Using Large Language Models to Measure U.S. Retirement Plan Design at Scale: Methods and Evidence*

Taha Choukhmane John Dedyo Cormac O'Dea
Lawrence Schmidt

February 2026

Abstract

We show how large language models can transform labor-intensive data collection in economics by constructing a comprehensive dataset of U.S. retirement plan characteristics from regulatory filings. Using a hand-collected dataset of 6,200 plans as training data, we fine-tune LLMs to automatically extract matching formulas, vesting schedules, and auto-enrollment provisions from unstructured text in regulatory filings. Our approach combines snippet extraction, retrieval-augmented generation, and double-coding validation to ensure data quality. Using this LLM-based pipeline we produce a dataset covering nearly 150,000 distinct plans over 21 years (2003-2024), yielding over one million plan-year observations: more than twentyfold the coverage of the original hand-collected data. This expansion enables the first comprehensive population-level analysis of retirement plan design. We document substantial increases in plan generosity, the increasingly powerful role of federal safe-harbor regulations in shaping firm plan offerings, and the dramatic rise of automatic enrollment from near zero to 40% of plans, illustrating how LLMs can make previously infeasible empirical research practical at scale.

1 Introduction

Defined contribution retirement plans such as 401(k) accounts are a cornerstone of the U.S. retirement system. Nearly 100 million Americans participate in employer-sponsored plans

*Choukhmane: MIT Sloan School of Management, tahac@mit.edu; Dedyo: Yale University, johnny.dedyo@yale.edu, O'Dea: Yale University, cormac.odea@yale.edu; Schmidt: MIT Sloan School of Management, ldws@mit.edu. We are very grateful to the Yale University Tobin Center for Economic Policy for co-funding this work. The authors have no relevant financial relationships or other potential conflicts to disclose.

that provide tax-advantaged saving, and the vast majority of these employers offer matching contributions that subsidize employee savings. Yet despite the central role of plan design features in shaping retirement outcomes, comprehensive data on these features across the universe of 401(k) plans remains scarce. Existing data sources are limited in two key ways: administrative datasets from plan sponsors provide rich information about participant behavior but only on their set of clients (Vanguard (2023), T. Rowe Price (2025)), while hand-collected datasets that sample from regulatory filings cover only a non-representative subset of plans due to the prohibitive cost of manual data collection (Arnoud et al. (2021)). These data limitations mean we lack a complete picture of the institutional environment shaping retirement savings choices for most American workers, including the match formulas, vesting schedules, and auto-enrollment provisions that determine how plan design translates into retirement wealth accumulation across the full distribution of firms and workers.

Our dataset construction exploits a key institutional feature of U.S. retirement regulation: all plan sponsors must file an annual Form 5500 with the Department of Labor. Plans exceeding 100 participants must include a Summary Plan Description attachment containing detailed narrative information about plan features such as matching formulas, vesting schedules, and automatic enrollment provisions. While these documents are publicly available, their format (typically three to five pages of relevant unstructured text in a one hundred page filing) has historically made large-scale data extraction prohibitively labor-intensive. The hand-collected dataset used by Choukhmane et al. (2025) required 15 undergraduate research assistants and an outsourced coding team to process approximately 6,200 plans spanning roughly 70,000 plan-years.

This paper leverages recent advances in large language models to dramatically expand this scope. We use the hand-collected dataset as training data to fine-tune LLMs that can automatically extract and tabulate plan features from Form 5500 filings. Our approach combines snippet extraction, retrieval-augmented generation, and double-coding validation to produce high-quality structured data. The resulting dataset, compiled with the help of a single research assistant (now one of the authors), covers nearly 150,000 distinct retirement plans over 21 years (2003-2024), yielding around 1.1m plan-year observations. This more than twentyfold expansion in coverage makes comprehensive population-level analysis of retirement plan design feasible for the first time.

We use this expanded dataset to document new facts about the retirement plan landscape and how it has evolved over two decades. First, we show that matching formulas are highly concentrated: just three matching structures account for over one-quarter of all plans offering employer contributions, with the Basic Safe Harbor formula (100% match up to 3% of salary, then 50% match from 3% to 5%) being the most prevalent. The share of plans meeting or

exceeding safe harbor requirements grew from approximately 10% in 2004 to nearly 55% by 2022, suggesting that federal regulations meaningfully shape plan design choices. Second, we document substantial growth in plan generosity over time, with matching caps, match rates, and maximum employer contributions all trending upward between 2004 and 2023, though all three measures showed temporary declines during the Great Recession. Third, we document one of the most substantial shifts in retirement plan design over the past two decades: automatic enrollment grew from essentially nonexistent in 2003 to nearly 40% of plans by 2023, with automatic escalation following a similar but slightly delayed trajectory. Finally, vesting schedules have become more generous, with the share of plans fully vesting within three years increasing from 40% to 65% over the sample period.

Our paper proceeds as follows. Section 2 outlines the method and describes the resulting data set. Section 3 summarizes how each of matching, vesting, and auto-enrollment in firm retirement plan characteristics evolved between 2003 and 2023.

2 Methods

This section summarizes our procedure for producing our data. Section 2.1 introduces the hand-collected data. Section 2.2 describes our procedure for gathering retirement plan description text from PDF documents. Section 2.3 describes the process by which large language models generate the structured data from this text. Figure 1 offers a pictorial overview of our data acquisition process to be described in detail in the remainder of this section.

2.1 Hand-Collected Data

The hand-collected data used for training is that collected for Choukhmane et al. (2025).[1] That paper constructed a novel dataset of employer 401(k) match formulas by having research assistants systematically code plan characteristics from Form 5500 filings. The RAs extracted the specific parameters describing employer matching schedules, vesting schedules, and auto-features: details that are contained in Form 5500 regulatory filings (those same filings we describe below). That dataset contains information for the largest approximately 5,000 plans in the US, as well as approximately 1,500 smaller plans.

One part of the data collection is worth noting as it will be relevant below. RAs were instructed to make an initial determination as to whether a formula was 'simple' or 'more complicated'. Simple plans were expressed in percentage of salary (we give an example below

[1]See Arnoud et al. (2021) which summarizes results from an early version of the data, and Choukhmane et al. (2024) who use it

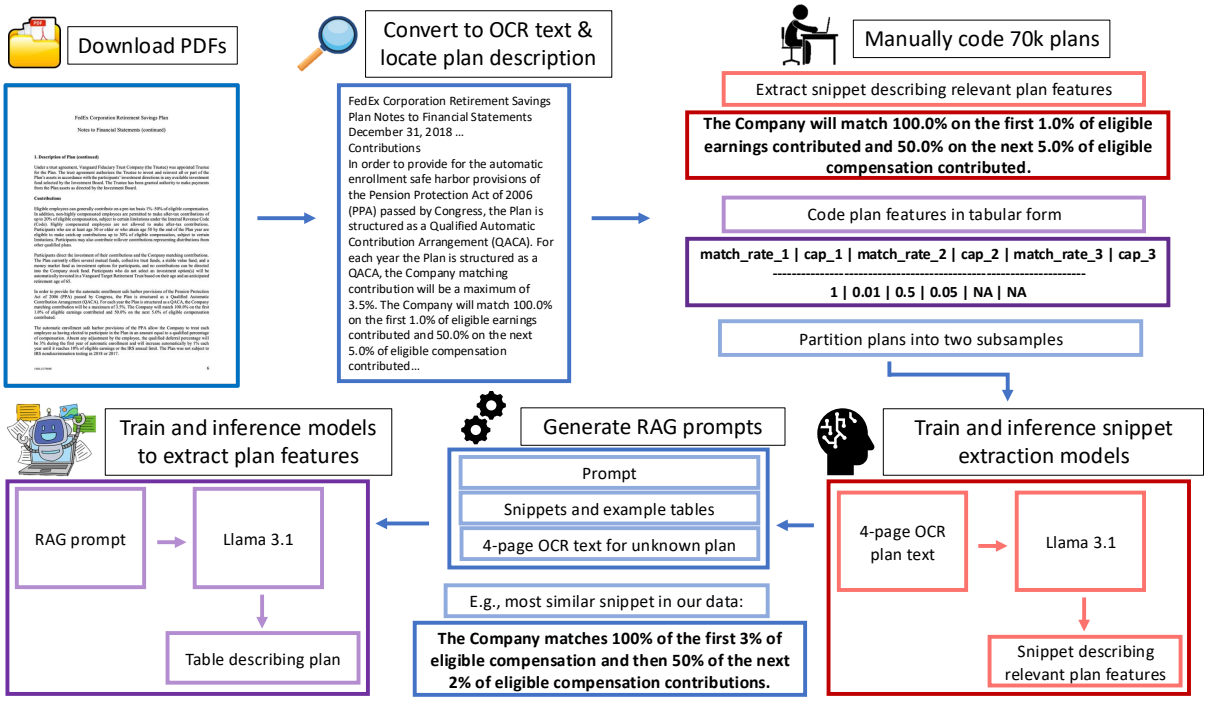


Figure 1: A flowchart of our data acquisition process.

in Figure 3). Common reasons for a plan to be classified as 'more complicated' include that it differed for different workers under the plan, it changed mid-year, it described the plan feature in a non-standard way (e.g., dollar- instead of percentage-based matching contributions), or it had qualification requirements beyond those imposed by the U.S. government. Approximately two-thirds of plans were adjudicated to be 'simple' (and so amenable to codification). The remaining third were 'more complicated', and we did not codify them. Our LLMs will also be trained to make this distinction.

2.2 Text Extraction

Each observation in our ultimate data set is derived from a Form 5500 filing: a regulatory document submitted each year that provides detailed information about an employee benefit plan. Form 5500 filings are publicly available through the Department of Labor. The first step of our process is to extract the plan text from these filings. The pipeline for doing so is as follows:

1. Download metadata on Form 5500 files and using it, identify Defined Contribution plans.

2. In parallel across thousands of CPUs, perform the following for each row in the sample:

(a) Download PDF.

(b) Determine where matching, auto-enrollment, and vesting are discussed, and save a shortened PDF.

(c) Use optical character recognition (OCR) to extract plaintext from the shortened PDF.

We describe these two steps below.

2.2.1 Defining The Sample

The Department of Labor provides Form 5500 filings as raw PDF documents, accompanied by metadata files with some characteristics of the plan (including plan type, number of participants, and gross accounting flows). Our population of interest is all plans which are either 401(k) or 403(b). There are 1,368,950 unique plans, covering 10,911,248 plan-years in total, of which there are 159,237 plans and 1,291,057 plan-years with more than 100 participants.

2.2.2 The PDF Pipeline

Most Form 5500 filings are around one hundred pages long. Before attempting to use OCR, we identify the section containing the plan details. Our program reads the original PDF file page by page until it finds the description of the plan, as determined by the inclusion of key phrases.[2] It then halts and creates a new PDF with the current page and the subsequent 3 pages. This reduces processing time and file sizes while retaining the relevant information for downstream analysis. The result of the pipeline is a large dataset of plaintext for each plan-year identifier, which is a format suitable for natural language processing. These plaintext plan descriptions are supplied to the LLMs in the process described in Section 2.3.

Figure 2 shows how the sample changes across each step in the text extraction process. We began with data from 1,368,950 firm retirement plans.[3] During PDF processing, some files were removed out for computational errors including file corruption, file length, or OCR text extraction failure. However the vast majority of filtered-out PDFs were rejected because they made no mention of retirement plan details. Figure 2 tracks the sample size in each year for plans which have 100 or more participants. Large plans are far more likely to remain in the sample since most smaller plans do not have a requirement to file a description of the

[2]These key phrases are found using a case-insensitive regular expression. The phrases are: "Description of Plan", "Description of the Plan", "Plan Description", "Summary of Significant Accounting Policies", and "Description of the 401(k) Pension Plan".

[3]This is distinct from the number of observations.

plan (though some choose to do so). The PDF pipeline produces the final sample size of 1,149,649 observations comprising 146,704 distinct retirement plans.

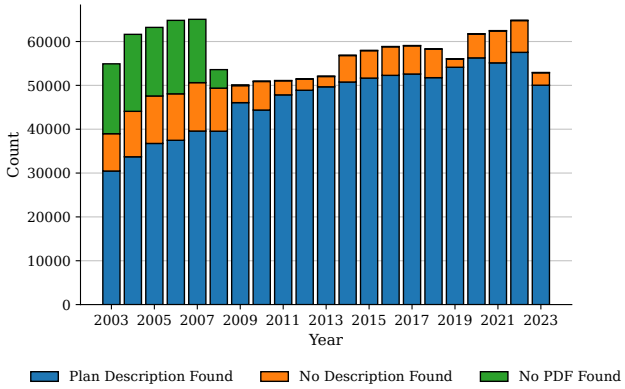


Figure 2: Sample size in each year for large plans.

Notes: This figure presents our sample size in the PDF pipeline for each year. For plotting, we restrict our sample to plans with 100 or more participants. Each bar has total height equal to the number of large plans in our universe for each year. 'Plan Description Found' is the number of plan-years for which we were able to find a description of the plan using a keyword search, described in Section 2.2.2 (this keyword search was successful for 87.2% of large plans). 'No Description Found' is the number of plan-years for which we were able to extract plaintext from the plan PDF, but failed to find a description of the plan. 'No PDF Found' is the number of plan-years for which we were unable to extract the text. The vast majority of these are due to the plan being missing from the Department of Labor index files (and therefore we could not download a PDF); however, this category also contains plan-years for which there were errors in text extraction. Such errors occur for only 0.05% of plan-years and would be invisible in the plot.

2.3 LLM Pipeline

Once we have obtained extracted text descriptions, we can apply our LLM pipeline that uses the text to create a table detailing the retirement plan features for each of nearly 150,000 U.S. retirement plans from 2003–2024. For each plan feature, the pipeline proceeds as follows:

1. Partition the hand-collected dataset to enable independent double-coding of the OCR text universe.

2. Fine-tune 'snippet extractor' LLMs using the hand-collected training dataset, and inference them on the hand-collected dataset.

3. Create retrieval-augmented generation (RAG) prompts from the hand-collected dataset and newly generated snippets. Use them to train 'tabulator' LLMs that will classify the plans as 'simple' or 'more complicated' (a distinction we described in Section 2.1).

4. Perform LLM inference on the universe of OCR text as follows:

(a) Inference the snippet extractor LLMs on the universe of all OCR text gathered as described in Section 2.2.

(b) Create RAG prompts for the OCR text universe, where the RAG examples come from the hand-collected dataset.

(c) Inference the tabulator LLMs on those prompts.

5. Process the LLM output to produce the final dataset.

The sections that follow give an in-depth description of each stage.

The pipeline is run separately for matching, auto-features, and vesting, which increases accuracy by allowing the LLM to specialize in exactly one of the three plan features. The task of examining all features at once is far more complex than examining each in isolation, and that complexity would add unnecessary noise to the data. In the following Sections 2.3.2 to 2.3.4, we explain each step of the process in terms of employer matching contributions. Sections B.0.1 and B.0.2 provide details and analytics for the collection of auto-features and vesting data.

	Value
`match_rate_1`	1.00
`cap_1`	0.03
`match_rate_2`	0.50
`cap_2`	0.05
`match_rate_3`	*NA*
`cap_3`	*NA*

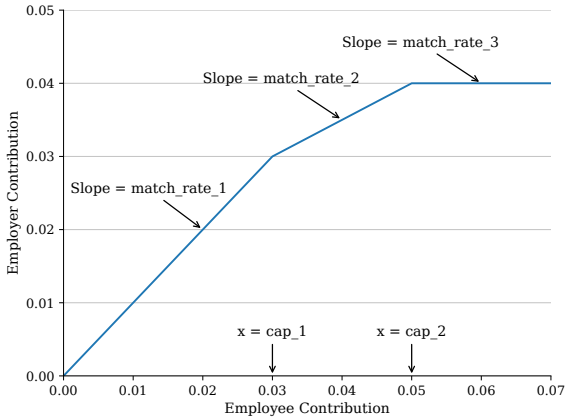


Figure 3: Output of the LLM for a *simple* retirement plan (left: tabular; right: graphical).

2.3.1 Subsampling for Purposes of Double-Coding Data

For data entry tasks involving generating structured outputs, it is often advantageous to have multiple human evaluators perform the same tasks independently. By restricting attention to cases in which hand-entered inputs agree, we can be more confident in the quality and accuracy of the data. When building fine-tuned models, we mimic this process by training two different models on independent subsets of the data. Incorporating this double-coding/partitioning process enables us to condition on whether the LLMs produce identical

output as a control for errors due to the stochasticity inherent in large language model outputs. In future iterations, we plan to use larger models to help adjudicate disagreements, but our current analysis restricts attention to the subset of cases in which these double-coded outputs agree.

The hand-collected dataset has 70,977 observations, and for training we partition it into halves at the plan level (i.e., if a plan is in a given partition, that partition contains all the data we have for that plan for all years, none of which appears in the other partition). Given that formulas and plan language are often identical across years, we partition at the plan-level rather than the plan-year level in order to better ensure independence of training examples and measurement errors across the two subsamples. This results in two roughly equal-sized train datasets that each contain completely different retirement plans. For ease of discussion later, we label the partitions $\mathbb{1}$ and $\mathbb{2}$.

2.3.2 Snippet Extraction

Our hand-collected data include text snippets summarizing various plan features which were extracted by our human RA team. To facilitate our downstream retrieval-augmented generation process, we use these hand-collected snippets to fine-tune LLMs to generate structured outputs identifying the relevant subsets of text. This allows us to generate custom, highly relevant examples of the desired structured outputs to use in our prompts for the final step of our process.

We begin by fine-tuning an open source industry LLM, Meta's Llama 3.1 8B Instruct (Grattafiori et al. (2024)), on prompts generated using our hand-collected data. Using the Huggingface Trainer API (Wolf et al. (2020)), we fine-tune two different copies of Llama 3.1, one on each partition of the train set. We refer to these two LLMs as the 'snippet extractors,' as they are trained to read the OCR-extracted plan document text and extract the section describing the match schedule. Each is trained on the RA-collected text snippets in its respective partition. Training lasts for three epochs (meaning that the LLM sees each train example three times—approximately 107,000 examples), uses the `adamw_torch` optimizer (an optimization algorithm introduced by Loshchilov and Hutter (2019)), and alters around 1% of the model weights. The snippet extractor trained on partition $\mathbb{1}$ (resp. $\mathbb{2}$) is denoted by $g_1(\cdot)$ (resp. $g_2(\cdot)$).

Prompt 1: Snippet Extraction for Matching

###INSTRUCTIONS###
You are a legal expert analyzing a snippet of text extracted from a firm's Form 5500 filings for the year [YYYY], which relates to their retirement plan.
Extract verbatim the sections from the text that describe employer contribution rules and any eligibility requirements. Include all text that describes employer matching contributions, along with the exact contribution rates and percentages of income. These often relate to matching contributions (including keywords like matching, non-discretionary, discretionary, or profit sharing). Copy the text describing employer contribution rules exactly as it appears in the plan language. It is very important that you include all text describing employer matching contributions (e.g., 100% match up to 5% of salary) to the plan in [YYYY]. Also include any requirements to be eligible for these contributions. Do not summarize. Do not make a bulleted list. Do not add any additional notes.
If you do not see any text describing employer matching contributions, your answer should be exactly "No mention of employer matching contributions.".
###CHECKLIST###
Before finalizing your answer, please check it against this list of "do"'s and "don't"'s:
- DO include too much text rather than too little text.
- DO copy the entire sentences and/or paragraphs verbatim without attempting to summarize. (The only exception here is that you can address OCR-related issues in the text to strip out extraneous characters, etc).
- DO include information about matching, discretionary, profit sharing, and non-discretionary employer contributions.
- DO include information about any dollar caps and/or information about eligibility criteria and/or rules which differ across different classes of workers.
- DON'T forget to copy a table summarizing the match rules, if the match is described in that format.
- DON'T try to summarize anything. Copy any text that is relevant for describing how employer contributions are calculated and how much the employer contributed in [YYYY].
- DON'T add additional notes/annotations which aren't direct quotes from the document. Your responses should only include the plan language.
When in doubt, please err on the side of including too much, not too little, text in your snippets.
###PLAN TO ANALYZE###
[PLAN]

Snippet extraction is necessary to generate RAG prompts, described in detail in Section 2.3.3. In fine-tuning and inference, the snippet extractor is provided with the full four-page OCR-extracted text from the company plan document and prompted to retrieve any language describing company match schedules and eligibility as faithfully as possible. We pass Prompt 1 to the snippet extractor trained to specialize in matching conributions. As many documents describe the plan features in a number of years, we replace [YYYY] in Prompt 1 with the year corresponding to the plan-year in which we are interested.

After three epochs of fine-tuning, the LLM-extracted snippets in their respective out-of-sample test sets[4] are remarkably similar to the RA-extracted snippets (which were copy-pasted by hand from company PDFs); see Figure 4. From reviewing many examples from this process, we found that training for more epochs tended to yield less favorable results out of sample, as the LLMs would often 'memorize' potentially erroneous information from the hand-collected data.

[4]That is, the partition of the hand-collected data on which the model was not trained.

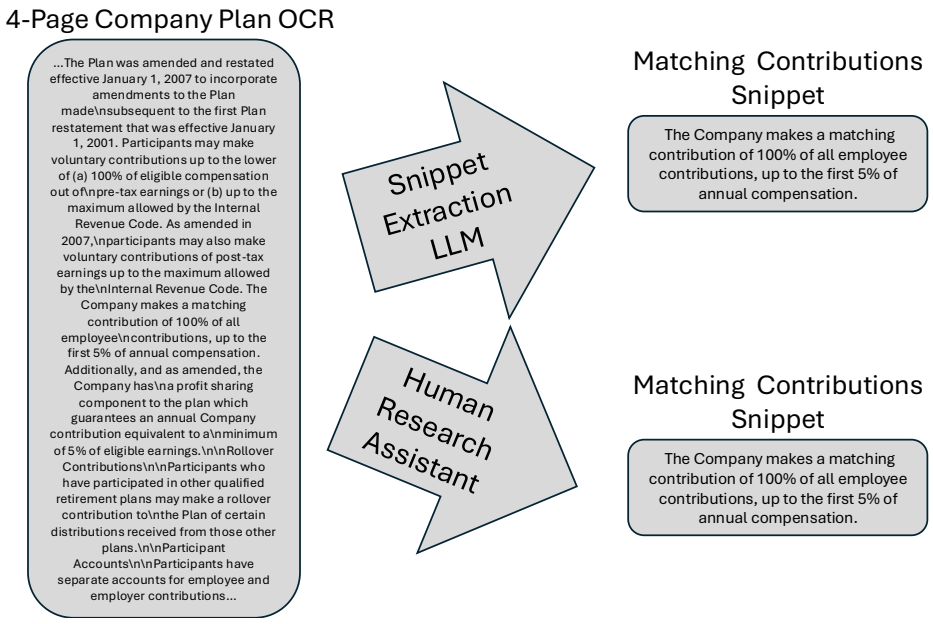


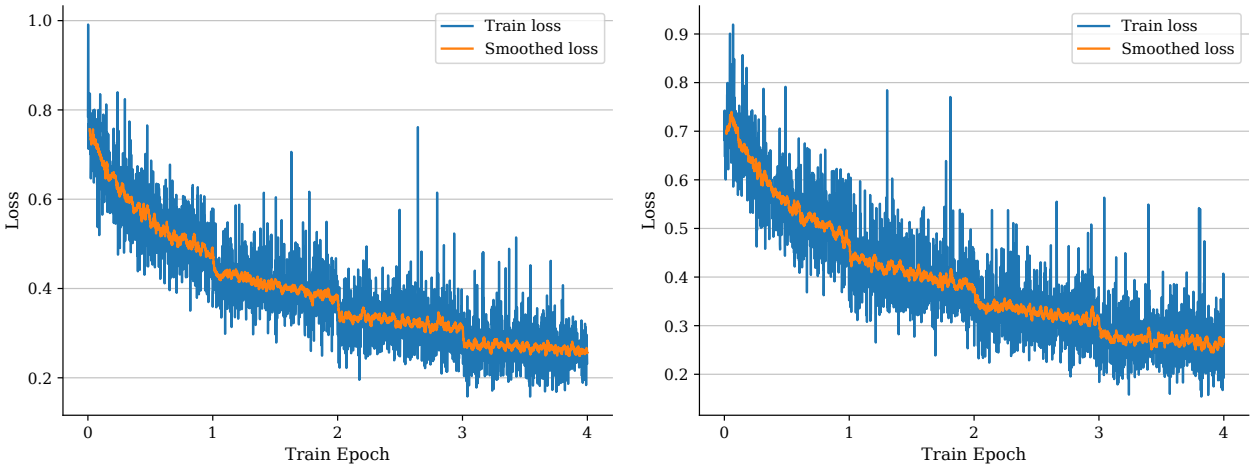
Figure 4: Demonstration of snippet extraction input and output using JetBlue's 2007 retirement plan. The hand-collected snippet is identical to the LLM-extracted snippet even though JetBlue was not a training example for the LLM.

Following fine-tuning, we inference each snippet extractor on the out-of-sample partition of our hand-collected dataset to generate realistic text snippets for training the next set of LLMs, the 'tabulators,' which will encode the retirement matching information from the plan documents as an easily-analyzable table. That is, we carry out $g_1(\mathbb{2})$ and $g_2(\mathbb{1})$. The model outputs are used to train the tabulator LLMs downstream. The tabulator trained on partition $\mathbb{1}$ (resp. $\mathbb{2}$) is trained using snippets from $g_2(\mathbb{1})$ (resp. $g_1(\mathbb{2})$). This ensures that none of the tabulators are trained using any outputs of an in-sample LLM, making our accuracy estimates on the test set realistic for true out-of-sample data. The tabulators are fine-tuned and inferenced using retrieval-augmented generation prompts, described in the following Section 2.3.3.

2.3.3 Retrieval-Augmented Generation and Plan Feature Tabulation

The LLMs that code plan features in tabular format ('tabulators') are fine-tuned in a similar way starting with Llama 3.1, one on each partition of the train set. Here, training lasts for four epochs (meaning that the LLM sees each train example four times—approximately 142,000 examples in total), uses the `adamw_torch` optimizer, and alters around 1% of the model weights. The quantity minimized during training ('loss') is depicted in Figure 5 as a function of the number of epochs. We refer to the tabulator trained on partition $\mathbb{1}$ (resp. $\mathbb{2}$)

as $f_1(\cdot)$ (resp. $f_2(\cdot)$), and the composition $f_1 \circ g_1(\mathbb{2})$ denotes the results of inferencing f_1 on partition $\mathbb{2}$ using prompts generated with snippets from g_1. With this notation, it is clear that these results provide a realistic estimate of accuracy in the wild, since neither f_1 nor g_1 had any retirement plans from $\mathbb{2}$ in its train set.



(a) Training loss for the LLM trained on partition $\mathbb{1}$.

(b) Training loss for the LLM trained on partition $\mathbb{2}$.

Figure 5: Training loss for the two LLMs that tabulate retirement matching information. Each epoch is an entire pass through the training data. After four epochs, the LLM has seen each example four times.

The tabulators are inferenced using RAG prompts, a prompting technique introduced in Lewis et al. (2021) to improve model accuracy in knowledge-intensive tasks (see Gao et al. (2024) for a recent review). In our RAG implementation (Prompt 2), the five most similar text snippets from our in-sample dataset alongside their correct tabular match schedules are provided alongside an out-of-sample query to which the answer is unknown. We use the `sentence_transformers` library (Reimers and Gurevych (2019)) to compute embeddings and calculate similarity scores.

Prompt 2: RAG For Matching

INSTRUCTIONS
You are a legal expert who is experienced with understanding firms' retirement plans and the rules which determine when and how much employers contribute to retirement plans. This is a full-length document from a firm's Form 5500 filings from the year [YYYY] that contains information about retirement saving contributions. The document to code will appear at the end of this prompt within < < < > > > . Your objective is to summarize the matching formula which was applicable to the plan year [YYYY] in tabular form. The goal will be to provide a simple way of consistently characterizing employer matching formulas. Make sure to include information about matching contributions as well as nonmatching / nonelective / profit-sharing contributions. We are only interested in information only for the year [YYYY], not past years. I mention this because occasionally there will be updated information provided about matching rules associated with past years in the document. Most frequently this happens when the matching program is "More complicated". Here are a few additional details. If the match schedule involves two different matching rates, cap_2 should be the amount of additional (not total) contribution the worker would need to make to capture the full match. For example, if the employer matches 100% on the first 3% and 50% on the next 2% of the worker's income, please code cap_2 as 2%. Likewise, cap_3 should be the amount of additional contribution the worker would need to make to capture the full match. Always include six columns, three for match rate and three for cap rate. If there is no value available for higher matches and caps, please report "NA". Now we will proceed with several examples.
EXAMPLES
Here are some relevant excerpts from other plans with similar textual content:
< <Input language for the year [YYYY1]: [EXAMPLE1]> >
Correct output: [TABLE1]
< <Input language for the year [YYYY2]: [EXAMPLE2]> >
Correct output: [TABLE2]
< <Input language for the year [YYYY3]: [EXAMPLE3]> >
Correct output: [TABLE3]
< <Input language for the year [YYYY4]: [EXAMPLE4]> >
Correct output: [TABLE4]
< <Input language for the year [YYYY5]: [EXAMPLE5]> >
Correct output: [TABLE5]
DOCUMENT TO CODE
< < <Input language for the year [YYYY]: [DOCUMENT] > > >

To find the five most semantically similar text snippets, we use the cosine similarity score. The cosine similarity of two embeddings v and w is given by $\cos\theta = \frac{v \cdot w}{\|v\|\|w\|}$, which grows as the angle between the vectors shrinks. Intuitively, it is common to think of vector embeddings as storing semantic content in direction (Mikolov et al. (2013)), and in keeping, the cosine similarity score is maximized when the vectors are parallel and minimized when the vectors are antiparallel.

In our implementation, the snippet extractor is first run on the out-of-sample query. Then, a lightweight LLM, MiniLM-L12-v2 (Wang et al. (2020)), calculates the cosine similarity between the embedding of the out-of-sample snippet and all snippets in our in-sample dataset. The cosine similarities are ranked, and the five most similar snippets are appended to the prompt, alongside their corresponding correct output (which comes from the hand-collected data). To ensure that all RAG examples are distinct, the ranked cosine similarities are filtered by the plan identifier varible, meaning that a snippet in the top five is excluded

from the RAG prompt if a snippet with higher semantic similarity has comes from the same retirement plan in a different year. This typically occurs when a company with a highly similar plan to the out-of-sample query has maintained the same plan description for multiple years—in this case, the RAG prompt will contain only one example from that company.

The prompt contains the full OCR-extracted text for the out-of-sample query to prevent cascading errors. Including the full document removes the snippet extraction process as a significant error channel,[5] as snippet extraction merely gathers high quality RAG examples. The tabulator LLM sees the entire plan description, regardless of the content of the snippet.

RAG prompting is a key performance enhancer. Due to their high semantic similarity to the unknown query, the RAG examples effectively 'give the answer' to the LLM, or at the very least, examples of the most likely plan structure with different numbers substituted in.

2.3.4 Out of Sample Inference

Each of the 1,149,649 true out-of-sample plan-year observations is double-coded, once by the LLMs trained on the partition $\mathbb{1}$, $f_1 \circ g_1(\cdot)$, and once by $f_2 \circ g_2(\cdot)$, which were trained on partition $\mathbb{2}$. The snippet extractors and tabulators are paired based on their training data, so the performance of each pair is completely independent. This independence enables us to condition on whether the LLM outputs are identical for a given plan. We also set the temperature parameter to 0.01, which further reduces random variation.[6]

Here, we report basic analytics describing the pipeline output for employer match schedules. Corresponding analytics for auto-features and vesting schedules are reported in Sections B.0.1 and B.0.2.

	More complicated	Simple
2004	22.8%	77.2%
2023	17.8%	82.2%
Total	19.7%	80.3 %

Table 1: Share of matching plans classified as more complicated vs. simple.

Notes: This table presents the share of plans in each of the two possible categories for 2004, 2023, and all years (Total). These proportions are similar to those in the hand-collected data (in which 70% of plans were adjudicated to be 'simple'.)

[5]For example, the case where there *is* a description of a matching contribution that snippet extractor *fails to find* would be particularly troublesome. With our implementation, that error affects only the RAG examples, not the plan text that the tabulator is provided.

[6]The temperature parameter controls the spread of the probability distribution when selecting each output token. A lower temperature tightens the distribution around the most likely tokens, while a higher temperature widens the distribution. Our choice of 0.01 makes the output effectively deterministic.

For the plans in our final dataset, the LLMs produced identical output describing the match schedule in 88.3% of cases (summing the bottom row of Table 2). Given that the LLMs agree that the plan is simple, they produce identical tables for 96.3% of plans (Table 3). Table 1 breaks down the proportion of plans that the LLMs could encode in tabular format for 2004, 2023, and across the entire sample. The exact prompts used in training and inference are given in Prompts 1 and 2.

Models agree	Agree, more complicated	Agree, simple	Disagree on distinction	Total
False	0%	2.7%	9.1%	11.8%
True	19.7%	68.6%	0%	88.3%

Table 2: Match schedule agreement proportions in out-of-sample universe for $f_1(x; g_1(x))$ and $f_2(x; g_2(x))$.

Notes: The leftmost column, 'Models agree' is a simple boolean that flags whether the models produce identical output. The 'Agree, more complicated' column is whether the models both classify the plan as 'More complicated'. The 'Agree, simple' column is a boolean flagging whether both models produced a tabular description of the match schedule. The 'Disagree on distinction' column is whether one model classified the plan as 'more complicated' while the other classified it as 'simple'. The 'Total' column is the row-wise sum, meaning that the models produce the same output for 88.3% of observations.

Models agree	Agree, more complicated	Agree, simple	Disagree on distinction
False	0%	3.7%	100%
True	100%	96.3%	0%

Table 3: Match schedule agreement proportions in out-of-sample universe conditioned on agreement type between $f_1(x; g_1(x))$ and $f_2(x; g_2(x))$.

Notes: The columns are defined as in Table 2, however here each column is normalized to 1, which allows us to note that, conditional on the models agreeing that a plan is 'simple', they produce the same match schedule in 96% of cases.

2.3.5 Constructing the Final Dataset

The final step to generate usable data is to parse the textual output from the LLM. Since the LLM output follows a predictable pattern, this is done using regular expressions. The extracted text is compared to expected characteristics[7] and flagged if there are any errors. Since each plan is double-coded, the error-free LLM outputs are then compared to each

[7]The characteristics are: the number of values reported, that the number of column titles matches the number of values, and the entry types (numeric or string).

other. If the LLMs produce identical output,[8] the data is accepted and processed into a dataframe. If the LLMs do not produce identical output, the observation is flagged and any corresponding LLM fields in the final dataset are left missing.

Finally, there is one special case worth mentioning. This is if there is no mention of particular plan feature in the plan description. In this scenario, the tabulator LLMs have been trained to classify the plan as 'simple' and produce a table with all 'NA' values—indistinguishable from a plan that clearly states the nonexistence of the feature. As the economic relevance of the distinction between not mentioning a feature in the plan documents and an explicit lack of the feature is negligible, we elect to conflate the two to reduce noise in LLM output, which is increasing in the number of classification categories.

2.4 Audit

An audit, with two new research assistants hand-coding data collected by the LLMs is ongoing. Results will be available in March 2026.

3 The evolving landscape of DC retirement plans

3.1 Matching

After describing the distribution of matching formulas in our data, we use our LLM-extracted panel to show four facts about how employer matching has evolved over the past two decades.

The distribution of employer matching formulas. Figure 6 illustrates the heterogeneity which exists in employer match schedules. However, Table 4 gives the ten most common matching formulas in our dataset, revealing significant concentration in just a few schedules. Over one-third of plans (40.5%) offer no employer matching contributions. Among plans that do offer matching, the most prevalent formula is the Basic Safe Harbor structure: dollar-for-dollar matching on the first 3% of employee contributions and 50-cent matching on contributions between 3% and 5% of salary (10.2% of plans). The next most common arrangements are a 50% match up to 6% of salary (9.6%) and a 100% match up to 4% (5.4%). These three formulas alone account for over one-quarter of all plans offering employer contributions, suggesting that plan sponsors gravitate toward a small set of conventional matching structures.

[8] "Identical" is used loosely here. In practice, the regular expression is flexible enough that the output need not be perfectly identical, which is critical since LLMs are inherently stochastic and may include random variations to the table structure that do not imply the output is incorrect, like including extra whitespace or column separators.

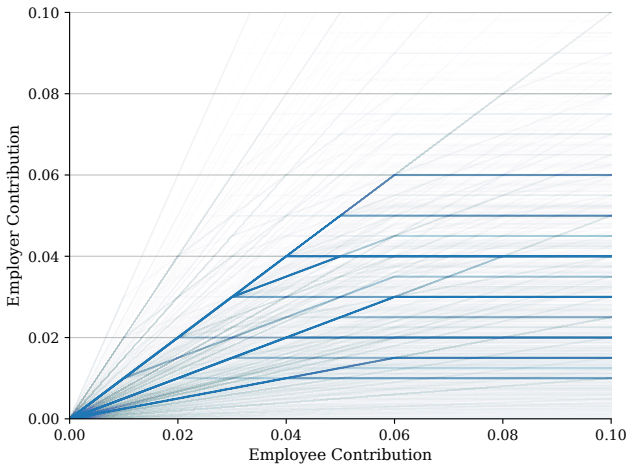


Figure 6: All nonzero match schedules in our universe shaded by prevalence.

Figure 7 gives a cross-sectional snapshot of matching plan generosity in 2023. Panel (a) reveals considerable heterogeneity in matching caps, with the most common threshold being 5% of salary (22% of plans), followed closely by 6% (21%) and 4% (13%). The prevalence of these specific thresholds likely reflects safe harbor requirements and conventional benchmarks in plan design. Panel (b) shows that employers overwhelmingly favor either full (100%) matching or 50% matching on initial contributions, with 42% of plans offering dollar-for-dollar matching and 18% offering 50-cent-per-dollar matching on the first tier. Very few plans adopt intermediate match rates. Panel (c) shows that maximum employer contributions cluster strongly around 3% and 4% of salary, with the modal maximum contribution of 3% aligning precisely with the Basic Safe Harbor minimum.

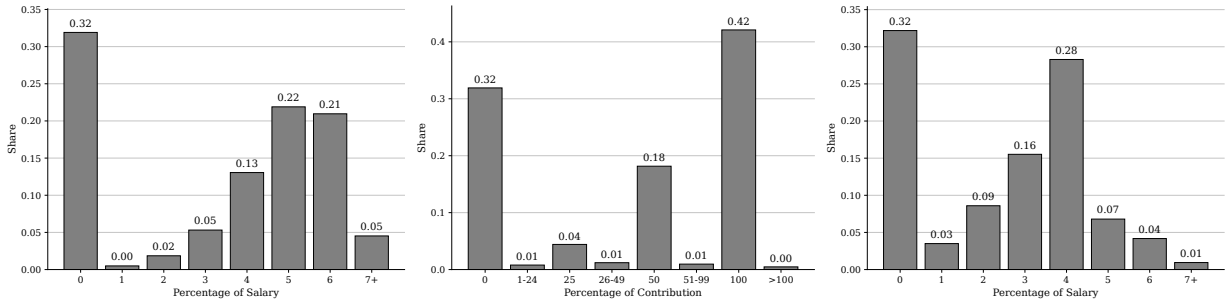
Fact 1: Employer matches have become substantially more generous. Figure 8 tracks three measures of plan generosity from 2004 through 2023, showing raw means alongside a composition-adjusted profile.[9] All three measures show a notable decline during the Great Recession, consistent with widespread match suspensions during this period. Beyond these temporary falls, each measure has trended upward over the two decades. The mean maximum employer match rose from 1.5% of salary in 2003 to 2.4% in 2023—a 60% increase—and the median jumped from 1.0% to 3.0%. The composition-adjusted measures increase by less, showing that a substantial portion of the increase in generosity arises due to new plans entering rather than existing plans changing their formulas.

[9]The composition-adjusted profile plots the year coefficients from the fixed effects regression $y_{it} = \hat{\alpha}_i + \hat{\beta}_t \text{Year}_t + \varepsilon_{it}$, where we set 2003 as the base year and include the average fixed effect of plans observed in that year. This approach isolates changes in generosity within continuing plans, holding the composition of firms fixed at 2003 levels.

Plan	Proportion
0-0-0-0-0-0	40.5%
100-3-50-5-0-0	10.2%
50-6-0-0-0-0	9.6%
100-4-0-0-0-0	5.4%
50-4-0-0-0-0	3.5%
100-3-0-0-0-0	2.7%
100-5-0-0-0-0	2.7%
25-6-0-0-0-0	2.6%
100-6-0-0-0-0	2.5%
25-4-0-0-0-0	2.0%

Table 4: Top ten most common plans.

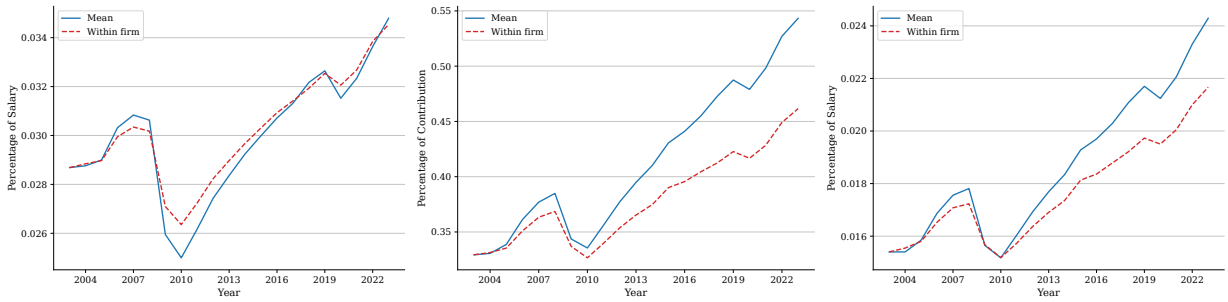
Notes: This table presents the ten most common match schedules in our data. When calculating the proportions, we restrict our sample to plans with a match that we could encode in tabular format. The 'Plan' column follows the format of Figure 3: 100-3-50-5-0-0 means that the employer matches 100% of employee contributions up to 3% of salary, 50% of contributions from 3% to 5% of salary, and makes no matching contributions thereafter.



(a) Matching cap. (b) Match rate on first dollar. (c) Maximum employer match.

Figure 7: Matching plan generosity in 2023.

Notes: This figure presents summary measures of matching plan generosity across all plans in the year 2023 that could be codified. Each plot shows the proportion of plans with at each level of generosity. In (a), the 'matching cap' is defined as the percentage of salary an employee must defer to fully exhaust the match—beyond the cap, there are no further employer matching contributions. In (b), the 'match rate on the first dollar' is defined as the employer match rate up to the first matching cap. In (c), the 'maximum employer match' is defined as the percentage of salary the employer would contribute if the participant fully exploited the match by deferring at least the matching cap. The x-axis labels define the center of a bin (with the zero bin in (a), (b), and (c) representing values in the interval $[0, 0.5)$).

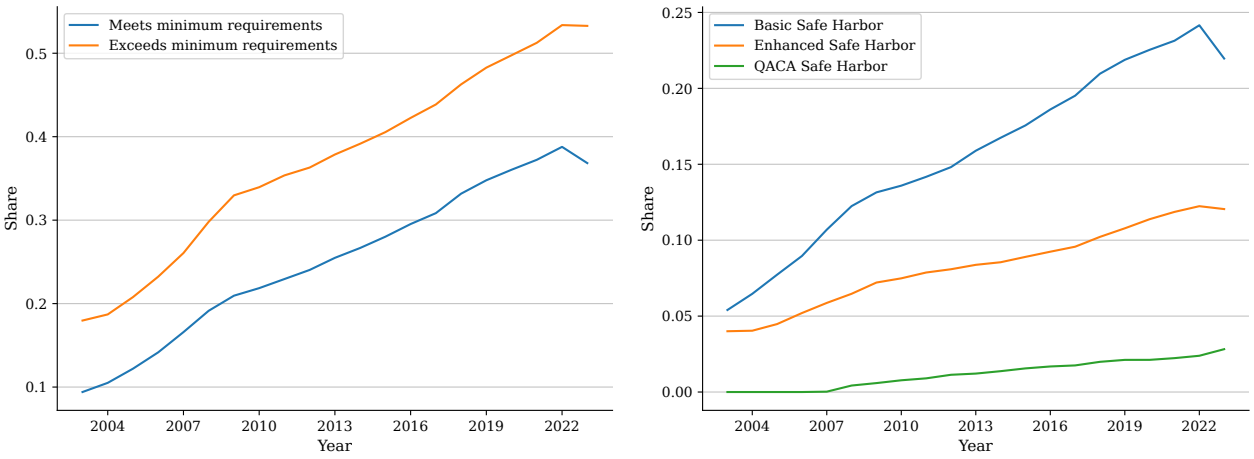


(a) Matching cap. (b) Match rate on first dollar. (c) Maximum employer match.

Figure 8: Matching plan generosity over time.

Notes: This figure presents summary measures of plan generosity for each year in our data. The sample is those plans we could codify—we do not require nonzero employer matching. In each subplot, the 'within firm' line plots the year coefficients in the fixed effects regression $y_{it} = \hat{\alpha}_i + \hat{\beta}_t \text{Year}_t + \varepsilon_{it}$. 2003 is the base year and, in the 'within-firm' line, we set the fixed effect to be the average fixed effect of plans observed in that year. In (a), the 'matching cap' is defined as the percentage of salary at which the employee has fully exhausted the match. In (b), the 'match rate on the first dollar' is defined as the employer match rate up to the first matching cap. In (c), the 'maximum employer match' is defined as the percentage of salary the employer would contribute if the participant fully exploited the match.

Fact 2: The Safe Harbor formula has become the dominant matching design. While the match schedule is a firm choice, safe-harbor exemptions for certain plan characteristics mean that federal regulations can shape these choices. Figure 9 documents this rise: panel (a) shows that the share of matching plans meeting or exceeding safe harbor requirements grew from approximately 10% in 2004 to nearly 55% by 2022, with most of this increase representing plans that exactly meet minimum safe harbor thresholds rather than exceeding them. Panel (b) decomposes this trend by safe harbor type, showing that the Basic Safe Harbor formula grew from roughly 5% of plans in 2004 to 24% by 2022. The Enhanced Safe Harbor formula showed more modest growth from about 5% to 12% over the same period, while QACA Safe Harbor provisions remained relatively uncommon throughout. This growth has come largely at the expense of previously popular formulas like "50% on 6%," which declined from roughly 20% to 14% of plans. Safe Harbor status offers employers relief from nondiscrimination testing, creating a strong regulatory incentive that appears to be reshaping plan design nationwide.



(a) Safe harbor matching plans.

(b) Proportion of each safe harbor category.

Figure 9: Safe harbor matching plans over time.

Notes: This figure gives the prevalence of safe harbor plans in each year for which we have data. The sample is those plans that we can codify that offer a nonzero matching contribution. In (a), 'Meets minimum requirements' is defined as exactly following one of the three safe harbor specifications (Basic, Enhanced, or QACA). 'Exceeds minimum requirements' is defined as offering a match more generous than the minimum requirements, while still satisfying the definition of safe harbor. In (b), each line plots the proportion of plans that exactly follow the minimum requirements for each of the safe harbor definitions.

Fact 3: Employer matching formulas are becoming increasingly concentrated. The dominance of Safe Harbor is part of a broader pattern: despite employers choosing from over 1,000 distinct matching formulas observed in our data, plan design is converging toward

a small number of dominant options. The top 5 formulas accounted for 47% of all plans in 2003 but 60% by 2022. The Herfindahl-Hirschman Index (HHI) rose by 45% over the same period, from 0.073 to 0.106 (Figure 10). Importantly, the number of distinct formulas in use has remained roughly stable (around 420–470 per year), so the concentration increase reflects a shift toward the top formulas rather than a disappearance of rare designs. As plans converge on the Basic Safe Harbor formula and a handful of simple dollar-for-dollar structures, the long tail of less common designs has shrunk in relative terms. The prevalence of specific thresholds documented in Figure 7—matching caps clustering at 5–6% of salary, match rates clustering at 50% and 100%—is consistent with plan sponsors gravitating toward a small set of conventional, regulation-influenced plan designs.

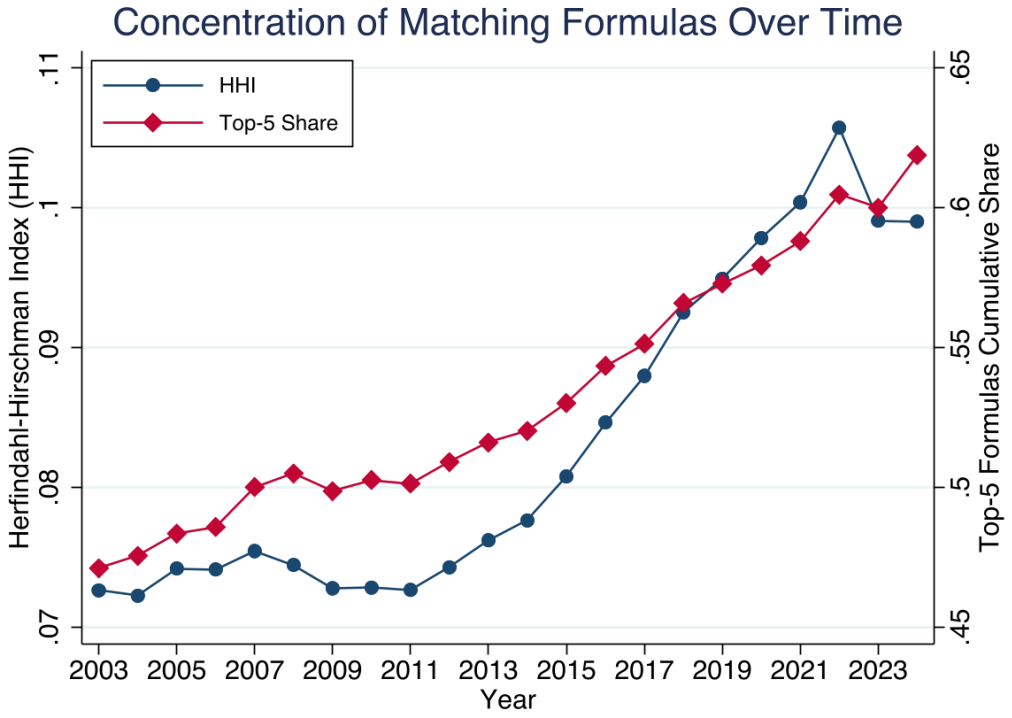


Figure 10: Concentration of matching formulas over time. The HHI (blue, left axis) and the cumulative market share of the top 5 formulas (red, right axis) both trend upward steadily from 2003 to 2022, indicating that plans are converging toward a smaller set of dominant designs.

Fact 4: Matching formulas rarely change, and Safe Harbor formulas are the stickiest of all. The concentration patterns documented above are reinforced by the remarkable persistence of matching formulas once adopted. In any given year, only about 4% of plans change their matching formula, and this hazard rate declines with tenure: from 4.3%

in the first year to around 2.3% by year 10. After a decade, roughly 70% of plans are still using the same formula they started with; after 15 years, this share is still above 60%. This inertia helps explain the composition-adjusted finding from Figure 8: much of the increase in average generosity over time comes from new plans entering with more generous formulas rather than from existing plans raising their plan generosity.

Safe Harbor formulas are especially persistent. Their cumulative change probability after 10 years is just 14%, compared to 32% for non–Safe Harbor formulas, making them more than twice as stable (Figure 11). After 15 years, only about 19% of Safe Harbor plans have ever changed their formula, versus 41% of other plans. The annual hazard rate for exact Safe Harbor plans hovers around 1–2%, compared to 3–5% for non–Safe Harbor plans (Figure 12). The regulatory benefits of Safe Harbor status appear to create a strong lock-in effect: once a plan qualifies for the nondiscrimination testing exemption, there is little incentive to redesign. Combined with the rapid adoption documented in Figure 9, this stickiness implies that the Safe Harbor formula's dominance is likely to persist and deepen in the years ahead.

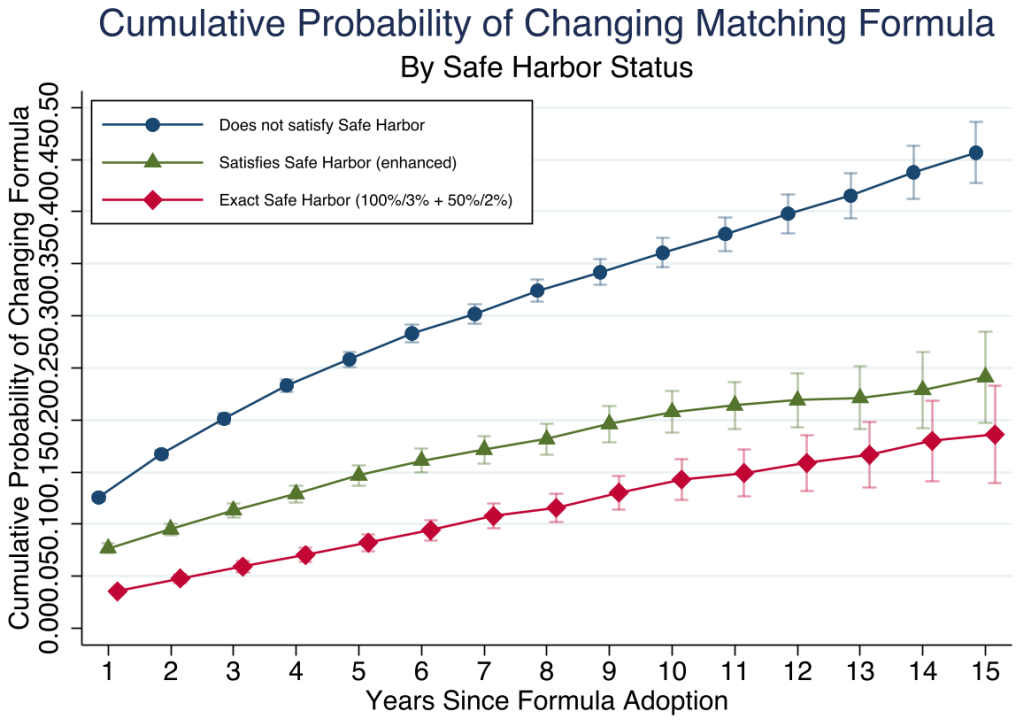


Figure 11: Cumulative probability of changing formula, by Safe Harbor status. Exact Safe Harbor plans (red) are far less likely to change than non–Safe Harbor plans (blue), with enhanced Safe Harbor plans (green) falling in between.

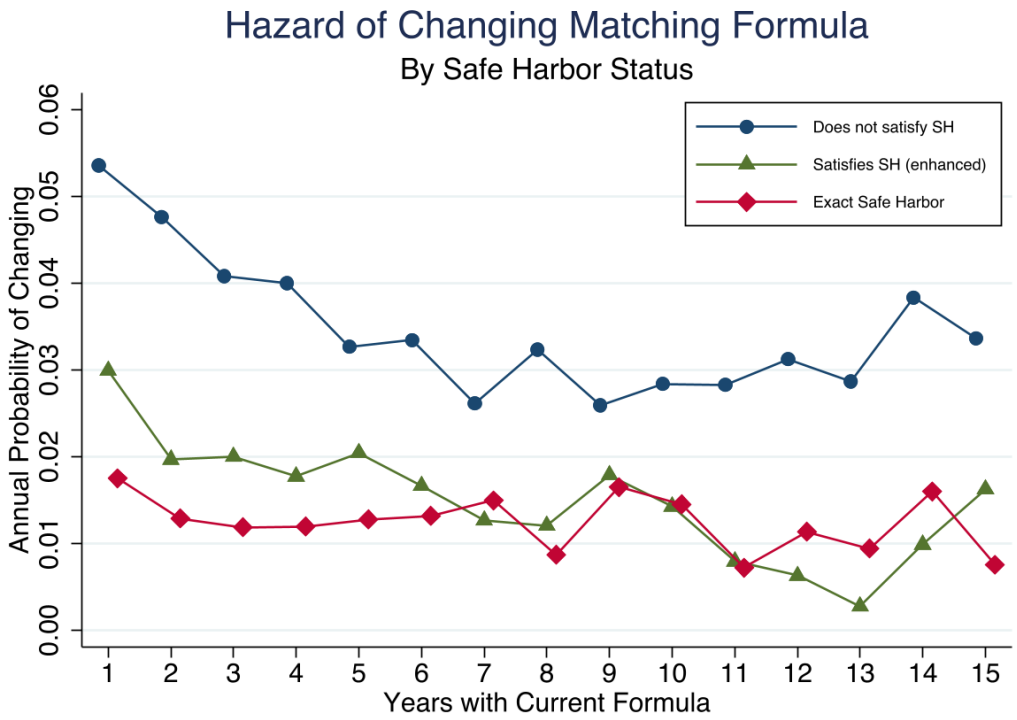
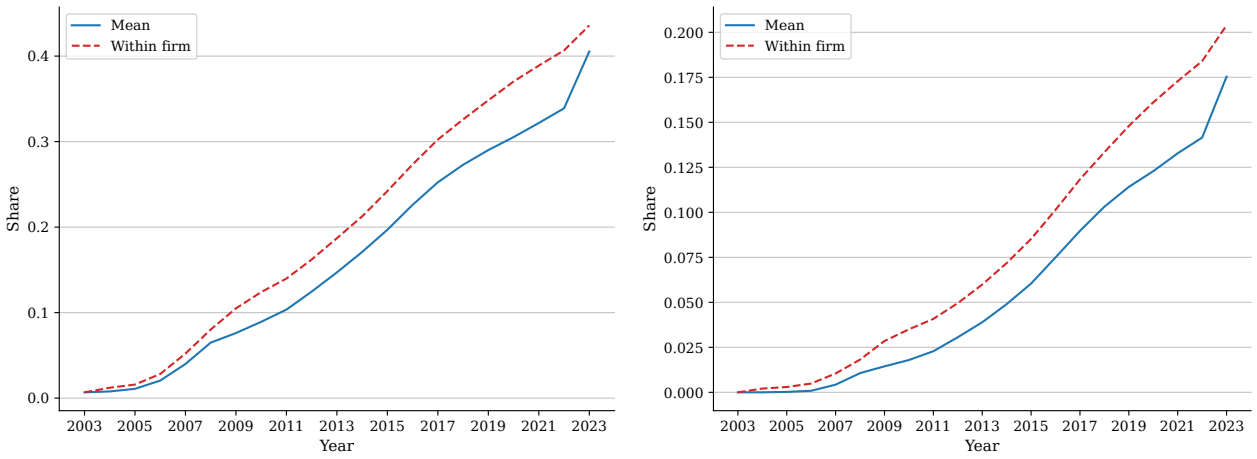


Figure 12: Annual hazard rate of changing matching formula, by Safe Harbor status and years with current formula. Non–Safe Harbor plans face a 3–5% annual probability of change, while exact Safe Harbor plans hover around 1–2%.

3.2 Auto-enrollment

Figure 13 documents one of the most dramatic shifts in retirement plan design over the past two decades: the rise of automatic enrollment from a rarely-used behavioral nudge to a mainstream feature of employer-sponsored plans. Panel (a) shows that auto-enrollment adoption grew from essentially zero in 2003 to approximately 40% of plans by 2023. The composition-adjusted trend tracks the mean closely, indicating that this increase reflects genuine policy changes by continuing firms rather than compositional turnover. Panel (b) shows that automatic escalation followed a similar but slightly delayed adoption curve, growing from near zero to roughly 17% of plans by 2023.



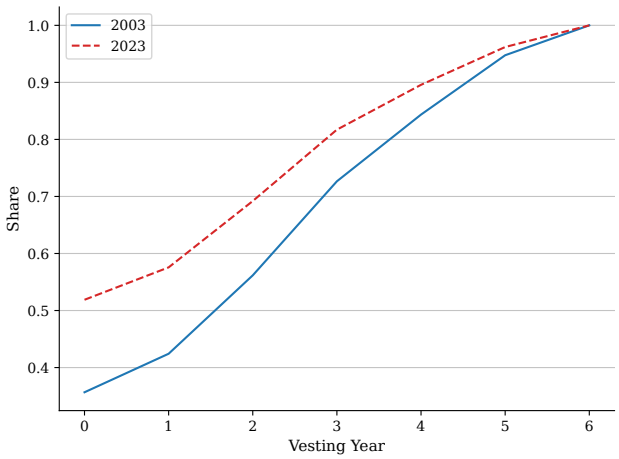
(a) Proportion offering auto-enrollment over time (b) Proportion offering auto-escalation over time

Figure 13: Presence of auto-features over time.

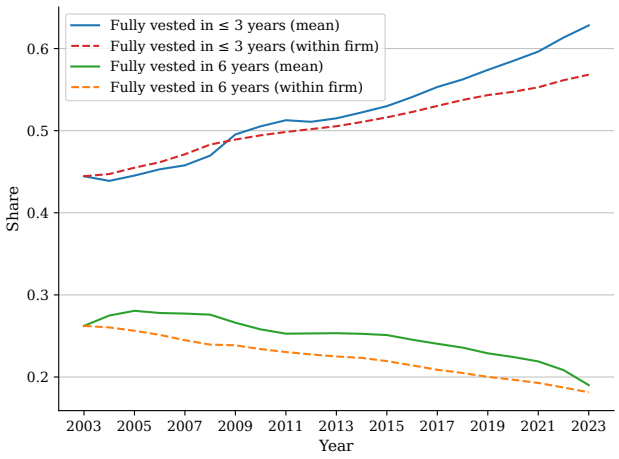
Notes: This figure summarizes the presence of auto-features for each year in our data. We do not do any filtering before calculating the proportions for the 'mean' line. The 'within firm' line plots the year coefficients in the fixed effects regression $y_{it} = \hat{\alpha}_i + \hat{\beta}_t \text{Year}_t + \varepsilon_{it}$. In the 'within-firm' graph, we set 2003 as the base year and include the average fixed effect of plans observed in that year. In (a), a plan is deemed to offer auto-enrollment if it has a nonzero initial deferral. In (b), a plan is deemed to offer automatic escalation in a similar manner—if it offers a nonzero automatic increase rate.

3.3 Vesting

Figure 14 summarizes the evolution of vesting schedules in our data. Panel (a) compares the average proportion vested at each year of plan participation for plans in 2003 versus 2023, showing a shift toward faster vesting over time. Panel (b) quantifies this shift by tracking the proportion of plans that fully vest within three years and the proportion that require the maximum six years. The share of plans with quicker vesting (three years or less) increased from approximately 40% in 2003 to nearly 65% by 2023, while the share requiring six years to fully vest declined from roughly 30% to 20% over the same period. The composition-adjusted trends closely track the means.



(a) Average proportion vested each year for plans in 2003 vs. 2023



(b) Proportion of plans in each year that fully vest in 6 years or in fewer than 3 years.

Figure 14: Evolution of Vesting

Notes: Each of the 'within-firm' lines plots the year coefficients in the fixed effects regression $y_{it} = \hat{\alpha}_i + \hat{\beta}_t \text{Year}_t + \varepsilon_{it}$. We set 2003 as the base year and include the average fixed effect of plans observed in that year. In (a), the 'mean' line plots the proportion vested in each year of plan participation for the earliest year and latest year in our data. In (b), the blue 'mean' line plots the proportion of plans that fully vest in no more than three years; the green 'mean' is the proportion of plans that require six years to fully vest.

4 Conclusion

This paper demonstrates how large language models can overcome traditional barriers to comprehensive data collection in economics. By using a hand-collected dataset as training data, we scaled the extraction of retirement plan characteristics from 6,200 plans to nearly 150,000 plans spanning two decades—an expansion that required only a small team of four authors rather than a team of fifteen research assistants. This dramatic reduction in data collection costs makes population-level analysis feasible where it was previously prohibitively expensive.

Beyond documenting the evolution of retirement plan design, our dataset enables new avenues for economic research. The comprehensive coverage of matching formulas, vesting schedules, and auto-enrollment provisions, linked to administrative tax records, provides the variation needed to identify key parameters in household decision-making. Ongoing work will use plan changes affecting millions of workers to estimate the distribution of elasticity of intertemporal substitution and to study how features like automatic enrollment interact with traditional price incentives in shaping retirement wealth accumulation. More broadly, our methodology is an example of how LLMs can transform empirical economics by making large-scale extraction of structured data from unstructured regulatory filings practical and scalable.

A Extra Exhibits

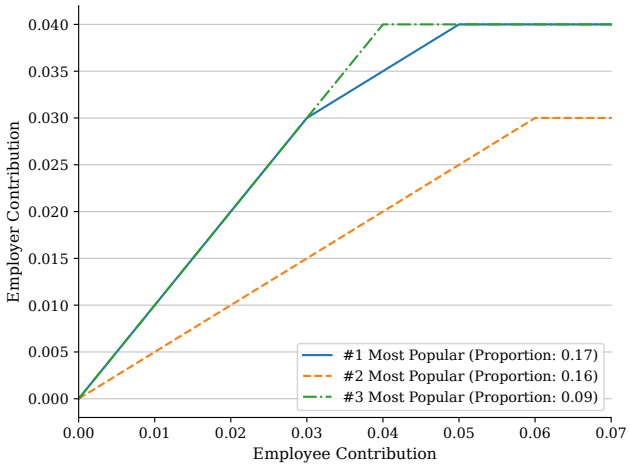
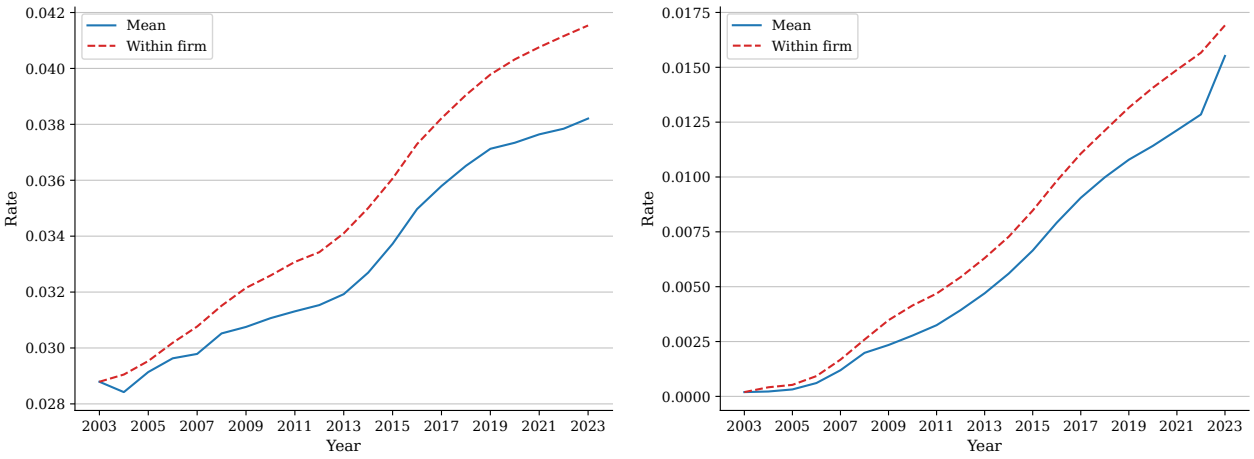


Figure 15: Top three most common match schedules.

Notes: This figure presents the three most common match schedules in our data. The proportions differ from those in Table 4 since we restrict our sample in this figure to plans that (1) we could encode in tabular format, and (2) offer a nonzero matching contribution.



(a) Average default rate for firms offering ae.

(b) Average default rate for all firms.

Figure 16: Automatic enrollment generosity over time.

Notes: This figure summarizes the generosity of automatic enrollment in our sample. In both subplots, we filter the data to remove plans for which we were unable to describe automatic enrollment in tabular format. The 'mean' line plots a simple average of the default rate for the filtered sample. The 'within firm' line plots the year coefficients in the fixed effects regression $y_{it} = \hat{\alpha}_i + \hat{\beta}_t \text{Year}_t + \varepsilon_{it}$. 2003 is the base year and in the 'within firm' graph with the average fixed effect of plans observed in that year. In (a), we further filter the data to require a nonzero default rate. In (b), we do not require a nonzero default rate.

B LLM Pipeline for Other Plan Features

B.0.1 Auto-features

Table 5 demonstrates the LLM output for a plan that has simple auto-enrollment and auto-escalation. The RAs who hand-coded plan data coded auto-enrollment and auto-escalation as simple if the same rules applied to all workers. That is, if any new enrollee in the plan would automatically defer the same straightforward percentage of salary, which would increase in each year by a constant increment to a fixed percentage-based cap. Such plans can be fully described in the tabular format shown in Table 5. We also allow for either (or both) of `auto_enrollment_offered` and `auto_increase_offered` to be classified as complicated, independently of one another. As with matching, this could happen if either of the auto-features applies only to some employees, is defined in terms of dollar amounts, or has special eligibility requirements.

In the case of auto-features, we trained the LLM to codify automatic enrollment and automatic increases in the same response. As these features are highly similar and often even described in the same sentence, this does not add significant noise to the data collection process. The exact text of the prompts used to tabulate auto-features is given in Prompts 3 and 4. For the plans in our final dataset, the LLMs produced identical output describing

	Value
`auto_enrollment_offered`	*Yes*
`initial_deferral`	0.03
`auto_increase_offered`	*Yes*
`auto_increase_amount`	0.01
`auto_increase_cap`	0.06

Table 5: Output of the LLM for *simple* auto-features.

	More complicated	Simple
2004	0.0%	100.0%
2023	0.2%	99.8%
Total	0.1%	99.9%

Table 6: Share of auto-features classified as more complicated vs. simple.

Notes: This table presents the share of auto-features in each of the two possible categories for 2004, 2023, and all years (Total). These proportions are similar to those in the hand-collected data (in which 98% of plans were adjudicated to be 'simple'.)

the auto-enrollment and auto-escalation provisions in 97.9% of cases (summing the bottom row of Table 7). Given that the LLMs agree that the plan is simple, they produce identical tables for 98.7% of plans (Table 8). Table 6 breaks down the proportion of plans that the LLMs could encode in tabular format for 2004, 2023, and across the entire sample.

Models agree	Agree, more complicated	Agree, simple	Disagree on distinction	Total
False	0%	1.3%	0.74%	2.04%
True	0.14%	97.8%	0%	97.94%

Table 7: Auto-features agreement proportions in out-of-sample universe for $f_1(x; g_1(x))$ and $f_2(x; g_2(x))$.

Notes: The leftmost column, 'Models agree' is a simple boolean that flags whether the models produce identical output. The 'Agree, more complicated' column is whether the models both classify the plan as 'More complicated'. The 'Agree, simple' column is a boolean flagging whether both models produced a tabular description of the both auto-enrollment and auto-escalation. The 'Disagree on distinction' column is whether one model classified the plan as 'more complicated' while the other classified it as 'simple'. The 'Total' column is the row-wise sum, meaning that the models produce the same output for 97.94% of observations.

Models agree	Agree, more complicated	Agree, simple	Disagree on distinction
False	0%	1.3%	100%
True	100%	98.7%	0%

Table 8: Auto-features agreement proportions in out-of-sample universe conditioned on agreement type between $f_1(x; g_1(x))$ and $f_2(x; g_2(x))$.

Notes: The columns are defined as in Table 7, however here each column is normalized to 1, which allows us to note that, conditional on the models agreeing that a plan is 'simple', they produce the same description of the auto-features in 98.7% of cases.

Prompt 3: Snippet Extraction for Auto-Features

INSTRUCTIONS
You are a legal expert analyzing a snippet of text extracted from a firm's Form 5500 filings for the year [YYYY], which relates to their retirement plan.
Extract verbatim the sections from the text that describe any auto-enrollment features associated with the plan, as well as to which groups of workers the rules apply (all employees vs new hires, etc). Please also be sure to include any language which describes default contribution rates (e.g., 3% of salary) for auto enrolled employees. Also make sure to include information about auto-escalation if it appears (e.g., increasing by 1% until the employee contributes 6% of salary). Copy the text describing auto-enrollment rules exactly as it appears in the plan language. Do not summarize. Do not make a bulleted list. Do not add any additional notes.
If you do not see any text describing auto enrollment features, your answer should be exactly "No mention of auto-enrollment.".
It is very important that you only provide the final output without any additional comments, notes, or remarks
CHECKLIST
Before finalizing your answer, please check it against this list of "do"'s and "don't"'s: - DO include too much text rather than too little text.
- DO copy the entire sentences and/or paragraphs verbatim without attempting to summarize. (The only exception here is that you can address OCR-related issues in the text to strip out extraneous characters, etc).
- DO include information about any dollar caps and/or information about eligibility criteria and/or rules which differ across different classes of workers.
- DO include any language describing default contribution amounts for plans which have auto-enrollment.
- DON'T try to summarize anything. Copy any text that is relevant for understanding auto-enrollment policies.
- DON'T add additional notes/annotations which aren't direct quotes from the document. Your responses should only include the plan language.
When in doubt, please err on the side of including too much, not too little, text in your snippets.
###PLAN TO ANALYZE###
[PLAN]

Prompt 4: RAG for Auto-Features

INSTRUCTIONS
You are a legal expert who is experienced with understanding firms' retirement plans and the rules which determine whether employees are automatically enrolled in the retirement plan (automatic enrollment or auto-enrollment) and whether their contributions increase automatically from year to year (automatic escalation or auto-escalation). This is a full-length document from a firm's Form 5500 filings from the year [YYYY] that contains information about retirement saving contributions. The document to code will appear at the end of this prompt within < < < > > >.
Your objective is to summarize the auto-enrollment and auto-escalation provisions of the plan, which were applicable to the plan year [YYYY] in tabular form. The goal will be to provide a simple way of consistently characterizing auto-enrollment and auto-escalation features. For auto-enrollment, make sure to include information about whether the plan has auto-enrollment and about the default contribution rate (e.g., 3% of salary) for auto-enrolled employees. For auto-escalation, make sure to include information about whether the plan offers auto-escalation, the auto-escalation increase rate (e.g., 1% increase every year), and the maximum contribution rate for auto-escalation (e.g., contribution automatically increase only up to 10%). We are only interested in information for the year [YYYY], not past years. I mention this because occasionally there will be updated information provided about auto-enrollment and auto-escalation rules associated with past years in the document.
Here are a few additional details. Always include five columns, two for auto-enrollment (whether it is offered and the default contribution rate) and three columns for auto-escalation (whether it's offered, the annual automatic increase rate, and the maximum contribution rate for auto-escalation). If there is no value available or no mention of auto-enrollment or auto-escalation, please report "NA". Now we will proceed with several examples.
EXAMPLES
Here are some relevant excerpts from other plans with similar textual content:
< <Input language for the year [YYYY1]: [EXAMPLE1]> >
Correct output: [TABLE1]
< <Input language for the year [YYYY2]: [EXAMPLE2]> >
Correct output: [TABLE2]
< <Input language for the year [YYYY3]: [EXAMPLE3]> >
Correct output: [TABLE3]
< <Input language for the year [YYYY4]: [EXAMPLE4]> >
Correct output: [TABLE4]
< <Input language for the year [YYYY5]: [EXAMPLE5]> >
Correct output: [TABLE5]
DOCUMENT TO CODE
< < <Input language for the year [YYYY]: [PLAN] > > >

B.0.2 Vesting

Table 9 presents the LLM output for a plan that offers a simple vesting schedule. In keeping with the other plan features, the RAs who hand-coded plan features coded vesting as simple if the same rules applied to all workers. For vesting, this means that any new enrollee in the plan would follow an identical vesting schedule. Such a plan can be fully described in the tabular format of Table 9. In accordance with matching and auto-features, a vesting schedule is classified as more complicated if the vesting schedule applies only to some employees or has special eligibility requirements. The exact prompts used to tabulate vesting schedules are given in Prompts 5 and 6.

For the plans in our final dataset, the LLMs produced identical output describing the

vesting schedule in 75.8% of cases (summing the bottom row of Table 11). Given that the LLMs agree that the plan is simple, they produce identical tables for 81.9% of plans (Table 12). Table 10 breaks down the proportion of plans that the LLMs could encode in tabular format for 2004, 2023, and across the entire sample.

	Value
`vesting_year0`	*NA*
`vesting_year1`	*NA*
`vesting_year2`	0.2
`vesting_year3`	0.4
`vesting_year4`	0.6
`vesting_year5`	0.8
`vesting_year6`	1.0

Table 9: Output of the LLM for a *simple* vesting schedule.

	More complicated	Simple
2004	22.2%	77.8%
2023	14.0%	86.0%
Total	17.2%	82.8%

Table 10: Share of vesting schedules classified as more complicated vs. simple.

Notes: This table presents the share of vesting in each of the two possible categories for 2004, 2023, and all years (Total). These proportions are similar to those in the hand-collected data (in which 72% of plans were adjudicated to be 'simple'.)

Models agree	Agree, more complicated	Agree, simple	Disagree on distinction	Total
False	0%	12.9%	11.2%	24.1%
True	17.2%	58.6%	0%	75.8%

Table 11: Vesting agreement proportions in out-of-sample universe for $f_1(x; g_1(x))$ and $f_2(x; g_2(x))$.

Notes: The leftmost column, 'Models agree' is a simple boolean that flags whether the models produce identical output. The 'Agree, more complicated' column is whether the models both classify the plan as 'More complicated'. The 'Agree, simple' column is a boolean flagging whether both models produced a tabular description of the vesting schedule. The 'Disagree on distinction' column is whether one model classified the plan as 'more complicated' while the other classified it as 'simple'. The 'Total' column is the row-wise sum, meaning that the models produce the same output for 75.8% of observations.

Models agree	Agree, more complicated	Agree, simple	Disagree on distinction
False	0%	18.1%	100%
True	100%	81.9%	0%

Table 12: Vesting agreement proportions in out-of-sample universe conditioned on agreement type between $f_1(x; g_1(x))$ and $f_2(x; g_2(x))$.

Notes: The columns are defined as in Table 11, however here each column is normalized to 1, which allows us to note that, conditional on the models agreeing that a plan is 'Simple', they produce the same vesting schedule in 81.9% of cases.

Prompt 5: Snippet Extraction for Vesting

INSTRUCTIONS
You are a legal expert analyzing a snippet of text extracted from a firm's Form 5500 filings for the year [YYYY], which relates to their retirement plan.
Extract verbatim the sections from the text that describe any vesting features associated with the plan, as well as to which groups of workers the rules apply (all employees vs new hires, etc). Please also be sure to include any language which describes the progression of vesting over time (e.g., 50% immediately, 75% after one year, 100% after two years). Copy the text describing vesting rules exactly as it appears in the plan language. Do not summarize. Do not make a bulleted list. Do not add any additional notes.
If you do not see any text describing vesting features, your answer should be exactly "No mention of vesting.".
It is very important that you only provide the final output without any additional comments, notes, or remarks
CHECKLIST
Before finalizing your answer, please check it against this list of "do"'s and "don't"'s:
- DO include too much text rather than too little text.
- DO copy the entire sentences and/or paragraphs verbatim without attempting to summarize. (The only exception here is that you can address OCR-related issues in the text to strip out extraneous characters, etc).
- DO include information about any dollar caps and/or information about eligibility criteria and/or rules which differ across different classes of workers.
- DO include any language describing vesting amounts for plans which have vesting.
- DON'T try to summarize anything. Copy any text that is relevant for understanding vesting policies.
- DON'T add additional notes/annotations which aren't direct quotes from the document. Your responses should only include the plan language.
When in doubt, please err on the side of including too much, not too little, text in your snippets.
PLAN TO ANALYZE
[PLAN]

Prompt 6: RAG for Vesting

INSTRUCTIONS
You are a legal expert who is experienced with understanding firms' retirement plans and the rules which determine whether employees are vested in the retirement plan and how their vesting status progresses over time. This is a full-length document from a firm's Form 5500 filings from the year [YYYY] that contains information about retirement saving contributions. The document to code will appear at the end of this prompt within < < < > > >.
Your objective is to summarize the vesting provisions of the plan, which were applicable to the plan year [YYYY] in tabular form. The goal will be to provide a simple way of consistently characterizing vesting features. Make sure to include information about whether the vesting schedule is available and the fraction of employee contributions that are vested in each year (e.g., 0% in the first year, 50% in the second year, etc.) for enrolled employees. We are only interested in information for the year [YYYY], not past years. I mention this because occasionally there will be updated information provided about vesting rules associated with past years in the document.
Here are a few additional details. Always include 8 columns, one for whether the vesting schedule is available and seven columns for the progression of vesting over time. If there is no information available or no mention of vesting, please report "NA". Now we will proceed with several examples.
EXAMPLES
Here are some relevant excerpts from other plans with similar textual content:
< <Input language for the year [YYYY1]: [EXAMPLE1]> >
Correct output: [TABLE1]
< <Input language for the year [YYYY2]: [EXAMPLE2]> >
Correct output: [TABLE2]
< <Input language for the year [YYYY3]: [EXAMPLE3]> >
Correct output: [TABLE3]
< <Input language for the year [YYYY4]: [EXAMPLE4]> >
Correct output: [TABLE4]
< <Input language for the year [YYYY5]: [EXAMPLE5]> >
Correct output: [TABLE5]
DOCUMENT TO CODE
< < <Input language for the year [YYYY]: [PLAN] > > >

References

Arnoud, A., T. Choukhmane, J. Colmenares, C. O'Dea, and A. Parvathaneni (2021). The Evolution of U.S. Firms' Retirement Plan Offerings. Evidence from a New Panel Data Set. Technical report, NBER.

Choukhmane, T., J. Colmenares, C. O'Dea, J. L. Rothbaum, and L. D. Schmidt (2024, August). Who benefits from retirement saving incentives in the u.s.? evidence on gaps in retirement wealth accumulation by race and parental income. Working Paper 32843, National Bureau of Economic Research.

Choukhmane, T., L. Goodman, and C. O'Dea (2025). Efficiency in Household Decision Making: Evidence from the Retirement Savings of US Couples. *American Economic Review*.

Gao, Y., Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang (2024). Retrieval-augmented generation for large language models: A survey.

Grattafiori, A., A. Dubey, A. Jauhri, et al. (2024). The llama 3 herd of models.

Lewis, P., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela (2021). Retrieval-augmented generation for knowledge-intensive nlp tasks.

Loshchilov, I. and F. Hutter (2019). Decoupled weight decay regularization.

Mikolov, T., W.-t. Yih, and G. Zweig (2013, June). Linguistic regularities in continuous space word representations. In L. Vanderwende, H. Daumé III, and K. Kirchhoff (Eds.), *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, pp. 746–751. Association for Computational Linguistics.

Reimers, N. and I. Gurevych (2019, 11). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

T. Rowe Price (2025, Q2). Reference point: T. Rowe Price defined contribution plan data. Annual report, T. Rowe Price Retirement Plan Services, Inc. Data as of December 31, 2024.

Vanguard (2023). How america saves 2023. Technical report.

Wang, W., F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou (2020). Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, pp. 38–45. Association for Computational Linguistics.