Finance and Economics Discussion Series

Federal Reserve Board, Washington, D.C.
ISSN 1936-2854 (Print)
ISSN 2767-3898 (Online)

Can LLMs Improve Sanctions Screening in the Financial System? Evidence from a Fuzzy Matching Assessment

Jeffrey S. Allen, Max S. S. Hatfield

2025-092

Please cite this paper as:
Allen, Jeffrey S., and Max S. S. Hatfield (2025). "Can LLMs Improve Sanctions Screening in the Financial System? Evidence from a Fuzzy Matching Assessment," Finance and Economics Discussion Series 2025-092. Washington: Board of Governors of the Federal Reserve System, https://doi.org/10.17016/FEDS.2025.092.

Can LLMs Improve Sanctions Screening in the Financial System? Evidence from a Fuzzy Matching Assessment

Jeffrey S. Allen*† Max S. S. Hatfield*

September 2025‡

Abstract

We examined the performance of four families of large language models (LLMs) and a variety of common fuzzy matching algorithms in assessing the similarity of names and addresses in a sanctions screening context. On average, across a range of realistic matching thresholds, the LLMs in our study reduced sanctions screening false positives by 92 percent and increased detection rates by 11 percent relative to the best-performing fuzzy matching baseline. Smaller, less computationally intensive models from the same language model families performed comparably, which may support scaling. In terms of computing performance, the LLMs were, on average, over four orders of magnitude slower than the fuzzy methods. To help address this, we propose a model cascade that escalates higher uncertainty screening cases to LLMs, while relying on fuzzy and exact matching for easier cases. The cascade is nearly twice as fast and just as accurate as the pure LLM system. We show even stronger runtime gains and comparable screening accuracy by relying on the fastest language models within the cascade. In the near term, the economic cost of running LLMs, inference latency, and other frictions, including API limits, will likely necessitate using these types of tiered approaches for sanctions screening in high-velocity and high-throughput financial activities, such as payments. Sanctions screening in slower-moving processes, such as customer due diligence for account opening and lending, may be able to rely on LLMs more extensively.

Keywords: Large Language Models, Sanctions Screening, Model Cascading

*Federal Reserve Board. The views expressed in this paper are solely those of the authors and should not be interpreted as reflecting the views of the Federal Reserve.

†Corresponding author. E-mail: jeff.allen@frb.gov.

‡We would like to thank Dave Mills, Sonja Danburg, Marc Rodriguez, Jillian Mascelli, Seung Jung Lee, Nitish Sinha, Sarah Liebschutz, Deirdre Ryan, and Koko Ives of the Federal Reserve Board, Anne Hansen of the Richmond Fed, Manish Agarwal of the Boston Fed, and participants in the 2025 System Payment Researchers conference at the Federal Reserve Bank of Chicago for their feedback.

1 Introduction

Financial institutions and payment service providers are prohibited from extending financial services to and processing transactions on behalf of entities that have been sanctioned by relevant authorities, such as the Office of Foreign Assets Control (OFAC).[1] High transaction volumes generally compel financial intermediaries to adopt automated sanctions screening systems that compare names, addresses, and other relevant details in financial applications and payment messages to information contained in publicly available sanctions lists. These systems often use approximate string ("fuzzy") matching algorithms to mitigate challenges associated with complex variations in the way names, addresses, and localities are captured in the financial system (Fadavi, 2023).[2]

Even with fuzzy logic, noisy string representations leave sanctions screening systems vulnerable to costly inaccuracies. False negatives may lead to monetary penalties and other enforcement actions from sanctions authorities. False positives increase compliance burden, slow down transactions due to manual reviews, and can harm customer experience. Of course, weak sanctions screening systems also have broader national security implications, as threat actors are able to move money more seamlessly.

Against this backdrop, we ask whether large language models (LLMs) are better than traditional fuzzy matching methods at handling complex sanctions screening cases. They are compelling supplements or alternatives for at least two reasons. First, the architecture underlying LLMs (Vaswani et al., 2017) enables them to learn semantic, syntactic, and contextual patterns that can help sort through subtle but meaningful string deviations. Second, they encode some world knowledge that can aid in evaluating names, addresses, and other transaction party details. For example, they appear to know the basic street layout of many major cities. They also know many corporate structure suffixes and their common abbreviations. While these properties have the potential to drive gains in screening matches, they are especially likely to help rule out legitimate transaction parties whose identifying information and location details are similar to those of sanctioned entities.

We investigated our research question by evaluating the relative performance of LLMs and fuzzy matching algorithms in assessing the similarity of hypothetical transaction party details. To facilitate the assessment, we used GPT-4o to help generate a test dataset based on the sanctions list published by OFAC. The test data contain the names and addresses of sanctioned entities from ten countries, along with subtle and plausible deviations from the original strings that simulate difficult sanctions screening cases. We then tested the string-matching capabilities of language models from four different model families: Claude, Llama, Mistral, and Nova. For each model family, we tested one larger and one smaller model, which we refer to as LLMs and small language

[1] Other sanctions authorities include the UK's Office of Financial Sanctions Implementation, the UN Security Council, and the European Union.

[2] OFAC's own sanctions list search tool uses two classical fuzzy matching techniques, edit distance and Jaro-Winkler, as well as a phonetic matching algorithm, to aid in screening (OFAC, 2021).

models (SLMs), respectively. We compared the average classification performance of the LLMs and SLMs to several fuzzy matching methods: Levenshtein, Jaro-Winkler, token sort, and token set. We augmented the fuzzy methods by implementing a rigorous textual pre-processing and normalization routine. We also tested a more dynamic approach that selects from and weights multiple fuzzy algorithms depending on relative string length.

On average, the LLMs in our study reduced sanctions screening false positives by 92 percent and increased screening matches by 11 percent relative to the highest-performing fuzzy matching system. The SLMs performed similarly well, reducing false positives by 80 percent and increasing screening matches by the same amount as the LLMs. Overall, the significant false positive gains and meaningful improvement in matching performance are promising for sanctions screening accuracy. Indeed, fuzzy matching methods sacrifice false positives in favor of reducing false negatives. LLMs could serve as a single mechanism for balancing both priorities. Additionally, comparable performance between the LLMs and SLMs could support scalability in the financial system, as smaller models tend to be cheaper to run, have higher API limits, and are a bit faster. It is also feasible for many firms to run open-weight SLMs on proprietary hardware at scale.

The language models' improved accuracy comes with a computing performance tradeoff, though. In our study, the LLMs were, on average, over four orders of magnitude slower than the fuzzy matching algorithms. The SLMs were about 25 percent faster than the LLMs but still considerably slower than the fuzzy methods. Importantly, we did not evaluate the potentially substantial time savings that could come from the language models' ability to reduce false positives over the time-consuming manual reviews that these types of hits trigger.

To speed up the screening process and minimize the cost of invoking the language models, we propose a model cascade that leans on exact and fuzzy matching for more certain cases and escalates less certain ones to the language models for review. The cascade reduces runtime by 45 percent over the pure language model system with no loss of accuracy. Relying on the fastest language models within the cascade reduces runtime by another 50 percent with very similar screening accuracy. Integrating these models in financial activities that are characterized by very high velocities and throughput, such as payments, would likely require cascading and other types of tiered systems. Sanctions screening in slower moving activities, such as customer onboarding and due diligence, as well as some types of credit and insurance review, may be able to use language models more extensively.

The remainder of this paper proceeds as follows. In section 2, we review related work at the intersection of AI and finance. Section 3 discusses our research design, focusing on the data and models underlying the study. Section 4 presents our screening results, including descriptive properties of the model outputs and classification performance. Section 5 examines computing performance and presents our model cascade system. Section 6 concludes by summarizing the implications of this work and identifying areas for future research.

2 Related Work

This paper seeks to advance the emerging body of literature on generative AI (GenAI) in finance. Aldasoro et al. (2024) identify a range of opportunities, challenges, and financial stability implications of GenAI adoption in the four major financial activities: intermediation, insurance, asset management, and payments. Eisfeldt and Schubert (2024) examine how GenAI affects financial sector occupations and review GenAI uses in financial research. Some have explored GenAI applications in central banking and financial regulation (Araujo et al., 2024; Kazinnik and Brynjolfsson, 2025). Similar to this paper, others have focused on how GenAI can be used in more specific tasks, such as analyzing central bank communications (Dunn et al., 2024; Fischer et al., 2023; Hansen and Kazinnik, 2024; Silva et al., 2025), generating synthetic data to study rare financial events (Kazinnik, 2023), assessing financial sentiment (Zhang et al., 2023), and making cash management decisions in payment systems (Aldasoro and Desai, 2025).

The paper also relates to research on the use of AI for detecting fraud, money laundering, and security threats in the financial system. A rich body of work examines specific machine learning methods for fraud detection (West and Bhattacharya, 2016; Hilal et al., 2022). Others have proposed simulation methodologies to support modeling research in fraud (Allen, 2025) and money laundering (Altman et al., 2023) detection, mostly in retail payment systems. Recently, Desai et al. (2025) developed a machine learning system geared toward detecting anomalies from various sources in high-value payment systems. While not nearly as broad as the fraud and money laundering detection literature, a few papers, like ours, are situated in the sanctions screening domain. For example, researchers have proposed ways to enhance edit distance metrics and leverage natural language processing techniques for sanctions screening (Nino et al., 2019; Kim and Yang, 2024).

Finally, our methodology leans on ideas from model cascading (Viola and Jones, 2001) and the related areas of selective classification (El-Yaniv et al., 2010) and early exits in deep learning (Bolukbasi et al., 2017; Teerapittayanon et al., 2016). The goal of these systems is to reduce runtime, computing power, and economic cost, while minimizing loss of accuracy by sending less uncertain instances to simpler, faster models and reserving more complex cases for larger, more expensive models. Recently, scholars have extended model cascading to natural language processing (Varshney and Baral, 2022) and LLMs. The cost of LLM inference has been a significant motivating factor for the development of LLM cascades. For example, Chen et al. (2023) propose FrugalGPT, a "budget-aware" LLM cascade that channels easier queries to smaller, cheaper models. They find that FrugalGPT drives cost savings and even improved accuracy over single LLM systems for certain tasks. Nie et al. (2024) propose a dynamic cascade tailored to data streams that uses a logistic regression at the first layer and an LLM at the top layer. Like these systems, our proposed cascade relies on simple methods–specifically, fuzzy and exact matching–for less uncertain screening cases and only escalates higher uncertainty cases to the slower, more expensive language models.

3 Research Design

We compared the performance of LLMs to traditional fuzzy matching methods in assessing the similarity of hypothetical transaction party names and addresses to those of sanctioned entities. We did not attempt to reproduce a production grade sanctions screening environment. In most cases, this would involve screening incoming transaction party details against all sanctioned entities, possibly with filtering conditions to narrow down a candidate pool. Rather, we conducted pairwise comparisons of names and addresses. While smaller scale in nature than a comprehensive screening system, our study was carried out using production-grade architecture, subject to many of the same constraints of real-world sanctions screening applications. For example, our LLM-based scorers were invoked programmatically via cloud APIs and included buffering logic to accommodate service quotas. Additionally, pairwise comparisons are often one of the final steps in the sanctions screening process after a candidate pool has been narrowed down. The following sections discuss the data, LLMs, prompting strategies, and fuzzy matching comparison methods that we used in the study.

3.1 Data

Our data are derived from the sanctions lists published by OFAC (OFAC, 2025). Specifically, we used the combination of OFAC's specially designated nationals (SDN) list and the consolidated list for non-SDN sanctioned entities as of March 2025. The OFAC lists contain names, addresses, localities, aliases, and other information for sanctioned individuals and entities. In our study, we conducted pairwise comparisons of organization names and street addresses.[3] Table 1 summarizes four case types that we evaluated, using fictitious examples. We are particularly interested in the relative performance of the LLMs and fuzzy methods on the close match cases. The clear cases help us simulate a more comprehensive classification system.

Table 1: Pairwise Comparison Case Types

Case Type	ABC International Ltd.	123 Chestnut Street, NW
Clear negative	Main Street Bank	456 Magnolia Drive, SE
Negative close match	A&C International LLC	12 Chestnut Avenue, NW
Positive close match	ABC Int'l Limited	123 Chestnut St., Northwest
Clear positive	ABC International Ltd.	123 Chestnut Street, NW

The lists only contain original strings, so we needed to generate the candidate strings that served as our test data. Generating these for the clear negative and clear positive cases was straightforward. For the former, we selected a random name or address, depending on the test type, and for the latter, we selected the same name or address. For the close match cases, we used GPT-4o to help

[3]We focused on organization names and not individual names because the OFAC lists contain many phonetic transliterations of individuals' names. Evaluating phonetic similarities is beyond the scope of our study.

generate deviations from the original organization names and addresses. Specifically, we asked the model to propose subtle and plausible deviations from the original strings that meet the close match case typologies. Importantly, GPT-4o is not one of the LLMs underlying the pairwise comparisons in the main results. Using a different model for data generation helps ensure independence between our test data and evaluation systems.

To facilitate human-in-the-loop output review, we asked for the data in small batches of roughly 25 strings for one country at a time. Because of our detailed data evaluation process, we were only able to generate data for ten countries: Australia, Canada, France, Germany, Italy, Ireland, New Zealand, Spain, the United Kingdom, and the United States.[4] Our dataset includes 260 organization names and 429 street addresses. Each of the 689 strings has a variation for the four case types. Therefore, the classification results we present in section 4 are based on 2,756 observations for each model. Table 2 shows some representative data for the close matches.

Table 2: Representative Data for the Close Matches

Original	Positive close match	Negative close match
Havin Bank Limited	Havin Bank Ltd	Haven Banking Co.
NPC International	N.P.C. Int'l	NVC International
Mellat Insurance Company	Mellat Ins. Co.	Mellot Insurance Corp.
Via Lorenzo Rocci 14	Via L. Rocci, No. 14	Via Lorenzo Ricci 12
20 Rue Auguste Vacquerie	20 Rue A. Vacquerie	22 Rue Auguste Vauquelin
345 E. Railway Avenue	345 East Railway Ave.	345 W. Railway Street

3.2 Language Models

We used language models from four different model families available in the AWS Bedrock service to conduct the pairwise string comparisons. For each family, we tested one larger and one smaller variety. For ease of reference, we refer to the larger models as LLMs and the smaller models as SLMs. Table 3 summarizes the models that we tested.[5]

Table 3: Language Models Used in the Assessment

Family	Developer	LLM	SLM
Claude	Anthropic	Claude 4 Sonnet	Claude 3.5 Haiku
Llama	Meta	Llama 3.3 70B	Llama 4 Scout 17B
Mistral	Mistral AI	Mistral Large	Mistral Small
Nova	Amazon	Nova Pro	Nova Micro

Our core prompt represents a few-shot approach (Brown et al., 2020) that establishes the

[4] We selected the countries that we felt we would have the most success auditing based on language, alphabet, and personal experience.

[5] Although Claude Opus is bigger than Claude Sonnet, we used the latter due to speed and cost constraints.

model's persona, describes the comparison task, provides several examples of likely matches and non-matches, presents the two strings that need to be compared, and asks the model to produce a score on a scale of 0-100, where 100 represents the highest likelihood of a match. The fuzzy matching assessments are on a scale of 0-100 as well. We also tested a zero-shot prompt, which is identical to the few-shot prompt but withholds the examples, and a chain-of-thought reasoning prompt in the spirit of Wei et al. (2022), which presents the model with a step-by-step framework for thinking through example assessments before asking it to rate the pair. While the few-shot approach slightly edged out the other two on classification performance, accuracy differences were negligible. The Appendix presents the few-shot and chain-of-thought prompts. We used different templates for organization names and street addresses, and we set the temperature to zero for our model runs to limit response variation across iterations.[6]

3.3 Fuzzy Matching Methods

We compared the performance of the LLMs and SLMs to four widely used fuzzy matching methods: Levenshtein, Jaro-Winkler, token sort, and token set. Levenshtein, which is also known as edit distance, is based on the number of operations, including insertions, deletions, and substitutions, that it takes to transform one string into another (Levenshtein, 1965). Jaro-Winkler is derived from the number of matching characters between two strings, the number of transpositions, and the length of the strings, with greater weight assigned to commonalities at the beginning of the strings (Jaro, 1989; Winkler, 1990). Token sort and token set are two variations on edit distance introduced by SeatGeek (2011). The former measures the edit distance of two strings that have been transformed by tokenizing the strings and sorting them in alphabetical order. The latter works by tokenizing two strings, extracting and sorting the intersection, constructing new comparison strings using the sorted intersection and remainders, computing the edit distance among the transformed strings, and taking the highest score. In addition to these core methods, we assessed the weighted ratio implemented by Bachmann (2021), which dynamically selects candidate algorithms based on relative string length, applies small penalties to token-based methods and scaling factors to partial ratio methods, and returns the highest score.[7] For comparability, we used the normalized similarity version of these metrics on a 0-100 scale, where higher values indicate higher levels of similarity.

[6]The temperature is an inference parameter that controls the likelihood that the language model opts for higher- or lower-probability query responses. Setting the temperature to zero instructs the model to select the highest-probability response.

[7]The weighted ratio selects from normalized Indel similarity, partial ratio, token set, token sort, partial token set, and partial token sort.

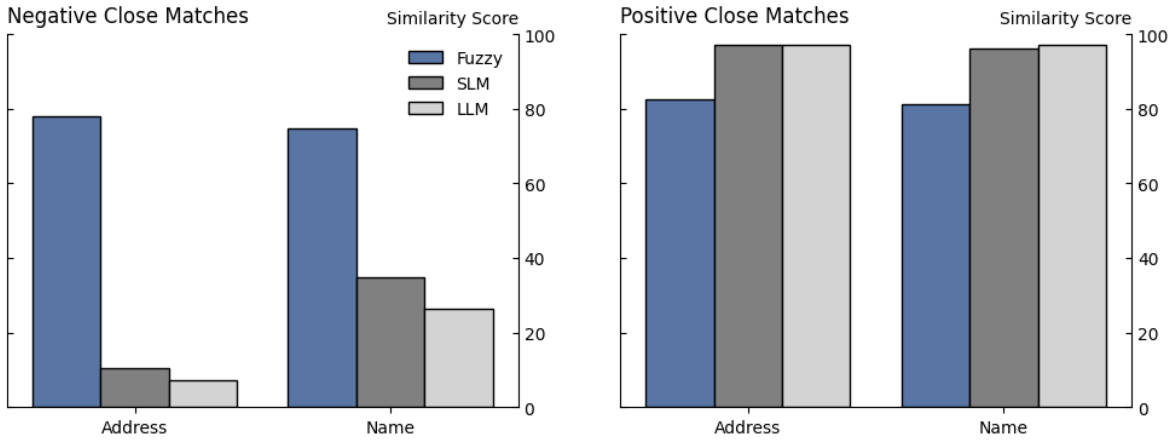

Figure 1. Average Similarity Scores for Close Matches. The bar plots depict the average similarity scores on a scale of 0-100 generated by the fuzzy methods (blue bars), SLMs (dark gray bars), and LLMs (light gray bars) for the negative close matches (left plot) and positive close matches (right plot), split out by addresses (left grouping) and names (right grouping). The fuzzy methods produce similar ratings across all case and test type combinations, while the language models are better able to distinguish between the negative and positive case types. No. of observations: addresses = 429, names = 260.

4 Screening Results

This section presents the results of our assessment along several dimensions. We begin by exploring the descriptive properties of the models, focusing on the average similarity scores and the distribution of the scores. We then present the classification performance of the models on the raw text. Next, we show how the language models perform relative to the fuzzy methods after putting the name and address strings through a rigorous pre-processing and normalization framework and relative to a more dynamic fuzzy matching method.

4.1 Descriptive Properties

Figure 1 presents the average similarity scores produced by the fuzzy matching methods, SLMs, and LLMs for the close matches, split out by addresses and names. Lower scores are better for the negative cases, and higher scores are better for the positive cases. Across the board, the language models outperform the fuzzy matching algorithms. The latter produce similar ratings on average for both sets of cases and test types, which is undesirable. Meanwhile, there is a clear divergence in the way the language models rate the negative and positive cases. For the negative close matches, the language models' scores are considerably lower than those of the fuzzy methods, but the gap is wider for addresses than names. For the positive close matches, there are less dramatic but still meaningful differences in the ratings. The LLMs slightly outperform the SLMs, but performance is comparable, especially for the positive cases.

Figure 2 captures the distributions of the scores for the three methods across the close match

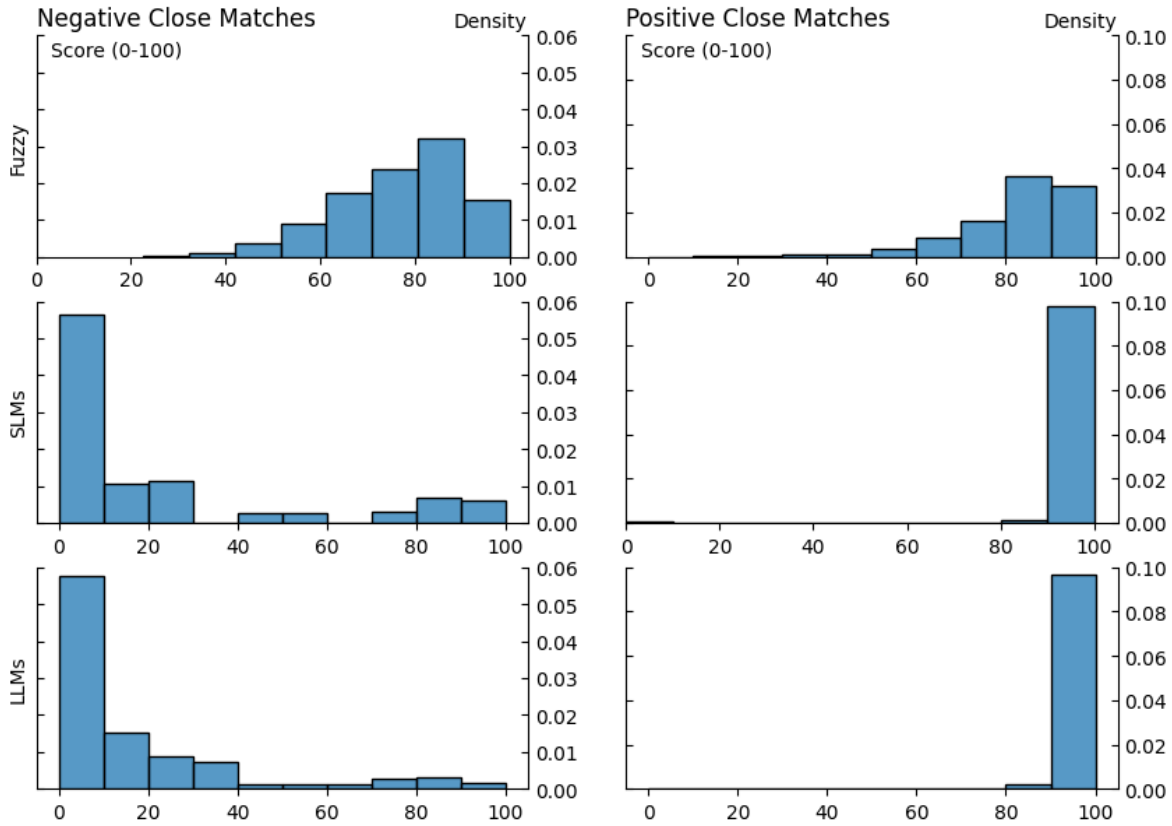

Figure 2. Distribution of Similarity Scores for Close Matches. The histograms depict the distribution of similarity scores on a scale of 0-100 generated by the fuzzy methods (top row), SLMs (middle row), and LLMs (bottom row) for the negative close matches (left column) and positive close matches (right column). Each histogram is based on 2,756 scores, representing 260 and 429 pairwise name and address comparisons, respectively, for four constituent models. The language models tend to be much more certain in their ratings than the fuzzy methods, with scores clustering near zero for the negative cases and 100 for the positive cases.

cases. The most important takeaway is that the language models show high levels of certainty in their ratings. Their score distributions are concentrated between 0-10 for the negative cases and 90-100 for the positive cases, while the fuzzy methods produce smoother distributions. The distributional differences between the case types are also consistent with the average scores depicted in figure 1. The SLMs and LLMs are better able to distinguish between the negative and positive cases, while the fuzzy methods tend to give high scores for both case types.

4.2 Classification Performance

Sanctions screening systems generally use similarity scores to classify records based on a matching threshold (Fadavi, 2023). Figure 3 depicts the average classification scores for the three model types across four realistic matching thresholds and for three related metrics: the F1 score, precision, and

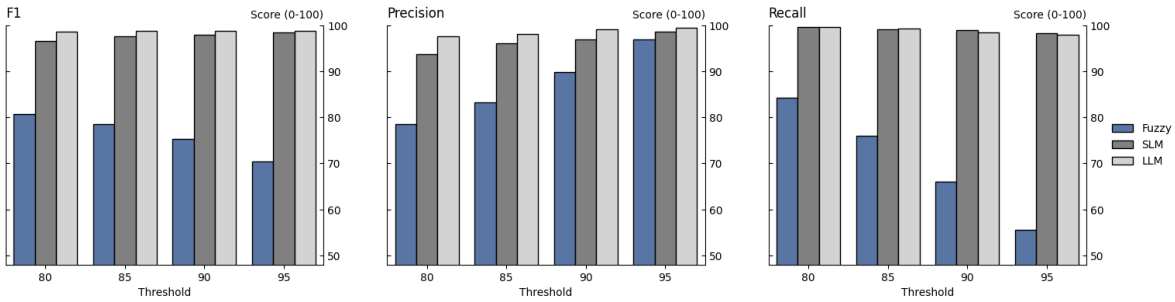

Figure 3. Classification Performance Across Realistic Matching Thresholds. The bar plots depict the average classification performance across four matching thresholds (80, 85, 90, 95), as measured by three metrics: F1 score (left), precision (middle), and recall (right). The SLMs (dark gray bars) and LLMs (light gray bars) decisively outperform the fuzzy methods (blue bars) based on the F1 summary metric. The next two plots show that while the fuzzy methods demonstrate the classic tradeoff between precision and recall as we raise the matching threshold, the language models' performance is stable. Average scores are based on 2,756 assessments performed by each underlying model.

recall.[8] In contrast to figures 1 and 2, which focus on the close match cases, the classification scores are computed on the full sample, including the clear positives and clear negatives (see: section 3.1). The F1 score (left), which is the harmonic mean of precision and recall, shows the language models decisively outperforming the fuzzy methods across all matching thresholds, while the performance of the SLMs and LLMs is comparable. Differences in performance among the individual LLMs and SLMs, which we do not depict here, were negligible.[9]

The F1 scores for the fuzzy methods deteriorate noticeably as the matching threshold increases. The reason for this is clear in observing the constituent elements of F1, precision and recall.[10] The fuzzy methods exhibit the classic tradeoff between the two. As we raise the threshold, fewer records are classified as matches. Consequently, precision increases but recall plummets. By contrast, the language models show stability with the changing threshold. This follows directly from the similarity score distributions depicted in figure 2. Because the language models tend to be very certain one way or the other–that is, their scores cluster around 0 and 100–performance is not as sensitive to adjustments in the matching threshold. In practice, this means that practitioners could likely select a very high matching threshold, without materially affecting detection rates.

[8]The classification results presented throughout this paper reflect in-sample performance. We did not train any models, so we are not as concerned about overfitting. The fuzzy methods are deterministic, and the language models are pre-trained with no fine tuning.

[9]The Mistral models, which are developed by a French AI firm, slightly outperformed the other models on assessments originating from non-English language speaking countries, but the differences were small.

[10]In this context, precision captures the share of records classified as matches that are truly matches. Recall captures the share of true matches that are correctly identified by the system.

4.3 Pre-processing, Normalization, and Dynamic Fuzzy Methods

Many deviation patterns in organization names and street addresses are recurring. For example, "Street" is often represented as "St." Therefore, we may be giving the LLMs too much credit for repeatedly recognizing the same patterns, when we could normalize recurring deviations to a common representation. To examine the relative performance of the LLMs and fuzzy methods after making such adjustments, we developed a rigorous textual pre-processing and normalization routine, which is presented in detail in the Appendix. The process involves converting all text to lowercase, removing punctuation, stripping extra whitespace, and normalizing recurring abbreviations and acronyms to common words. As examples, all instances of "ltd" were replaced with "limited," and all instances of "rd" were replaced with "road." We made 17 such adjustments for the organization names and 37 for the addresses.[11]

Pre-processing and normalization are challenging. Without implementing complex exception handling rules, erroneous normalization introduces errors. For example, it is not uncommon for streets to be named after saints, which, like the word street, typically use the abbreviation "St." (for example, St. Mary's Lane). There are also two reasons why pre-processing and normalization are likely to be more successful in our simulation than in a real-world sanctions screening system. First, our sample does not include individual names, which have fewer opportunities to normalize recurring patterns than organization names. Second, the primary languages of the countries in our sample (English, French, German, Italian, and Spanish), share the same core Roman alphabet. Normalization is less reliable for names and addresses that have been transliterated from languages like Russian, Arabic, Chinese, and Farsi–the most represented origin languages on the OFAC lists.

As we discuss further in section 5, the fuzzy algorithms are computationally efficient, so it is feasible to pursue methods that dynamically select from multiple potential fuzzy matching approaches. To this end, we also assessed the performance of the dynamic weighted ratio (Bachmann, 2021), which is discussed in 3.3.

Figure 4 provides a holistic summary of how the language models perform relative to the fuzzy matching methods under various scenarios. The figure captures the average percentage difference in false and true positives for the language models compared to the fuzzy methods for all matching thresholds between 80 and 95. The blue and light gray bars represent performance relative to the average of the fuzzy algorithms on the raw and normalized text, respectively. The dark gray bar captures performance relative to the weighted ratio, which is applied after pre-processing and normalization.

The LLMs and SLMs reduce false positives by 91 and 75 percent, respectively, on the raw text and improve detection rates by 40 percent. The false positive gains are robust to pre-processing and normalization, but the detection rate performance gap narrows significantly to about 17 percent.

[11] We tokenized the strings by word, and the replacements were carried out on standalone tokens. We did not replace sub-strings.

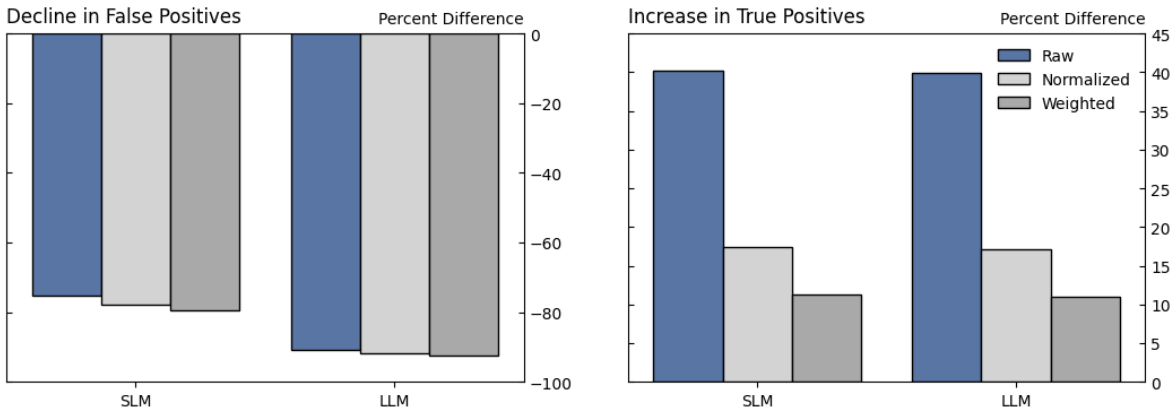

Figure 4. Average Language Model Performance Relative to the Fuzzy Methods (Matching Thresholds: 80-95). The bar plots capture the average percent decline in false positives (left plot) and percent increase in true positives (right plot) achieved by the SLMs (left grouping) and LLMs (right grouping) relative to the fuzzy methods over the matching thresholds 80-95 on the raw text (blue bars) and pre-processed and normalized text (light gray bars). The dark gray bars represent the language model performance compared to the weighted ratio, which also includes pre-processing and normalization. The false positive performance is robust across all methods, while the performance gap narrows for the true positives.

The weighted ratio with pre-processing and normalization performs even better from a detection perspective, but it does not change the false positive picture.

Overall, the weighted ratio represents the best performance balance among the fuzzy approaches. Compared to the weighted ratio, the LLMs and SLMs reduce false positives by 92 and 80 percent, respectively, and detect about 11 percent more matches. Thus, even with rigorous text pre-processing and dynamic fuzzy matching approaches, the LLMs and SLMs show significant reductions in false positives and meaningful gains in matching performance, reinforcing the notion that language models have the potential to simultaneously balance precision and detection.

5 Computing Performance and Model Cascading

To assess computing performance, we carried out a smaller scale test designed to avoid triggering API limits. The API request limits per minute ranged from 200-800 for the LLMs and 200-1600 for the SLMs in our study. Given the lower bound on both model types, we examined the runtime on 200 pairwise comparisons of names and addresses, for a total of 400 evaluations for each model.

Because the language models are considerably slower than the fuzzy algorithms, we also designed and assessed a model cascade system that only invokes language models for high uncertainty cases and defers to simpler methods for easier cases. Specifically, the cascading algorithm first tests for an exact string match and returns 100 if one exists. Next, it calculates the fuzzy weighted ratio. If the ratio is below a user-defined lower bound, it returns 0, and if the ratio is above a user-defined upper bound, it returns 100. Otherwise, it escalates to a language model. The intuition is that

above the upper bound and below the lower bound, there is a high probability that the strings are matches and non-matches, respectively. In these cases, it is more efficient from an economic and computing performance perspective to abstain from LLM inference.

The challenge, of course, is setting the lower and upper bounds. The lower bound is meant to quickly rule out true negatives while also minimizing false negatives. The upper bound is meant to quickly identify true positives, while minimizing false positives. For this assessment, we set the lower bound to 45 and the upper bound to 98. We chose the thresholds by examining the distribution of the weighted ratio against screening outcomes. False positives increase when the upper bound is set lower than 98, while false negatives increase when the lower bound is set below 45. Practitioners could similarly set the threshold boundaries by examining historical data and screening performance.

The model cascade only uses one SLM or LLM at a time as the final layer, and below we present the average performance for the cascade approach across all the models. However, we also assessed the performance gains for the cascade using only the fastest SLM and LLM in our study, which were Llama 4 Scout and Mistral Large, respectively. The Claude models were the slowest in our simulation, and they drive up the average runtime considerably. Table 4 presents the computing performance results for the language models under three different scenarios: the flat system, which represents the average runtime for the four language models when they evaluate every record, the average performance for the four language models using the cascade, and the cascade using Llama 4 Scout (SLM) and Mistral Large (LLM).

Table 4: Average Runtime (in seconds) for 400 Pairwise Evaluations

Measurement	Model Type	Flat (Average)	Cascade (Average)	Cascade (Fastest)
Total	LLM	179.8	99.27	49.42
	SLM	135.2	82.21	45.55
Per Record	LLM	0.450	0.248	0.124
	SLM	0.338	0.206	0.114

Under the flat system, the LLMs took about three minutes, on average, to carry out 400 pairwise comparisons, at a rate of just under half a second per record.[12] By comparison, it only took the fuzzy algorithms about seven milliseconds, on average, to carry out the evaluations. Accordingly, the LLMs were over four orders of magnitude slower than the fuzzy methods. The SLMs were about 25 percent faster than the LLMs but still much slower than the fuzzy algorithms. The cascade reduced the LLM runtime by 45 percent compared to the flat system. The fastest model cascade was another 50 percent faster than the overall cascade average. Performance gains were

[12]The median runtime was 40 percent less than the average runtime for both the SLMs and LLMs. The mean was driven up the Claude models, which were the slowest in the cohort we tested.

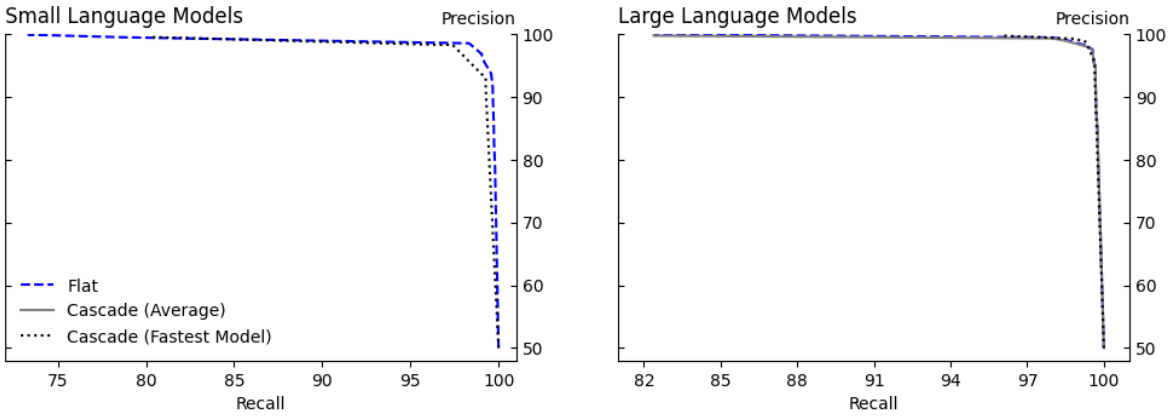

Figure 5. Screening Accuracy of the Model Cascade. The figure plots precision-recall curves for all matching thresholds between 0 and 100 for the flat system (blue dashed line), the average cascade (solid gray line), and cascade using the fastest model (black dotted line). Performance is very similar across the different systems. The areas under the curves are around 99 percent for each system.

comparable for the SLMs. For both model types, the runtime approached about a tenth of a second per evaluation under the fastest model system.

An important question is whether the cascade compromises on screening accuracy. To this end, figure 5 plots precision-recall curves for all matching thresholds between 0 and 100 for the three systems separately for the SLMs and LLMs. More area under the curve indicates higher performance. Accuracy among the three systems is virtually identical. The precise areas under the curve are all above 99 percent, with just slightly lower performance for the fastest SLM. The results suggest that practitioners could pursue these types of performance-optimized systems without significant deterioration in screening accuracy.

We did not test the computing performance effects of the buffering logic needed to accommodate API limits because these limits are likely to vary for many different types of cloud and AI service provider accounts. However, we used this type of logic to accommodate API limits in producing the main results, and it is clearly another source of friction. On the other hand, we also did not quantify the extent to which the language models' false positive gains can reduce manual interventions, but the time saved could be substantial. The need for sanctions screening speed will vary based on the financial activity in question. In slower moving processes, such as the customer onboarding and due diligence required for account opening, lending, and insurance underwriting, it may be feasible to increase the scope of LLM usage throughout the screening process, including potentially incorporating reasoning models. In other settings, such as payment processing, screening needs to occur near real-time. In these higher velocity systems, inference latency and other frictions, such as API limits, will compel the use of tiered approaches like the model cascade we propose here.

6 Conclusion

Inaccurate sanctions screening can lead to costly errors for financial institutions and payment processors, including potential enforcement actions and increased compliance burden. Weak sanctions screening systems also harm national security, as sanctioned entities can move money more easily. We show how LLMs and SLMs can help drive significant sanctions screening accuracy gains over more traditional fuzzy logic, especially in reducing false positives. The false positive gains were robust to various fuzzy matching enhancements, including string pre-processing and normalization and using dynamic approaches. SLMs performed comparably to LLMs in our study, which may support scaling, as they tend to be cheaper, have higher API limits, and are a bit faster. Some firms may also be able to deploy open-weight SLMs in sanctions screening applications using their own hardware.

While the language models in our study were more accurate than the fuzzy methods, they were also considerably slower. We did not attempt to quantify the time savings that could be achieved by reducing false positives, which may be substantial in practice. However, we did assess a model cascading approach that defers to simple matching methods for straightforward cases, while escalating more uncertain cases to the language models. The cascade dramatically reduced runtime with no loss of accuracy. These types of tiered approaches will be needed for LLM-based sanctions screening in settings that require near real-time screening, such as payment systems. Other types of financial activities, such as customer due diligence for account opening, credit review, and insurance underwriting, may be able to rely on LLMs more extensively.

Our findings and scope of research point to several potential areas for future inquiry. As a starting point, follow-up studies could scale up simulations beyond pairwise comparisons to screen against full sanctions lists. Researchers could also explore more efficient ways of giving LLMs access to sanctions list data, such as through fine tuning or retrieval augmented generation. Additionally, our model cascade only uses one type of language model as the final evaluation layer. Future work could evaluate sanctions screening decision rules for deferring to SLMs before escalating to LLMs and potentially incorporate deep reasoning models for the most complex cases. Finally, future research could explore LLM applications for similar types of verification and authentication challenges in the financial system.

References

Aldasoro, I. and A. Desai (2025). Ai agents for cash management in payment systems. *Available at SSRN*.

Aldasoro, I., L. Gambacorta, A. Korinek, V. Shreeti, and M. Stein (2024). Intelligent financial system: how ai is transforming finance. *BIS Working Papers* (1194).

Allen, J. (2025). Cardsim: a bayesian simulator for payment card fraud detection research. *Finance and Economics Discussion Series* (2025-017).

Altman, E., J. Blanuša, L. Von Niederhäusern, B. Egressy, A. Anghel, and K. Atasu (2023). Realistic synthetic financial transactions for anti-money laundering models. *Advances in Neural Information Processing Systems 36*, 29851–29874.

Araujo, D., S. Doerr, L. Gambacorta, and B. Tissot (2024). Artificial intelligence in central banking. *BIS Bulletin* (84).

Bachmann, M. (2021). Rapidfuzz documentation: Wratio. Online; accessed 25-June-2025.

Bolukbasi, T., J. Wang, O. Dekel, and V. Saligrama (2017). Adaptive neural networks for efficient inference. In *International conference on machine learning*, pp. 527–536. PMLR.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems 33*, 1877–1901.

Chen, L., M. Zaharia, and J. Zou (2023). Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Desai, A., A. Kosse, and J. Sharples (2025). Finding a needle in a haystack: a machine learning framework for anomaly detection in payment systems. *The Journal of Finance and Data Science 11*, 100163.

Dunn, W., E. E. Meade, N. R. Sinha, and R. Kabir (2024). Using generative ai models to understand fomc monetary policy discussions. *FEDS Notes*.

Eisfeldt, A. L. and G. Schubert (2024). Generative ai and finance. *Annual Review of Financial Economics 17*.

El-Yaniv, R. et al. (2010). On the foundations of noise-free selective classification. *Journal of Machine Learning Research 11*(5).

Fadavi, A. (2023). Economic sanctions on the rise: The ever-increasing importance of sanctions screening in a compliance programme. *Journal of Financial Compliance 6*(4), 333–345.

Fischer, E., R. McCaughrin, S. Prazad, and M. Vandergon (2023). Fed transparency and policy expectation errors: A text analysis approach. *Federal Reserve Bank of New York Staff Reports* (1081).

Hansen, A. L. and S. Kazinnik (2024). Can chatgpt decipher fedspeak? *Available at SSRN 4399406*.

Hilal, W., S. A. Gadsden, and J. Yawney (2022). Financial fraud: a review of anomaly detection techniques and recent advances. *Expert systems With applications 193*, 116429.

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical association 84*(406), 414–420.

Kazinnik, S. (2023). Bank run, interrupted: Modeling deposit withdrawals with generative ai. *Available at SSRN*.

Kazinnik, S. and E. Brynjolfsson (2025). Ai and the fed. *NBER Working Paper Series* (33998).

Kim, S. and S. Yang (2024). Accuracy improvement in financial sanction screening: is natural language processing the solution? *Frontiers in Artificial Intelligence 7*, 1374323.

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. In *Doklady Akademii Nauk*, Volume 163, pp. 845–848. Russian Academy of Sciences.

Nie, L., Z. Ding, E. Hu, C. Jermaine, and S. Chaudhuri (2024). Online cascade learning for efficient inference over streams. *arXiv preprint arXiv:2402.04513*.

Nino, R., A. Sison, and R. Medina (2019). Optimization of edit distance algorithm for sanctions screening risk score assessment. *International Journal of Advanced Trends in Computer Science and Engineering 8*(6), 1289–1295.

OFAC (2021). Frequently asked questions: No. 249. Online; accessed 25-June-2025.

OFAC (2025). Sanctions list service: Specially designated nationals list. Online; accessed 31-March-2025.

SeatGeek (2011). Fuzzywuzzy: fuzzy string matching in python. Blog post. Online; accessed 25-June-2025.

Silva, T. C., K. Moriya, and R. M. Veyrune (2025). From text to quantified insights: a large-scale llm analysis of central bank communication. *IMF Working Papers 2025*(109), A001.

Teerapittayanon, S., B. McDanel, and H.-T. Kung (2016). Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pp. 2464–2469. IEEE.

Varshney, N. and C. Baral (2022). Model cascading: Towards jointly improving efficiency and accuracy of nlp systems. *arXiv preprint arXiv:2210.05528*.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. *Advances in neural information processing systems 30*.

Viola, P. and M. Jones (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, Volume 1, pp. I–I. Ieee.

Wei, J., X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems 35*, 24824–24837.

West, J. and M. Bhattacharya (2016). Intelligent financial fraud detection: a comprehensive review. *Computers & security 57*, 47–66.

Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *ERIC*.

Zhang, B., H. Yang, and X.-Y. Liu (2023). Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models. *arXiv preprint arXiv:2306.12659*.

Appendix: Prompts and Text Pre-processing

This appendix includes additional details on our prompts (see: section 3.2) and pre-processing and normalization routine (see: section 4.3).

Prompts

We tested zero-shot, few-shot, and chain-of-thought (CoT) prompts. We have different templates for the addresses and organization names. While the few-shot prompts edged out the other prompts on screening performance, the differences were negligible. The results from the few-shot prompts are the main ones we present in the paper. The few-shot prompts include four components, which are presented below. We concatenate these together into a single query for each pairwise comparison. The zero-shot prompts are identical to the few-shot prompts, but they withhold the examples. The CoT prompts retain the same basic structure as the few-shot prompts, but we replace the list of examples with step-by-step reasoning templates. We include four reasoning examples in each prompt, two for matches and two for non-matches. We present the CoT reasoning templates below, with two reasoning examples for brevity.

Few-Shot Prompt: Addresses

Persona

You are an expert evaluator specializing in judging whether two street address strings are a match.

General Task

Your task is to rate your confidence that two strings represent the same street address after accounting for common variations (for example: abbreviations, ordering, punctuation, formatting, etc.).

Few-Shot Examples

Here are a few examples of address pairs that are likely to be matches:

- Address 1: 63 South Capitol Lane; Address 2: 63 S. Capitol Ln.
- Address 1: 8 Marshall Road, Unit 14; Address 2: 8 Marshall Rd., No. 14
- Address 1: Karlstrasse 35; Address 2: Karlstr. #35

Here are a few examples of address pairs that are NOT likely to be matches:

- Address 1: 1001 K ST NW; Address 2: 1000 K ST NE
- Address 1: 50 Anson Rd.; Address 2: 50 Anston St.
- Address 1: 10 Rue de la Paix; Address 2: 12 Rue de la Paie

Specific Task

Now, carefully evaluate addresses 1 and 2 below:

- Address 1: {`String 1`}
- Address 2: {`String 2`}

Provide your confidence rating on a numeric scale from 0 to 100, where:

- 100 means you are completely certain addresses 1 and 2 represent the same street address.
- 0 means you are completely certain addresses 1 and 2 are different street addresses.

Respond ONLY with a single integer between 0 and 100. Do NOT provide any additional explanations or words.

Few-Shot Prompt: Organization Names

Persona

You are an expert evaluator specializing in judging whether two name strings for companies and other organizations are a match.

General Task

Your task is to rate your confidence that two strings represent the same underlying company or organization name after accounting for common variations (for example: acronyms, abbreviations, ordering, punctuation, formatting, etc.).

Few-Shot Examples

Here are a few examples of name pairs that are likely to be matches:

- Name 1: International Advisors LTD; Name 2: Int'l Advisors Limited
- Name 1: Inst. for Energy Research and Development; Name 2: Institute for Energy R&D
- Name 1: National Relief Organization; Name 2: Nat'l Relief Org.

Here are a few examples of name pairs that are NOT likely to be matches:

- Name 1: International Advisors LTD; Name 2: Global Advisors Corp.
- Name 1: Inst. for Energy Research and Development; Name 2: Association of Energy Researchers
- Name 1: National Relief Organization; Name 2: Medical Relief Organization

Specific Task

Now, carefully evaluate names 1 and 2 below:

- Name 1: {`String 1`}
- Name 2: {`String 2`}

Provide your confidence rating on a numeric scale from 0 to 100, where:

- 100 means you are completely certain names 1 and 2 represent the same underlying name.
- 0 means you are completely certain names 1 and 2 are different underlying names.

Respond ONLY with a single integer between 0 and 100. Do NOT provide any additional explanations or words.

Chain-of-Thought Reasoning Templates

CoT Reasoning: Addresses

Below are examples showing the step-by-step reasoning process.

Example 1 (MATCH):
Address 1: 63 South Capitol Lane
Address 2: 63 S. Capitol Ln.

Step 1: Identify differences - South vs S., Lane vs Ln.
Step 2: Consider variation types - Abbreviations
Step 3: Evaluate reasonableness - Common address abbreviations
Step 4: Assess intersection - 63, Capitol : Street number and core street name combination is strong similarity signal
Step 5: Certainty level - High certainty these are the same street addresses

Example 2 (MATCH)...

Example 3 (NOT A MATCH):
Address 1: 1001 K ST NW
Address 2: 1000 K ST NE

Step 1: Identify differences - 1001 vs 1000, NW vs. NE
Step 2: Consider variation types - Street number, directional
Step 3: Evaluate reasonableness - Numbers indicate opposite street sides, Directionals indicate different city locations
Step 4: Assess intersection - K, ST: potentially same street, but variations supersede
Step 5: Certainty level - Low certainty these are the same entity

Example 4 (NOT A MATCH)...

CoT Reasoning: Organization Names

Below are examples showing the step-by-step reasoning process.

Example 1 (MATCH):
Name 1: International Advisors LTD
Name 2: Int'l Advisors Limited

Step 1: Identify differences - International vs Int'l, LTD vs Limited
Step 2: Consider variation types - Abbreviations
Step 3: Evaluate reasonableness - Standard word and organizational structure abbreviations
Step 4: Assess intersection - Advisors: generic domain overlap
Step 5: Certainty level - High certainty these are the same underlying name

Example 2 (MATCH)...

Example 3 (NOT A MATCH):
Name 1: International Advisors LTD
Name 2: Global Advisors Corp.

Step 1: Identify differences - International vs Global, LTD vs Corp.
Step 2: Consider variation types - Company name, business structure
Step 3: Evaluate reasonableness - Similar but distinct name, different organizational structure
Step 4: Assess intersection - Advisors: generic domain overlap
Step 5: Certainty level - Low certainty these are the same entity

Example 4 (NOT A MATCH)...

Pre-processing and Normalization Steps

The pre-processing and normalization routine that we use in our analysis proceeds as follows:

1. Convert to lower case
2. Remove punctuation
3. Tokenize by word
4. Normalize recurring abbreviations and acronyms to common representations
5. Re-join tokens separated by a space
6. Strip extra whitespace

On step 4, we make 17 adjustments for the organization names and 37 for the addresses. The table below captures the name adjustments.

Target	Replacement
ltd	limited
inc	incorporated
corp	corporation
co	company
llc	limited liability company
org	organization
lp	limited partnership
llp	limited liability partnership
intl	international
sas	societe par actions simplifiee
immo	immobilien
imm	immobilien
sa	societe anonyme
srl	limited liability company
pty	proprietary
plc	public limited company
dac	designated activity company

The table below captures the address adjustments.

Target	Replacement
st	street
ave	avenue
avd	avenida
avda	avenida
bd	boulevard
boul	boulevard
blvd	boulevard
rd	road
dr	drive
ln	lane
ct	court
pl	place
wy	way
cir	circle
sq	square
ter	terrace
tce	terrace
cls	close
ch	chateau
po	post office
apt	apartment
ste	suite
bldg	building
fl	floor
no	number
n	north
nth	north
s	south
sth	south
e	east
w	west
ne	northeast
nw	northwest
se	southeast
sw	southwest
str	strasse
c	calle