly tailored or structured to enhance AI interpretability may yield potential benefits. However, additional research is needed to determine precisely which enhancements to ISDA's existing standards and documentation would most effectively improve the accuracy and efficiency of generative AI applications.

1.2 Benefits of Digitization and Automation in Collateral Management

Global regulators have indicated that standardizing and automating non-cleared margin processes help minimize delays and failures in margin calls[10]. Because these processes can be different across counterparties based on the terms of the relevant collateral agreement, automation requires having a machine-readable data input representing the relevant terms. This can be referred to as 'digitizing' the agreement.

By digitizing collateral agreements into machine-readable formats and automating collateral management workflows, firms can significantly improve efficiency and control. In addition, replacing manual tasks with automated systems allows firms to manage higher collateral volumes without resulting in bottlenecks or increasing risks during periods of market stress. Digitizing collateral agreements in a standardized data model ensures the necessary agreed data points to drive automation are accurately and efficiently maintained. Key benefits include:

- Improved efficiency and accuracy: Eliminating manual data entry and email-based processes accelerates counterparty onboarding and margin call handling, while also reducing human error, disputes and reconciliation efforts for collateral balances.

- Enhanced compliance: Digitized CSAs and standardized data enable demonstration of compliance with margin rules and reporting requirements. Automation ensures timely and auditable collateral movements that support strict regulatory deadlines.

- Risk reduction: Firms can more effectively manage liquidity and prevent shortfalls by leveraging real-time data flows and reducing their reliance on spreadsheets. Full automation (from contract negotiation to settlement) decreases operational risk and ensures collateral processes remain resilient even during stressful periods.

- Cost reduction: Automated systems enable optimal collateral allocation and seamless processing and result in lower long-term costs.

Digitization ultimately offers a more scalable and transparent collateral management function that can effectively handle growing trade volumes and increasing regulatory complexity. This is why organizations like ISDA are leading this transformation by focusing on digitizing CSA documentation and margin call data, as well as streamlining reconciliations to minimize risk.

1.3 Role of the CDM in Contract Digitization

The CDM has become an important tool for standardizing the digital representation of financial contracts and processes. At its core, the CDM is a machine-readable, standard data model that describes financial products, trades and lifecycle events in a cohesive manner. By establishing a unified approach to recording trade events and contract terms, the CDM eliminates inconsistent data representations that can lead to reconciliation issues. As well as reducing the burden of reconciliation, the CDM helps lower the risk of mismatches or settlement failures by ensuring consistent data definitions across all parties.

ISDA initially developed the CDM for over-the-counter derivatives to encode the complete trade lifecycle (eg, amendments, payments and defaults) in a standardized digital form. This initiative has expanded to include securities lending and repo markets, with the International Securities Lending Association and International Capital Market Association applying the CDM to their master agreements and transactions.

[10] Streamlining VM Processes and IM Responsiveness of Margin Models in Non-centrally Cleared Markets, Basel Committee on Banking Supervision and the International Organization of Securities Commissions, January 2024, www.bis.org/bcbs/publ/d569.pdf

In the collateral management space, the CDM can be used to create standard digital representations of CSA provisions, which make essential credit support terms machine-executable. This approach facilitates straight-through processing by enabling legal terms to flow directly into collateral systems without adding any manual interpretation. ISDA has also integrated the CDM with the ISDA Create[11] document negotiation and execution platform, allowing finalized agreements to be seamlessly output as structured CDM data, which minimizes downstream discrepancies and disputes.

The primary purpose of the CDM is to establish financial contracts on a unified, shared data foundation and enhance interoperability among firms and systems. Oversight of the CDM was transferred to FINOS in 2022 with the aim of fostering an open-source community to further encourage industry uptake and promote broader adoption. As more contracts are modeled using the CDM framework, the market will become more efficient and resilient, resulting in easier automation and regulatory reporting.

1.4 The Opportunity for Generative AI

Older contract analysis approaches using traditional AI, especially rules-based systems, rely on hand-crafted 'if-then' rules or template matching and need extensive training on labeled examples (example text or documents that are manually annotated for the task) for each new document type.

Modern LLMs (eg, GPT-4o) represent a significant leap beyond older solutions because they come pre-equipped with a broad understanding of language, enabling them to handle new tasks with minimal or no additional training, a level of flexibility that was previously unattainable. These LLMs are vast neural networks[12] pre-trained on extensive text corpora, which allows them to generate and comprehend language with near-human proficiency.

According to the Institute of International Finance[13], 86% of financial firms anticipate moderate to significant growth in the use of generative AI in the near future, particularly for risk analysis and document processing. Many banks believe generative AI will enhance risk management by accelerating risk detection and facilitating faster document queries and compliance checks[14].

Key differences between modern LLMs and traditional AI systems include:

- **Adaptability:** LLMs can be rapidly adjusted to new document types or clauses using prompts[15] rather than requiring separate earlier language models or rule sets for each document type. With traditional AI, introducing a new contract template or clause often requires writing new rules or retraining using hundreds or thousands of annotated samples. According to an article in the Massachusetts Institute of Technology Public Interest Technologist, LLMs signify a paradigm shift, as a single LLM can manage many tasks by decoupling learning from time-consuming training processes[16]. Instead of coding different specialized language models, firms can utilize one LLM that learns a new task at run time with significantly less effort.

[11] www.isda.org/isda-solutions-infohub/isda-create/

[12] Neural networks are complex computing systems that are inspired by the structure of the human brain

[13] IIF-EY Annual Survey Report on AI/ML Use in Financial Services, December 2023, www.iif.com/portals/0/Files/content/2023%20IIF-EY%20Survey%20Report%20on%20AI_ML%20Use%20in%20Financial%20Services%20-%20Public%20Report%20-%20Final.pdf

[14] Artificial Intelligence in Financial Services, Department of the Treasury, December 2024, home.treasury.gov/system/files/136/Artificial-Intelligence-in-Financial-Services.pdf

[15] Prompts are a set of instructions or inputs given in real-time to LLMs that guide how they respond or what kind of output they generate

[16] Few-shot Learning and AI beyond Code, Massachusetts Institute of Technology, December 2024, technologist.mit.edu/few-shot-learning-and-ai-beyond-code/

- **Context understanding:** Traditional AI solutions are limited to structured, rules-based tasks, while LLMs can adapt to complex contracts with bespoke wording. Traditional AI also has a limited ability to handle large texts (often only able to handle a few paragraphs) and frequently misses global context or cross-references without extensive additional programming. LLMs can take in increasingly larger amounts of text and can grasp context and nuance in lengthy documents. They can accurately interpret the relationships among clauses, definitions and numeric terms and can also interpret free-form text, enabling them to handle novel or varied phrasing that rigid rules-based systems might overlook. This ability to handle larger amounts of text and deeper comprehension results in a more accurate extraction of obligations and terms from contracts.

- **Few-shot learning capability:** LLMs can perform few-shot learning without requiring retraining on task-specific data. Few-shot learning is part of a larger set of prompting techniques that involve an LLM learning how to perform a task given a few examples. This contrasts with earlier learning models, which require hundreds of labeled training samples to extract clauses. Leveraging the extensive knowledge gained during pre-training, LLMs can accurately recognize and even normalize a clause after encountering one or two examples. Researchers attribute this to the LLM's complexity: when an LLM's internal representation is sufficiently rich, it can generalize from only a few examples[17]. This few-shot ability significantly alleviates the need to create many annotated examples for every new contract type or provision.

LLMs can take an unfamiliar contract and not only recognize its components but also transform them with minimal human guidance

- **Output generation:** Earlier language models, particularly those with rules-based systems, were unable to truly generate new text and were restricted to highlighting predefined clauses or filling in fields. LLMs can not only classify or extract data but can also generate coherent, contextually appropriate text as an output. This ability allows them to create natural-language summaries, suggest revisions or draft standardized versions of clauses.

In summary, while earlier language models and rules-based systems are important in their narrower tasks, they struggle to match the flexibility and efficiency of modern LLMs in contract analysis. LLMs offer adaptable intelligence, deeper language understanding and generative abilities that far exceed the rigidity and data demands of earlier tools. LLMs can take an unfamiliar contract and not only recognize its components but also transform them with minimal human guidance – for example, digitizing unstructured legal text into a structured summary or a standardized set of data fields.

Several AI-driven systems are already in use for parsing and standardizing CSAs, including solutions that support CDM digitization workflows[18]. While many of these systems leverage machine learning to extract and digitize clause data into CDM format, they do not use or only partially use generative AI.

[17] Few-shot Learning and AI beyond Code, MIT, December 2024, technologist.mit.edu/few-shot-learning-and-ai-beyond-code/

[18] For example, vendors such as Vermeg and Ark51 offer AI-based services that assist with CDM standardization by extracting and digitizing clause data into CDM format. See: www.isda.org/2024/06/10/vermeg-integrates-common-domain-model-into-colline-collateral-management-system/ and www.isda.org/2024/11/05/ark-51-adopts-cdm-for-csa-data-extraction/

Deploying LLMs in derivatives documentation does present challenges. General LLMs often require supplemental domain knowledge to accurately interpret specialized legal terminology, numerical data and cross-referenced clauses in CSAs. Integrating an LLM with industry-specific resources, such as the ISDA Documentation Taxonomy, is necessary to effectively tailor generative AI for derivatives markets.

1.5 The Role of Domain Data in Generative AI

LLMs are trained using billions of text segments called tokens[19] and are designed to understand and generate human-like text. Although these LLMs manage general English well, they struggle with specialized contractual clauses. One of the main reasons is the lack of publicly available data to pre-train the LLMs. Because of this, LLMs are not fully equipped to deal with legal texts, which have distinct terms, clause structures and numerical references. For LLMs to effectively analyze legal documents, they require more than just raw language processing capacity – they need domain-specific data and context.

> For LLMs to effectively analyze legal documents, they require more than just raw language processing capacity – they need domain-specific data and context

While this whitepaper uses prompting techniques like few-shot learning to provide domain-specific knowledge, the findings align with broader industry research indicating that fine-tuning LLMs with specific domain information can significantly improve performance on contract-related tasks. For example, LLMs often confuse 'threshold' with 'threshold amount' and 'minimum transfer amount' without guidance. Even smaller LLMs that are augmented with domain-specific data can outperform some larger state-of-the-art LLMs. This emphasizes the importance of a robust library of legal domain information like synonyms and variants[20].

Domain expertise also improves numeric accuracy. In CSA documents, interest rates, haircuts and thresholds typically fall within certain ranges, but numeric references can also refer to specific legal paragraphs. A domain-savvy LLM is more likely to distinguish references like 'paragraph 13(c)(ii)' from monetary amounts and avoid transcription errors. Techniques like retrieval-augmented generation – a process of retrieving external data sources like documents or databases as inputs to LLMs – and improved prompt design further enhance numerical precision[21].

Ultimately, combining LLMs with domain data leads to more accurate extraction of complex contractual information. Integrating LLMs with domain data and the CDM provides a pathway to more efficient collateral management, transforming derivatives documentation practices and unlocking significant long-term advantages.

[19] Tokens are pieces of text, such as words, parts of words or punctuation, which an LLM breaks down and processes to understand and generate language

[20] Large Language Models are legal but they are not: Making the case for a powerful LegalLLM, Jaykumar et al, December 2023, https://aclanthology.org/2023.nllp-1.22.pdf

[21] Optimizing Numerical Estimation and Operational Efficiency in the Legal Domain through Large Language Models, Huang et al, July 2024, arxiv.org/html/2407.19041v1

2. BENCHMARKING GENERATIVE AI FOR CSA CLAUSE EXTRACTION

This section outlines the approach taken to benchmark various LLMs for extracting CSA clauses and digitizing them into CDM format. The goal was to determine which LLMs achieved the highest accuracy in extracting CSA clauses and digitizing them into CDM format given the same set of prompts. The evaluation approach is outlined, along with a summary of the findings and insights on the performance of the LLMs. Many legacy CSAs exist as scanned PDFs and processing these documents may require an additional text extraction step. This study does not address the challenges of converting PDF documents into text format – it focuses specifically on an LLM's ability to extract, interpret and digitize relevant clauses once the raw text is available.

2.1 Evaluation Considerations

The benchmarking study measures various LLMs, including small, large, proprietary and open source. Various factors were considered when choosing the LLMs tested, including:

Privacy and Security Considerations

Financial contracts usually contain highly confidential information, and many firms do not allow this contract data to be stored or sent off-premises. This means external or cloud-based LLM services like OpenAI's LLMs may conflict with internal data policies. For this reason, some open-source LLMs that can be housed internally were included in the study.

Cost Considerations

Along with privacy and security considerations, the financial implications of deploying LLM technology have important practical considerations.

LLM usage costs can rapidly escalate. Commercial LLM vendors typically charge for their services based on token usage, usually calculated per 1,000 tokens. Processing a 10-page CSA document – or potentially something much longer in the case of the ISDA Master Agreement and its amendments – can use tens of thousands of tokens per document. This volume of data can lead to significant costs that accumulate quickly.

Alternatively, organizations can deploy LLMs on their own infrastructure. Although acquiring and operating the necessary infrastructure involves significant upfront capital investment, this approach can lead to lower per-document costs over time, especially for high-volume usage scenarios.

Given cost issues, the benchmarking study includes LLMs of varying sizes and computational requirements. Due to the substantial context windows[22] required to process CSA documents in various forms, LLMs like Mixtral 8x7B were not included because they lack a large enough context window to effectively take in the entirety of these complex legal documents. Context window extension strategies were not explored for this study. Instead, it focuses on LLMs that already provide sufficiently large context windows for CSA analyses.

[22] Context window is the maximum amount of text an LLM can process at once. Many credit support annexes (CSAs) that include the ISDA Master Agreement require larger context windows

2.2 Evaluation Approach and LLMs Tested

Evaluation Dataset

Sixty CSAs were assembled from multiple sources, including in-house samples and legacy CSAs (1994, 1995 and 2016 variants, governed by New York law). Each document was manually labeled and reviewed by subject matter experts to identify the following:

- Clause boundaries: Sections describing threshold, minimal transfer amount (MTA), rounding, base currency and eligible currency.
- Numeric fields: Exact text for amounts (eg, USD 50,000).
- CDM JSON output: A reference JSON[23] structured per CDM standards.

The target output for each CSA was a structured JSON adhering to the CDM schema and capturing each relevant clause in a standardized, nested format.

Evaluated Clauses

The benchmarking focus was on five clauses: base currency, eligible currency, MTA, threshold and rounding. The appendix[i] includes a table illustrating each clause, along with examples.

Evaluated LLMs

Multiple LLMs were tested to determine their suitability for extracting and standardizing CSA clauses. The evaluation included both proprietary LLMs and open-source LLMs. This approach allowed the trade-offs between accessibility and performance across the AI ecosystem to be assessed. Table 1 lists the LLMs tested.

Table 1: LLMs Tested

	LLM	Open Source	Size	Vendor
1	GPT-4o	N	L	OpenAI
2	GPT-o1	N	L	OpenAI
3	Claude 3.7 Sonnet	N	L	Anthropic
4	Claude 3.5 Sonnet	N	L	Anthropic
5	Claude 3 Opus	N	L	Anthropic
6	Llama 3.3 70B	Y*	S	Meta
7	DeepSeek-R1	Y	S	DeepSeek AI
8	Nova Pro	N	L	AWS

*Although Llama is often referred to as open source, its license imposes restrictions on certain types of use and redistribution, preventing it from being fully open source in the traditional sense

Detailed information on the LLMs used is provided in the appendix[ii].

[23] JSON is short for JavaScript Object Notation. It is a way to store and share data in a simple format that is easy for both people and computers to understand

Prompting the LLMs

Two important categories of information were included in the prompts to the LLMs. First, the CDM JSON schema and related information were provided in all testing scenarios to ensure LLMs comprehended the required output format. Second, each LLM was assessed under two distinct conditions to evaluate the impact of CSA-specific information on accuracy.

- **With CSA-specific information:** In this scenario, LLMs were given specific CSA details, variations and examples of how particular clauses generally appear in legal documentation. This approach added specialized domain data in the context of specific clauses in the CSA.

- **Without CSA-specific information:** In this scenario, CSA-specific information was withheld. This approach assessed the ability of the LLMs to extract and interpret clauses based solely on their own internal knowledge and general language understanding.

This enabled the influence of domain knowledge on LLM performance to be isolated and measured while keeping output requirements consistent. For transparency and reproducibility, representative examples of both prompt types are included in the appendix[iii].

Evaluation Metrics

The accuracy of the LLMs in extracting and digitizing the tested clauses was primarily measured by the standardized CDM JSON representation. The following questions were asked:

- Did the LLM extract each target clause?
- Did it capture the correct numeric values or text?
- Was the CDM JSON structure valid and correctly aligned with the schema?

These were combined into an overall per-document score (ie, percentage of clauses accurately captured in proper JSON).

Procedure

A standardized approach was used to evaluate LLM performance across all test cases. The following procedural framework was implemented:

LLM Configuration and Setup

To ensure deterministic and precise outputs suitable for legal contract analysis, the LLMs were configured to produce consistent and predictable results by minimizing output randomness and prioritizing high-confidence predictions, which is critical for accurate clause extraction in legal documents[24].

Extraction and Digitizing

For each document, the LLMs were tasked with:

- Extracting the five target clauses (MTA, threshold, rounding, base and eligible currency);
- Extracting relevant values and clause texts; and
- Digitizing the extracted information according to CDM JSON schema requirements.

[24] Many LLMs have settings to control output predictability. Temperature controls randomness in generation and lower values produce more focused responses. Top-p determines the range of probable tokens considered for selection. The study used temperature of 0 to eliminate randomness by always selecting the most probable token, while using top-p of 1 to ensure all possible tokens remain available for consideration, creating deterministic outputs ideal for precise legal document extraction

Validation Protocol

A multi-stage validation process was implemented:

- Automated schema validation to verify JSON structural compliance;
- Clause-by-clause comparison against human-annotated ground truth;
- Value extraction accuracy assessment for numeric and textual components; and
- Human review and validation of the comparison results.

This procedural framework enabled performance differences specifically related to domain knowledge to be assessed, while controlling other variables in the information retrieval and representation process.

Limitations and Assumptions

- **LLM version constraints:** The evaluation represents a snapshot of LLM capabilities at the time the study was done. Considering the rapid advancement cycle of LLMs, which involves frequent updates and architectural improvements, the performances of the LLMs may evolve significantly after publication.

- **Scope delimitation:** This study intentionally limited its analysis to five clauses within CSAs: MTA, threshold, rounding, base currency and eligible currency. While these are critical components, the performance metrics observed cannot be used to predict accuracy across the full spectrum of CSA provisions without further testing.

- **Document processing assumptions:** The methodology assumes access to accurately extracted text from legal documents. The PDF parsing challenge was intentionally avoided to focus on the performance of clause extraction. Real-world implementations would need to tackle document structure extraction and optical character recognition quality issues that could impact the overall performance of digitization.

- **Prompt engineering:** A consistent set of prompts were used across all tested LLMs to ensure a fair comparison. However, prompt engineering is continually evolving, and an iterative process and further refinement of the prompts could potentially improve the accuracy of extraction and digitization. The results should be interpreted as a comparison given the same information rather than the maximum accuracy of each LLM.

- **Generalizability considerations:** While the findings provide valuable insights on CSA analysis, caution should be exercised when extrapolating these results to other legal domains. Different legal document types exhibit unique linguistic patterns, structural conventions and domain-specific terminology that can significantly impact LLM performance dynamics.

3. BENCHMARKING RESULTS

3.1 Results

Analysis of the five targeted clauses reveals distinct performance patterns among different LLMs. This section describes these findings in detail. A consistent set of prompts detailing the CDM metadata and specifying the required JSON structure was used across all tested LLMs to ensure a fair comparison. However, further refining the prompting techniques on individual LLMs could further increase accuracy. The results should be interpreted as a comparison given the same information rather than the maximum accuracy of each LLM.

Figure Legend for All Charts

Color Coding

- Purple bars indicate those LLMs that received only CDM information (ie, prompts detailing the CDM metadata and specifying the required JSON structure), excluding CSA-specific information. These LLMs are also marked with an asterisk (*).
- Blue bars represent those LLMs that received both CDM and CSA-specific information (ie, specific CSA details, variations and examples of how particular clauses generally appear in legal documentation).

Bar Patterns

- Solid bars indicate proprietary LLMs.
- Patterned bars indicate open-source LLMs.

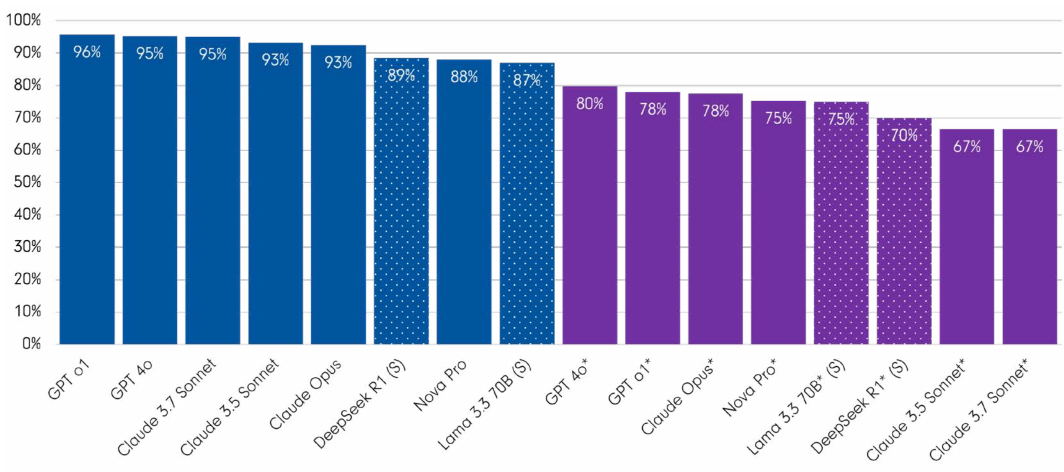
LLM Size

- LLMs marked with an (S) are considered small LLMs, while others are considered large. For the purposes of this study, an LLM's size is defined by its active parameters[25] used during a single call or inference. LLMs that use 70 billion active parameters or fewer are classified as small, with those exceeding that threshold classified as large. The appendixii shows the list of LLMs and details.

[25] Parameters are the 'knobs' (internal weights and biases) inside an LLM that get adjusted during training. These knobs help the LLM recognize and generate language patterns. More parameters generally mean the LLM can capture more detail about a language, allowing it to handle complex or unusual wording. Active parameters are a subset of an LLM's total parameters used during any single call/inference

Overall Performance Analysis

Chart 1: Overall Performance

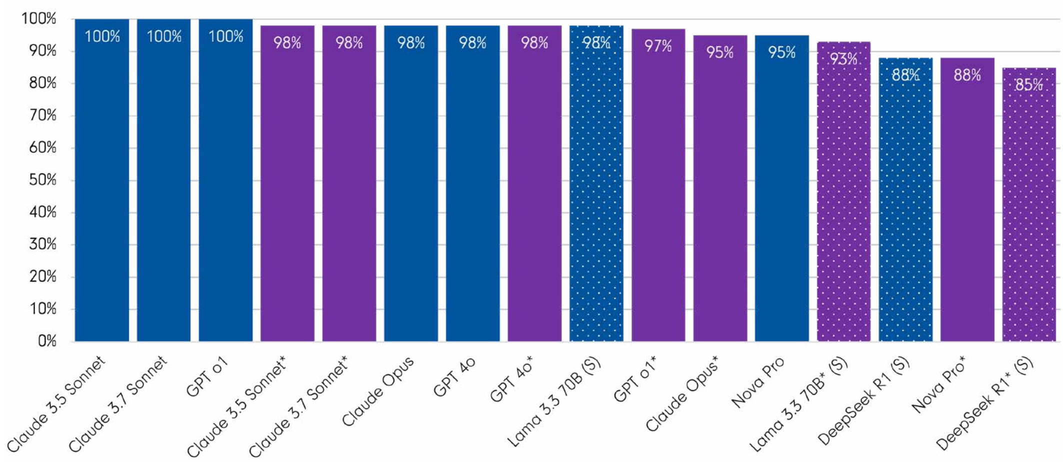


The total accuracy measurements show all LLMs improve when provided with CSA-specific information, with larger and proprietary models typically achieving better baseline performance without CSA-specific information, particularly LLMs like GPT-o1 and GPT-4o. However, some smaller open-source models (DeepSeek R1) achieve substantial or even larger percentage gains compared to some proprietary models when provided with CSA-specific information. In all instances, LLMs (both proprietary and open source) provided with CSA-specific information outperform all LLMs without CSA-specific information.

Performance varied across different clauses, even among the best-performing LLMs. This variation suggests some clauses require more nuanced interpretation than others, especially threshold and MTA clauses. The accuracy improvement observed when CSA-specific information was included highlights the essential role of domain-specific information in specialized legal contexts.

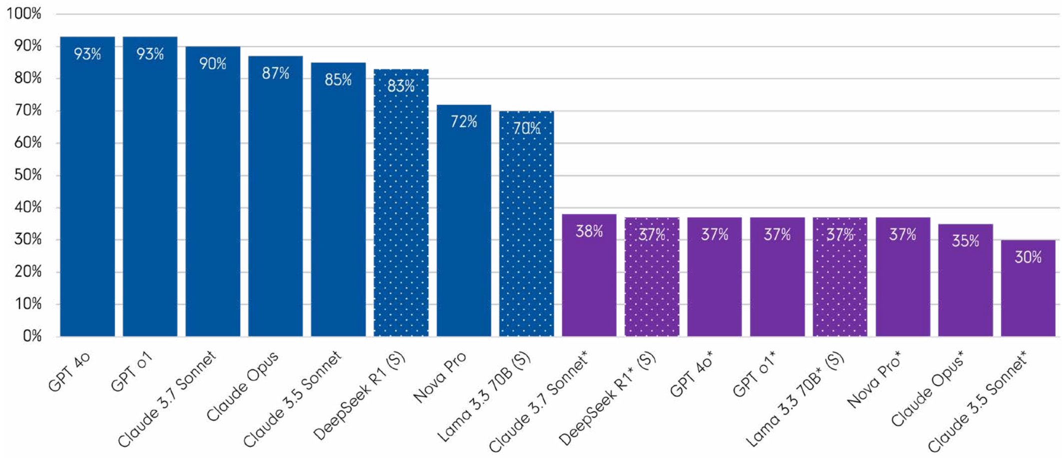
Performance by Clause Type

Chart 2: Base and Eligible Currency Clauses



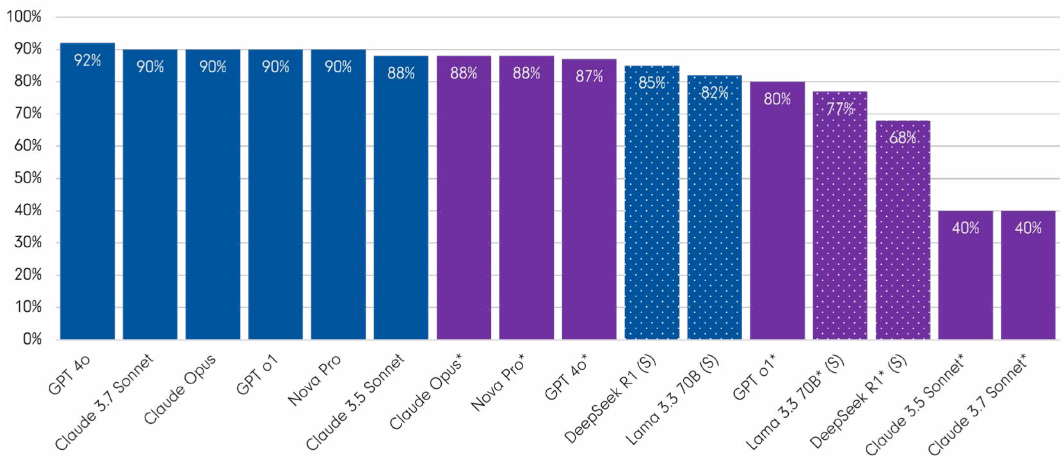
Almost all LLMs showed high accuracy in extracting base and eligible currency information. This strong performance likely stems from the standardized phrasing commonly used in these clauses, which includes direct references to specific currency types (USD, GBP, EUR). The consistent formatting of currency references resulted in smaller differences between LLMs with and without CSA-specific information compared to more complex clauses. The repeated occurrence of specific currency terminology made these clauses relatively easy for LLMs to extract.

Chart 3: Minimum Transfer Amount Clauses



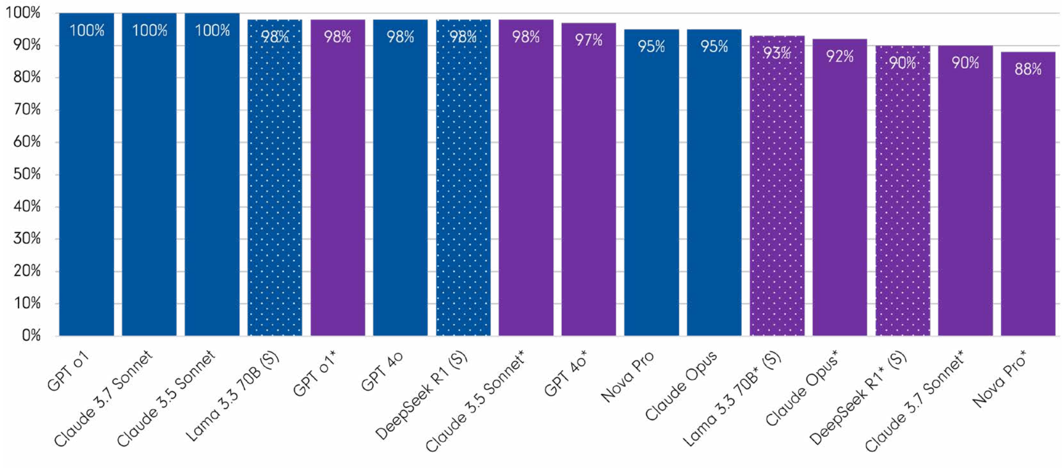
Accuracy scores for MTA clauses showed considerably more variation across LLMs. The linguistic complexity of these clauses presented challenges, particularly because their phrasing can overlap with threshold-related language, which can lead to potential confusion. All LLMs provided with CSA-specific information demonstrated substantial improvements in accuracy. Without CSA-specific information, LLMs frequently conflated references to 'minimum transfer amount' with thresholds or other numeric contractual terms.

Chart 4: Threshold Clauses



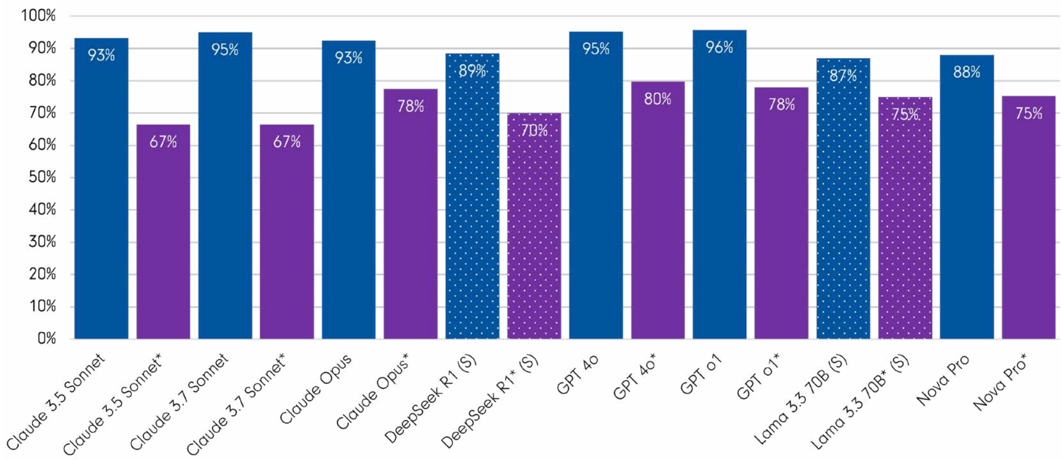
Variability in phrasing caused similar issues for threshold clauses, especially in test scenarios where CSA-specific information was not provided. Accuracy improved when LLMs received clear data on threshold variations and terms like 'threshold' and 'threshold amount' were expressed as two distinct concepts. The various threshold terms underscore the need for CSA-specific guidance to ensure precise extraction.

Chart 5: Rounding Clauses



Rounding clauses typically use direct and unambiguous language (eg, "rounded up to the nearest $1,000"), which leads to lower confusion rates across LLMs. The consistent presentation of rounding instructions in most CSA documents resulted in high accuracy scores throughout the test suite. Although LLMs still benefited from CSA-specific information, the performance gap between LLMs with and without CSA-specific information was less pronounced compared to threshold or MTA clauses.

Chart 6: Impact of Domain Knowledge



Proprietary LLMs generally start from higher baselines without CSA-specific information, and all tested LLMs (both proprietary and open source) demonstrate significant improvements once provided with CSA-specific information. This consistent gain across all evaluated LLMs highlights the universal importance of domain information, regardless of the LLM's size or architecture.

Key Observations

The analysis revealed several important insights.

- Domain knowledge boosts accuracy: Providing LLMs with CSA-specific information consistently enhances performance, especially for clauses that exhibit greater linguistic complexity, such as MTAs and threshold clauses.

- Standardized phrasing is easier to extract: Clauses in CSAs that typically use standardized phrasing (base and eligible currency and rounding) are easier for LLMs to extract accurately, irrespective of whether the LLMs were prompted with CSA-specific information.

- Open-source models improve significantly with domain knowledge: Larger proprietary LLMs typically exhibit better baseline performance, but smaller open-source LLMs can also greatly benefit from CSA-specific information.

- Prompt engineering as equalizer: Prompt engineering techniques (eg, few-shot prompting) act as an essential equalizer, significantly enhancing performance across various LLMs in specialized legal and financial fields.

- Nuanced clauses remain challenging: 100% accuracy is rarely achieved, especially for more nuanced clauses, due to inherent variations in legal language, subtle distinctions between similar clauses and complex cross-referencing within documents. Further refinements in prompting and additional CSA-specific information may be needed to address these challenges, such as providing more examples and clarifying the interpretation.

Providing LLMs with CSA-specific information consistently enhances performance, especially for clauses that exhibit greater linguistic complexity, such as MTAs and threshold clauses

Future research may include fine-tuning open-source LLMs using CSA-specific data, integrating more extensive clause sets and exploring more fine-grained prompting to enhance the accuracy of extraction in complex contract analysis tasks.

3.2 LLM Strengths and Weaknesses: General Findings

3.2.1 Comprehension of Complex Language

All evaluated LLMs demonstrated the ability to interpret legal text, but larger LLMs (eg, GPT-4o and Claude Opus) often exhibited superior skills in parsing subtle or uncommon phrasing. When faced with clauses in non-standard formats, these LLMs effectively recognized their equivalence to standard language, while smaller LLMs often missed these nuances. This advantage appears to be related to the more extensive training data that exposes larger LLMs to a broader range of legal expressions. Smaller LLMs also occasionally struggled when provided with the same CSA-specific information as the larger LLMs.

3.2.2 Following Instructions and Format Compliance

Larger LLMs not only understand content more deeply but also follow prompt requirements more consistently. In contrast, smaller LLMs sometimes omitted essential punctuation or incorrectly nested output, suggesting lower reliability in formatting tasks. This distinction was crucial for generating valid JSON structures that adhered to the CDM schema.

3.2.3 Reasoning Capabilities

CSA documents often contain interdependent clauses, such as thresholds contingent on credit rating triggers, which are located elsewhere in the document. The capacity to reason across document sections and connect related information was markedly stronger in larger LLMs. This cross-document reasoning was often absent in smaller LLMs, which typically extracted base thresholds while ignoring conditional language.

3.2.4 Hallucination and Factuality

While LLMs can produce hallucinations, this was relatively uncommon in the tightly constrained extraction tasks. The main issue was the misidentification of clauses. However, the risk of an output containing incorrect or fabricated details indicates that human review remains essential for verifying the accuracy of legal outputs, regardless of LLM performance.

The risk of an output containing incorrect or fabricated details indicates that human review remains essential for verifying the accuracy of legal outputs, regardless of LLM performance

3.2.5 Speed and Efficiency

Smaller LLMs (ie, Llama 3.3 70B) completed extraction tasks more rapidly, typically in under 10 seconds. Larger proprietary LLMs (ie, Claude Opus) often required twice as long, partially due to their more comprehensive reasoning processes.

3.2.6 Transparency and Explanations

LLMs such as GPT-4o, Deepseek R1 and Claude Opus exhibited a superior ability to clarify their clause identification methodology and reasoning, offering valuable features for auditability. This self-explanation capability – although optional in most cases – provides significant value for validating LLM decisions in financial environments where transparency is critical.

4. MULTI-AGENT (AGENTIC) FRAMEWORK FOR CSA EXTRACTION

4.1 Introduction to the Agentic Approach

In parallel with the benchmarking study, a novel agentic approach was developed to enable LLMs to process CSAs at scale. While this agentic approach was not directly used in the benchmarking study, it applies the same modular prompting techniques to address the extraction and digitization challenges.

The framework utilizes multiple specialized LLM agents, guided by a central orchestrator. Each agent focuses on extracting specific clauses while maintaining the full context of the document, with the orchestrator subsequently consolidating individual outputs into comprehensive results.

This framework was developed through collaboration with Arizona State University's Artificial Intelligence Cloud Innovation Center (AI CIC), powered by Amazon Web Services (AWS). The primary objective was to demonstrate how CSA clauses could be efficiently extracted into standardized code in CDM representation.

The solution addressed the challenge of having an LLM efficiently process all CSA clauses at scale and was implemented on AI architecture built on Amazon Bedrock, AWS Lambda and AWS S3. These technologies managed document processing workflows from upload to extraction and storage of the resulting CDM JSON representation. The writeup by AI CIC and the source code can be found on the AI CIC website[26].

4.2 Single LLM vs. Agentic Framework Approach

Although a single-LLM approach may seem simpler, the agentic framework provides several significant advantages that justify the extra architectural complexity.

Prompt Efficiency and Focus

Single comprehensive prompts that cover all clause types can become unwieldy, even for larger LLMs with bigger context windows and especially for LLMs with limited context windows. The agentic framework approach allows each agent to utilize shorter, specialized prompts tailored to specific clause categories.

Improved Accuracy Through Specialization

Even state-of-the-art LLMs like GPT-4o demonstrate reduced accuracy when tasked with extracting multiple clause types simultaneously. By distributing responsibility among specialized agents, each agent focuses on its specific extraction task, resulting in consistently higher overall accuracy compared to using a single LLM for extraction methods.

Enhanced Maintainability

As standards evolve and extraction requirements change, an agentic framework allows for modular updates. Adding or modifying specialized information for a specific clause only requires the relevant agent to be adjusted, ensuring established functionality in other components remains undisturbed. This modularity aligns with software engineering best practices and accommodates evolving standards.

[26] Clause extraction and classification: ISDA, AI CIC, September 2024, smartchallenges.asu.edu/challenges/clause-extraction-and-classification-isda

Framework Flexibility and Future Proofing

The agentic framework enables the selective assignment of tasks to specific agents based on their respective strengths and capabilities. Easily identifiable clauses (ie, rounding) can be directed to smaller LLMs, while complex clauses can utilize larger LLMs with superior reasoning and linguistic capabilities. This flexibility also facilitates the seamless integration of new LLMs as they emerge.

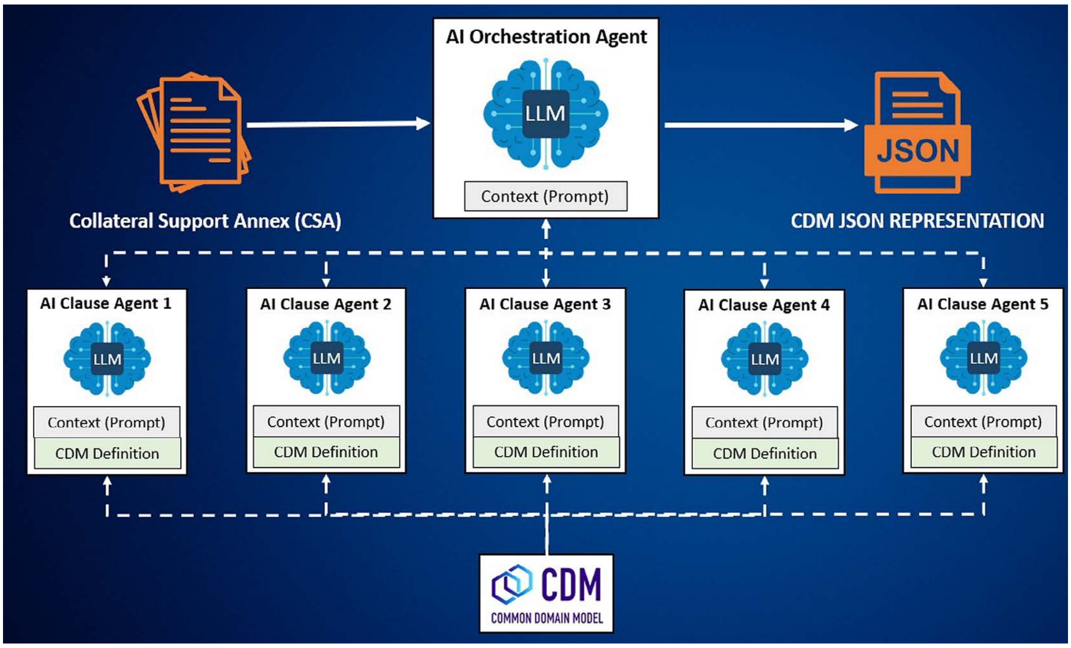
Processing Speed Through Parallelization

Unlike inherently slow approaches that utilize a single LLM, the agentic framework enables parallel processing. For large documents or high-volume workflows, this capability of parallelization greatly reduces overall processing time.

4.3 Architecture and Agent Roles

System Architecture

The framework consists of an orchestrator module and multiple specialized LLM agents, each dedicated to specific clause categories. The orchestrator, implemented as an AWS Lambda function, coordinates interactions with Amazon Bedrock for LLM processing and utilizes AWS S3 for storing documents and outputs. Each agent functions as a specialized component that receives the complete document text but returns only JSON data relevant to its assigned clause type.



Process Flow

- Document submission triggers the orchestrator via S3 event notification or API call.
- The orchestrator performs initial document preparation, including text extraction from PDF format when necessary.
- The orchestrator invokes each specialized agent, providing the complete document text along with agent-specific instructions.
- Each agent is responsible for one or more clauses defined within its action group functions. When an agent is invoked – for example, the 'rounding' agent – it calls the specific clause function and sends the customized prompt for rounding to the LLM endpoint to extract the clause information from the text.

- Agents return structured JSON and store these intermediate outputs.
- Upon completion of all agent tasks, the orchestrator consolidates the individual JSON snippets into a unified file conforming to the CDM schema.
- A final validation process ensures proper JSON formatting and consistent data types.
- The system outputs the consolidated JSON through appropriate channels (S3 storage or API response).

Agent Design Principles

Each agent functions autonomously, without a persistent state or direct communication with other agents, which preserves the system's modularity. While the CSA extraction implementation did not necessitate cross-agent data sharing, the architecture can accommodate orchestrator-mediated information exchange for more complex workflows that involve interdependent processing.

Execution Strategy

While the proof-of-concept implementation utilizes sequential agent execution for simplicity, the architecture fully supports parallel processing. In production environments, agents can operate concurrently as separate Lambda functions, with the orchestrator collecting results upon completion, substantially reducing overall processing time for complex documents.

Future refinements to the framework can include expanding the clause coverage, enriching the example set for variant identification and optimizing prompt engineering techniques to further enhance extraction accuracy and efficiency.

5. CONCLUSION AND IMPLICATIONS

The benchmarking study produced three pivotal findings.

Domain-Specific Knowledge Is Crucial

LLMs lacking CSA-specific information, as well as standard contract definitions and variations, struggled with accuracy. When LLMs were provided with CSA-specific information, such as the ISDA Documentation Taxonomy, they often achieved or exceeded 90% accuracy in extracting clauses. This highlights the significant value of ISDA's taxonomies and standardized documentation in enhancing the performance of LLMs. There is an opportunity for further research on the optimization of these resources specifically for AI interpretation.

When LLMs were provided with CSA-specific information, such as the ISDA Documentation Taxonomy, they often achieved or exceeded 90% accuracy in extracting clauses

Larger LLMs Demonstrate Advanced Nuance Handling

While larger LLMs typically demonstrated higher overall performance and advanced handling of nuances, smaller open-source alternatives also showed improvement when provided with CSA-specific information, offering a viable alternative to financial institutions with stringent data privacy requirements that necessitate on-premises deployment. However, there are trade-offs: while larger LLMs require substantial computational resources, smaller LLMs struggle with complex or heavily customized contracts.

Multi-agent Architecture Enhances Reliability

The agentic framework proof-of-concept developed in collaboration with Arizona State University's AI CIC represents a particularly promising approach. By assigning specialized tasks to different AI agents orchestrated by a central coordinator, this architecture improves processing efficiency and reduces management overheads. Each agent's focused expertise improves accuracy, while the modular design enhances auditability as each component's output can be independently verified.

Additional Findings

- The analysis showed that larger LLMs like GPT-4o and Claude Opus outperform smaller LLMs in understanding complex legal language, recognizing non-standard clauses and handling domain-specific terminology. They also excel in following instructions and formatting outputs correctly, which is crucial for structured tasks like CDM JSON generation.

- Larger LLMs' reasoning capabilities allow them to connect interdependent clauses across documents, a skill often lacking in smaller LLMs.

- While hallucinations were rare in controlled extraction tasks, occasional misidentification of clauses underscores the need for human oversight.

- Smaller LLMs can offer faster processing but at the cost of accuracy and reasoning depth.

- Transparency is another strength of larger LLMs, as they can explain their decisions more effectively, providing valuable auditability in legal and financial applications.

Future Research Directions

There are several areas where further research may be valuable. Advanced retrieval-augmented generation techniques could enable LLMs to reference authoritative industry resources and regulatory libraries during analysis, potentially reducing factual errors. Research into how organizations could align contract analytics and compliance checks with standardized data models could investigate whether such integration would improve operational efficiency and risk management.

Several other potential research avenues merit industry-wide exploration:

Documentation and LLM interpretability research: Financial institutions and standards organizations could investigate whether modifications to existing documentation, such as structured metadata tagging, semantic annotations or optimized formatting, might enhance machine readability. Research could assess which specific documentation changes would most effectively improve LLM analysis outcomes across the derivatives market.

Data model integration research: The financial services industry might benefit from additional research examining methodologies for effectively connecting LLMs with standardized data representation, such as current work being conducted on the CDM by FINOS[27]. This could help establish best practices that market participants could adopt when implementing such solutions.

Industry LLM solution analysis: As technology providers develop solutions that combine industry standards with LLMs, the market could benefit from research identifying common implementation challenges and success factors. This might help all stakeholders better understand the practical considerations involved in deploying these solutions.

Quality assurance methodology research: Market participants could explore frameworks for validating LLM-based contract analysis, potentially establishing industry benchmarks to ensure accuracy and consistency across implementations.

As generative AI technology evolves, it may significantly impact how derivatives documentation is drafted, negotiated and managed throughout the financial services industry. Organizations that strategically incorporate domain-specific knowledge, standardized data models and well-designed AI workflows can achieve operational benefits while maintaining necessary accuracy. The industry stands to benefit from thoughtful exploration of how these technological advances might best serve the derivatives market.

[27] AI Strategic Initiative Series: Applying LLMs to Simplify CDM, FINOS, www.finos.org/blog/ai-strategic-initiative-series-applying-llms-to-simplify-cdm

APPENDIX

i Clauses Tests

Table A illustrates each clause tested with an example credit support annex (CSA) (2016 VM CSA - NY Law) excerpt and corresponding Common Domain Model (CDM) representation.

Table A: Tested Clause and CDM Representation

Clause	CSA content	CDM JSON Representation

Base and Eligible Currency	Paragraph 12. As used in this Annex:
	“Base Currency” means the currency specified as such in Paragraph 13.
	“Eligible Currency” means each currency specified as such in Paragraph 13, if such currency is freely available.
	
	Paragraph 13(a)
	(i) “Base Currency” means: United States Dollars, unless otherwise specified here: “Base Currency” means US Dollars.
	(ii) “Eligible Currency” means the Base Currency and each other currency specified here: US Dollars.
		{
		"agreementTerms": {
		"agreement": {
		
		"creditSupportAgreementElections": {
		"baseAndEligibleCurrency": {
		"baseCurrency": "USD",
		"eligibleCurrency": [
		"USD"
		],
		
		"eligibleCurrencyInclBaseCurrency": true
		}
		}
		}
		}
		}

Minimum Transfer Amount (MTA)	Paragraph 12. Definitions
	“Minimum Transfer Amount” means, with respect to a party, the amount specified as such for that party in Paragraph
	13; if no amount is specified, zero.
	
	Paragraph 13(vii) Minimum Transfer Amount
	
	(A) “Minimum Transfer Amount” means with respect to Party A: US Dollars 5,000,000.
	“Minimum Transfer Amount” means with respect to Party B: US Dollars 5,000,000.
		"agreementTerms": {
		"agreement": {
		
		"creditSupportAgreementElections": {
		"minimumTransferAmount": [
		{
		"mtaType": {
		"fixedAmount": {
		"amount": 5000000,
		"currency": "USD",
		"party": "PARTY_1"
		}
		}
		},
		{
		"mtaType": {
		"fixedAmount": {
		"amount": 5000000,
		"currency": "USD",
		"party": "PARTY_2"
		}
		}
		}
		]
		}
		}
		}
		}

Continues on next page »

« Continued from previous page

Clause	CSA content	CDM JSON Representation

Threshold	Paragraph 13(iv) Thresholds.
	(B) “Threshold” means with respect to Party A: zero
		{
		"agreementTerms": {
		"agreement": {
		
		"creditSupportAgreementElections": {
		"threshold": [
		{
		"thresholdType": {
		"fixedAmount": {
		"amount": 0,
		"currency": "USD",
		"party": "PARTY_1"
		}
		}
		}
		]
		}
		}
		}
		}

Rounding	Paragraph 13(vii) Minimum Transfer Amount
	
	(B) Rounding. The Delivery Amount (VM) and the Return Amount (VM) will be rounded down to the nearest integral multiple of US Dollars 10,000
		{
		“agreementTerms”: {
		“agreement”: {
		
		“creditSupportAgreementElections”: {
		“creditSupportObligations”: {
		“rounding”: {
		“deliveryAmount”: 10000,
		“deliveryDirection”: “UP”,
		“returnAmount”: 10000,
		“returnDirection”: “DOWN”,
		“currency”: “USD”
		}
		}
		}
		}
		}
		}

ii Details on the LLMs Used

The study defined LLM size based on the number of active parameters, which are those utilized during a single inference or call. Active parameters represent a subset of an LLM's total parameters, reflecting the portion of the model engaged during runtime. In this context, LLMs that use 70 billion or fewer active parameters are classified as small, while those exceeding this threshold are considered large. For proprietary models that do not disclose exact parameter counts, estimates were derived using reported usage patterns and industry benchmarks.

Table B: LLMs Used for the Study

	LLM	Open Source	Vendor	Release Date	Size	Description
1	GPT-4o	N	OpenAI	Nov-24	Large	A proprietary, multimodal model that handles text, images and sound. Exact parameter count undisclosed but estimated to be ~200b parameters. Context window of 128k tokens. Classified as large based on estimated parameters
2	GPT-o1	N	OpenAI	Dec-24	Large	Another proprietary model by OpenAI with no public parameter count but estimated to be ~200b parameters. It emphasizes advanced reasoning and chain-of-thought approaches. Classified as large based on estimated parameters
3	Claude 3.7 Sonnet	N	Anthropic	Feb-25	Large	A proprietary Anthropic model with undisclosed parameters (estimated to be over 100b parameters). It is reported to feature 'hybrid reasoning' and an extended context window (often cited at 100k+ tokens). Classified as large based on estimated parameters
4	Claude 3.5 Sonnet v2	N	Anthropic	Oct-24	Large	Part of Anthropic's Claude family, details on its exact parameter count are not publicly available but estimated to have over 175b parameters. Vendor-stated context window can reach ~200k tokens. Classified as large based on estimated parameters
5	Claude 3 Opus	N	Anthropic	Mar-24	Large	Another Claude series model with enhanced capacity for longer context and complex workflows. Exact parameter count remains undisclosed but estimated to have over 100b parameters. Classified as large based on estimated parameters
6	Llama 3.3 70B	Y*	Meta	Dec-24	Small	An open-source 70b parameter model. In this study's framework, 70b active parameters is the upper bound for 'small'. Provides a sufficiently large context window (reportedly ~128k tokens) for legal tasks

Continues on next page »

« Continued from previous page

7	Nova Pro	N	AWS	Dec-24	Large	A proprietary, multimodal foundation model from AWS with supported context window of 300k tokens. Exact parameter count is undisclosed but estimated to be ~90b parameters. Classified as large based on estimated parameters
8	DeepSeek-R1	Y	DeepSeek AI	Jan-25	Small	Uses a Mixture of Experts (MoE) architecture, with a total of 671b parameters. However, only 37b parameters are activated at a time. Classified as small in this framework because its active parameter usage does not exceed 70B

iii Sample Prompts Used

Non-CSA-Specific Prompt for Rounding
ROUNDING_PROMPT = """\

Background Information

A Rounding clause in a Credit Support Annex (CSA) or ISDA Master Agreement specifies how Delivery Amounts and Return Amounts are rounded for operational purposes. This clause is important for practical implementation of collateral calculations:

Task Description

Identify Rounding provisions in a CSA (or ISDA Master Agreement) and then represent it in Common Domain Model (CDM) JSON using the rules below.
Carefully read all sections, schedules, and annexes of the agreement to locate any references to Rounding. Then follow these steps and instructions precisely.

Document Processing Guidelines

When analyzing a document for Rounding clauses:
1. Comprehensive Review: Thoroughly review the entire document, including all schedules and annexes, even if they appear after the main body of the agreement.

CDM Fields Reference

----

agreementTerms (1..1): Specification of the standard set of terms that define a legal agreement.
  agreement (1..1): Specification of the standard set of terms that define a legal agreement.
    creditSupportAgreementElections: Specification of the standard set of terms that define a legal agreement.
      creditSupportObligations (1..1): The Credit Support Obligations applicable to the agreement.
        rounding (0..1): The rounding methodology applicable to the Delivery Amount and the Return Amount in terms of nearest integral multiple of Base Currency units.
          currency (1..1): The currency in which the Delivery Amount and Return Amount rounding amounts are specified. [ISO currency code]
          deliveryAmount (1..1): The nearest integral multiple of Base Currency units to which the Delivery Amount will be rounded in accordance with the specified rounding direction. [number]
          deliveryDirection (1..1): The rounding rule applicable to the Delivery Amount (which can be (i) up to nearest; (ii) down to nearest). [UP,DOWN]
          other (0..1): Utilised where the clause data structure is not able to capture a material aspect

of the clause. [string]
returnAmount (1..1): The nearest integral multiple of Base Currency units to which the Return Amount will be rounded in accordance with the specified rounding direction. [number]
returnDirection (1..1): The rounding rule applicable to the Return Amount (which can be (i) up to nearest; (ii) down to nearest). [UP,DOWN]

Explanation:
(1..1): Required to be present in the JSON output.
(0..1): Optional to be present in the JSON output. Maximum 1 field can be present.
[string]: The value can be any string value.
[number]: The value can be any number value.
[ISO currency code]: The value must be a valid ISO currency code.
[UP,DOWN]: The value can be UP or DOWN.

Ensure the final JSON output includes all necessary parent objects and follows the exact structure, from agreementTerms down to the lowest level of the JSON object, including restrictions on how often the fields can be present.
"""

CSA-Specific Prompt for Rounding
ROUNDING_PROMPT = """\

Background Information

A Rounding clause in a Credit Support Annex (CSA) or ISDA Master Agreement specifies how Delivery Amounts and Return Amounts are rounded for operational purposes. This clause is important for practical implementation of collateral calculations:
- Delivery Amount: The amount of collateral that must be posted by a party when the exposure exceeds the threshold.
- Return Amount: The amount of excess collateral that must be returned to a party.
- Rounding Direction: Typically, Delivery Amounts are rounded up (increasing the collateral obligation) and Return Amounts are rounded down (decreasing the return obligation).
- Rounding Multiples: The fixed amount to which values are rounded (e.g., to the nearest multiple of 1,000).
Rounding provisions are typically found in Paragraph 13 of the CSA under "Credit Support Obligations" or in a similar section specifying operational details.

Task Description

Identify Rounding provisions in a CSA (or ISDA Master Agreement) and then represent it in Common Domain Model (CDM) JSON using the rules below.
Carefully read all sections, schedules, and annexes of the agreement to locate any references to Rounding. Then follow these steps and instructions precisely.

Document Processing Guidelines

When analyzing a document for Rounding clauses:
1. Comprehensive Review: Thoroughly review the entire document, including all schedules and annexes, even if they appear after the main body of the agreement.
2. Location Focus:
- Look in sections titled "Rounding," "Credit Support Obligations," or "Calculations"
- Check Paragraph 13 of CSAs, particularly under subsections about calculations or operational

procedures
 - Examine any sections discussing Delivery Amount or Return Amount
3. Identification Process:
 - Identify both the Delivery Amount rounding and Return Amount rounding
 - Note the rounding direction for each (UP or DOWN)
 - Identify the currency and fixed amount for rounding
 - Determine if the rounding applies unconditionally or varies based on conditions
----
Classification Variants

Variant 1: Delivery Amount Rounded Up / Return Amount Rounded Down
 The Delivery Amount is Rounded Up and the Return Amount is Rounded Down to the nearest multiple of a Fixed Amount in all cases without any conditions.
 Example from document: Rounding. (i) The Delivery Amount will be rounded up to the nearest integral multiple of EUR 100,000. (ii) The Return Amount will be rounded down to the nearest integral multiple of EUR 100,000.
 Example CDM json format:

```
{
  "agreementTerms": {
    "agreement": {
      "creditSupportAgreementElections": {
        "creditSupportObligations": {
          "rounding": {
            "currency": "EUR",
            "deliveryAmount": 100000,
            "deliveryDirection": "UP",
            "returnAmount": 100000,
            "returnDirection": "DOWN"
          }
        }
      }
    }
  }
}
```

Variant 2: Other Rounding
 - Any variant not covered by Variant 1, including:
 - Cases where rounding amounts differ based on conditions
 - Non-standard rounding directions (e.g., both rounded up or both rounded down)
 - Complex rounding rules
 - Rounding that depends on currency type or transaction type
 Example from document: Rounding. The Delivery Amount will be rounded up to the nearest integral amount of $50,000. The Return Amount will be rounded down to the nearest integral amount of$50,000 unless the Secured Party's Exposure at the time of the demand made pursuant to Paragraph 3(b) is equal to or less than zero, in which case the Return Amount is notsubject to rounding.
 Example CDM json format:

```
{
  "agreementTerms": {
    "agreement": {
      "creditSupportAgreementElections": {
        "creditSupportObligations": {
          "rounding": {
```

"other": "The Return Amount will be rounded down to the nearest integral amount of $50,000 unless the Secured Party's Exposure at the time of the demand made pursuant to Paragraph 3(b) is equal to or less than zero, in which case the Return Amount is notsubject to rounding"
}
}
}
}
}
}

Parsing Guidelines and Edge Cases
----
1. Identifying Standard Rounding
  - For Variant 1 classification, ensure ALL of the following conditions are met:
    - Delivery Amount is ALWAYS rounded UP
    - Return Amount is ALWAYS rounded DOWN
    - Both amounts are rounded to the nearest multiple of a FIXED amount
    - Rounding is UNCONDITIONAL (no conditions based on amount, currency, etc.)
    - Both amounts use the SAME currency for rounding
2. Common Non-Standard Rounding (Variant 2)
  - Watch for these patterns that indicate Variant 2:
    - Different rounding multiples for Delivery vs Return amounts
    - Rounding that changes based on thresholds or amounts
    - Currency-dependent rounding
    - Time-dependent rounding
    - Different rounding for different types of transactions
    - Non-standard rounding directions (both up, both down, or nearest)
3. Implicit Rounding
  - Some agreements may not explicitly state "rounded up" or "rounded down" but use equivalent language
  - "Not less than the nearest multiple" = rounded up
  - "Not more than the nearest multiple" = rounded down
  - "Nearest multiple" without direction specified = Variant 2
4. No Rounding Specified
  - If the document states explicitly that amounts will NOT be rounded, do not generate JSON output
  - If rounding is not mentioned at all, do not assume any default rounding and do not generate JSON output
----
Error Handling and Validation
----
1. Missing Information:
  - If rounding is mentioned but direction is not specified, use Variant 2
  - If currency is not specified, look for the Base Currency in the agreement
  - If no rounding multiple is specified, do not generate JSON output
2. Validation Checks:
  - Ensure currency codes are valid ISO codes
  - Verify that direction values are either "UP" or "DOWN"
  - Ensure rounding amounts are represented as numbers, not strings
3. Special Cases:
  - If the document specifies different rounding for different parties, use Variant 2

- If rounding applies only to certain types of collateral, use Variant 2
- If rounding changes based on market conditions, use Variant 2

CDM Fields Reference

----

agreement (1..1): Specification of the standard set of terms that define a legal agreement.
creditSupportAgreementElections: Specification of the standard set of terms that define a legal agreement.
creditSupportObligations (1..1): The Credit Support Obligations applicable to the agreement.
rounding (0..1): The rounding methodology applicable to the Delivery Amount and the Return Amount in terms of nearest integral multiple of Base Currency units.
currency (1..1): The currency in which the Delivery Amount and Return Amount rounding amounts are specified. [ISO currency code]
deliveryAmount (1..1): The nearest integral multiple of Base Currency units to which the Delivery Amount will be rounded in accordance with the specified rounding direction. [number]
deliveryDirection (1..1): The rounding rule applicable to the Delivery Amount (which can be (i) up to nearest; (ii) down to nearest). [UP,DOWN]
other (0..1): Utilised where the clause data structure is not able to capture a material aspect of the clause. [string]
returnAmount (1..1): The nearest integral multiple of Base Currency units to which the Return Amount will be rounded in accordance with the specified rounding direction. [number]
returnDirection (1..1): The rounding rule applicable to the Return Amount (which can be (i) up to nearest; (ii) down to nearest). [UP,DOWN]

Explanation:
(1..1): Required to be present in the JSON output.
(0..1): Optional to be present in the JSON output. Maximum 1 field can be present.
[string]: The value can be any string value.
[number]: The value can be any number value.
[ISO currency code]: The value must be a valid ISO currency code.
[UP,DOWN]: The value can be UP or DOWN.

Ensure the final JSON output includes all necessary parent objects and follows the exact structure, from agreementTerms down to the lowest level of the JSON object, including restrictions on how often the fields can be present.

Important Notes

----

1. Variant Classification Priority:
- Only use Variant 1 when rounding is completely standard and unconditional
- ANY deviation from standard rounding should be classified as Variant 2
- When in doubt, use Variant 2 and include the full text
2. No Rounding Rule:
- If the document explicitly states that Delivery Amount and Return Amount will NOT be rounded, do not generate JSON output
- If rounding is not mentioned at all in the document, do not generate JSON output
3. Complete Text Capture:
- For Variant 2, include the complete rounding provision text in the "other" field
- Do not truncate or summarize the text, as important details may be lost
4. Currency Consistency Check:
- Verify that the currency used for rounding matches the Base Currency or is explicitly specified
- If different currencies are used for different types of rounding, use Variant 2
""""

ABOUT ISDA

Since 1985, ISDA has worked to make the global derivatives markets safer and more efficient. Today, ISDA has over 1,000 member institutions from 76 countries. These members comprise a broad range of derivatives market participants, including corporations, investment managers, government and supranational entities, insurance companies, energy and commodities firms, and international and regional banks. In addition to market participants, members also include key components of the derivatives market infrastructure, such as exchanges, intermediaries, clearing houses and repositories, as well as law firms, accounting firms and other service providers. Information about ISDA and its activities is available on the Association's website: www.isda.org.
Follow us on LinkedIn and YouTube.