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

June 2026

PRELIMINARY

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 (2003-2023), yielding over one million plan-year observations: more than twentyfold the coverage of plans in the original hand-collected data. This expansion enables the first comprehensive population-level analysis of retirement plan design and illustrates how LLMs can make previously infeasible empirical research practical at scale. 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.

*Choukhmane: MIT Sloan School of Management, tahac@mit.edu; Dedyo: Harvard University, jdedyo@g.harvard.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.

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 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 (2003-2023), 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 land-

scape and how it has evolved over two decades. First, we show that matching formulas are highly concentrated: just three matching structures account for over 40% 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 formula requirements grew from approximately a fifth in 2004 to nearly a half by 2023, 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 45% to nearly 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. Section 4 concludes.

2 Methods

This section summarizes our procedure for producing our data. Section 2.1 describes how we gather retirement plan description text from PDF documents. Section 2.2 introduces the hand-collected data which are used as training examples for the LLM pipeline. 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 Text Extraction

Each observation in our ultimate data set is derived from a given plan's annual filing of Form 5500, the Annual Return/Report associated with its Employee Benefit Plan. These forms, which were introduced to satisfy reporting requirements related to the Employee Retirement Income Security Act (ERISA) of 1974 and the Internal Revenue Code, are publicly available through the Department of Labor. Form 5500 is filed for multiple types of benefit plans including health insurance and welfare benefit plans. Importantly, for any employer which

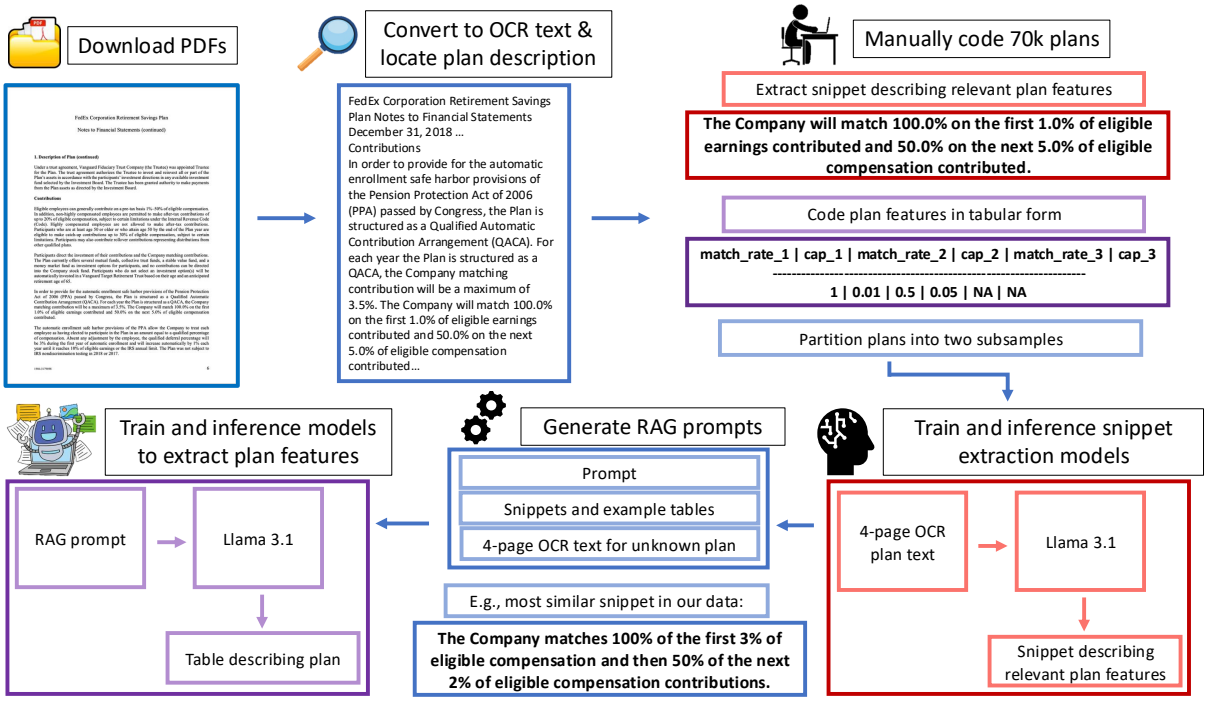


Figure 1: A flowchart of our data acquisition process.

sponsors a defined contribution retirement plan, its Form 5500 filing includes considerable detail about plan characteristics, many of which are already tabulated and available from CSV files on the DOL website (e.g., plan type, number of participants, and gross accounting flows). However, in addition to these standardized variables, all plans with more than 100 participants must also include a plain-English narrative summary of the plan, from which we use LLMs to extract a number of plan features described in the next section.

The first step of our process is to extract the text related to these summary plan descriptions 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 first page at which search terms for the plan description summary appear, then save a shortened PDF including that page and three subsequent pages.

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

We describe these two steps below.

2.1.1 Defining The Sample

Our population of interest is all plans which are either 401(k) or 403(b) and which have more than 100 participants.[1] We impose the latter restriction since most smaller plans do not have a requirement to file a description of the plan (though some choose to do so). There are 1,368,950 unique DC plans with any regulatory filing, covering 10,911,487 plan-years in total, of which there are 157,758 plans and 1,276,526 plan-years with more than 100 participants.

2.1.2 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. During PDF processing, some files were removed for computational errors including file corruption, file length, or OCR text extraction failure. Most issues with files not being found (green bars) occur prior to 2009, the year in which Form 5500 was required to be submitted electronically through the DOL's ERISA Filing Acceptance System (EFAST) system. In later years, the vast majority of filtered-out PDFs were rejected because we were unable to locate any of the keywords associated with the Summary Plan Description. Figure 2 tracks the sample size in each year for plans which have 100 or more participants. The PDF pipeline produces the final sample size of 1,149,649 plan-year observations (991,848 with more than 100 participants) comprising 146,704 (129,979 with more than 100 participants) distinct retirement plans potentially available to code.

[1] To satisfy the former restriction, we include only those plans that contain `2L` or `2J` in the `pension_benefit_code` field, which identifies them as defined contribution retirement plans.

[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".

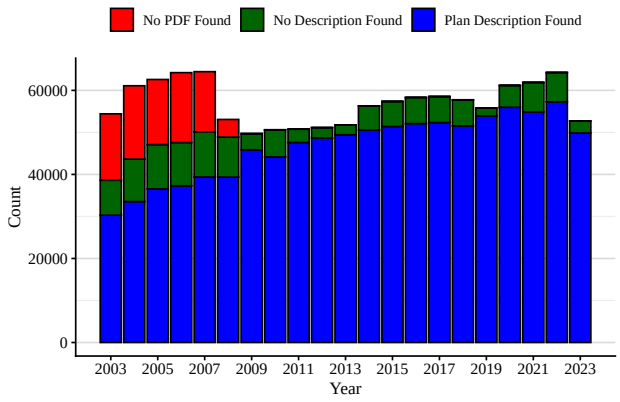


Figure 2: Sample size in each year for large plans.

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 2.1.2 (this keyword search was successful for 88.2% of large plans for which we were able to extract text from the PDF). '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 0.1% of large plan-years and would be invisible in the plot.

2.2 Hand-Collected Training Data

Next, we discuss the training data and the fields which we seek to extract from the summary plan description. Our build pipeline replicates the structured output which was hand-collected for Choukhmane et al. (2025).[3] That paper constructed a novel dataset of employer DC 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, which are described in a narrative format that varies substantially across documents. That dataset contains information for the largest approximately 4,700 plans in the US, as well as a random sample of approximately 1,500 smaller plans. In what follows, we describe the RA-extracted parameters, which the LLMs were also trained to extract.

For all plan features, RAs were instructed to make an initial determination as to whether a formula was '**simple**' or '**more complicated**' (we give examples below).

- **Simple** plans were expressed in straightforward percentage-based terms.

[3]See Arnoud et al. (2021) which summarizes results from an early version of the data, and Choukhmane et al. (2024) who use it in their analysis of distributional incidence of matching and tax benefits from the DC system.

- A plan is classified as '**more complicated**' if it is not simple. The most common conditions leading to this alternative classification are that it differed for different workers under the plan, it changed mid-year, and/or it described the plan feature in a non-standard way (e.g., dollar- instead of percentage-based matching contributions).

Approximately two-thirds of plans in the hand-collected dataset offered matching contributions classified as 'simple' and so amenable to further codification. The remaining third were 'more complicated', and the human research assistants did not tabulate any further details about matching rules. Our LLM pipeline follows the same approach in a much larger universe, seeking first to classify features as simple and, if so, to extract their details in a structured format.

Figure 3 gives an example of a simple match schedule, as well as the fields we extract. Match Rate 1 is the fraction of employee contributions the employer matches up to Cap 1, Match Rate 2 is the fraction of employee contributions from Cap 1 to Cap 2 matched, and Match Rate 3 is the fraction matched between Cap 2 and Cap 3.[4] In the case of Figure 3, the match schedule has only two tiers, so Match Rate 3 and Cap 3 are left empty. In the case of a more complicated match schedule, all six numeric fields in Figure 3 are left blank. An additional variable, `match_formula` is a boolean indicating whether the plan was classified as simple or more complicated, taking on the value 'Yes' or 'More complicated'.

The Company provided a safe harbor matching contribution equal to 100% of each Participant's Elective Deferrals not exceeding 3% of the Participant's Compensation, plus 50% of the next 2% of each Participant's Elective Deferrals.

	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: Fields collected for a *simple* retirement plan match schedule (left: text description; center: tabular; right: graphical).

For simple auto-features, we extract the fields given in Figure 4. Auto Enrollment Offered is a 'Yes'/'More complicated' boolean indicating whether the automatic enrollment provision of the plan has been classified as simple or more complicated. If auto-enrollment is simple,

[4]The LLM is trained to extract *marginal* caps rather than the cumulative caps we report throughout the paper, so the LLM data would code the plan in Figure 3 with `cap_2` $= 0.02$ rather than 0.05. Early testing showed improved LLM performance with marginal rather than cumulative caps, possibly since the marginal caps do not require the LLM to perform arithmetic in the case of plan language like that in Figure 3. In plotting, we transform the marginal caps reported by the LLM to cumulative by performing `cap_1`$' =$ `cap_1`, `cap_2`$' =$ `cap_2` $+$ `cap_1`$'$, and `cap_3`$' =$ `cap_2`$' +$ `cap_1`$'$.

then Initial Deferral gives the fraction of salary an employee automatically contributes to the plan upon enrollment (the field is left blank if auto-enrollment is classified as more complicated). Auto Increase Offered is another 'Yes'/'More complicated' boolean, this time indicating whether the automatic *increase* provision of the plan has been classified as simple or more complicated. In keeping, if the auto-increase is simple, then Auto Increase Amount gives the fraction of salary by which the employee salary deferral automatically increases each year, and Auto Increase Cap gives the fraction of salary deferred at which the automatic increase ceases. Both of these fields are left empty if the automatic increase is classified as more complicated.

Eligible participants are automatically enrolled in the Plan at 3% of their eligible compensation for each pay period unless a contrary election is made. The Plan will automatically increase elective contributions by 1% each plan year until the deferral rate reaches the plan-specified maximum of 6%.

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

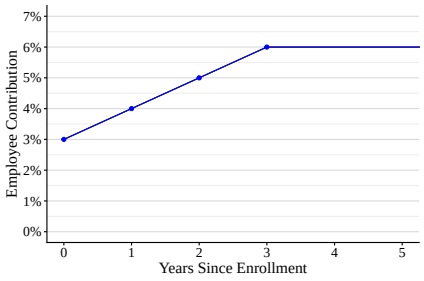


Figure 4: Fields collected for *simple* auto-features (left: text description; center: tabular; right: graphical).

Figure 5 gives an example of the fields we extract for a simple vesting schedule alongside the corresponding graphical depiction. Vesting Year 0 is the fraction of employer matching contributions vested upon enrollment in the plan (in this case, *NA* values represent zeros unambiguously). Vesting Year 1 is the fraction of such contributions vested after the first year enrolled in the plan. The remaining columns are defined analogously. Federal law requires employer contributions to vest after no more than six years of service, hence why Vesting Year 6 is the maximum that we collect. Employee contributions are always 100% vested, and we can safely ignore them in data collection. As always, the numeric Vesting Year fields are left blank in the case of a more complicated vesting schedule. As with matching, an additional variable, `vesting_offered` is a boolean indicating whether the plan was classified as simple or more complicated, taking on the value 'Yes' or 'More complicated'.

2.3 LLM Pipeline

Once we have extracted plaintext descriptions from the relevant subsets of the Form 5500 PDFs, we 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 for which the plain text

Participants vest in the Company's contributions at a rate of 20% per year of service after two years of service and become fully vested after six years of credited service.

	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

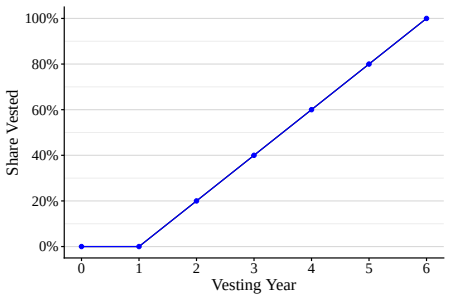


Figure 5: Fields collected for a *simple* retirement plan vesting schedule (left: text description; center: tabular; right: graphical).

summary plan descriptions are available from 2003–2023. For each plan feature, the pipeline proceeds as follows:

1. Partition the set of plans in our hand-collected dataset into two halves to enable independent double-coding of the OCR text universe.

2. Fine-tune 'snippet extractor' LLMs using the hand-collected training dataset, and use them to run inference on the other subset of the hand-collected dataset for use in subsequent prompts.

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 2.2) and collect plan features for simple plans.

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

   (a) Run inference using the snippet extractor LLMs on the universe of all OCR text gathered as described in Section 2.1.

   (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. Postprocess the LLM output to produce the final dataset.

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

To facilitate accuracy of fine-tuning, 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. We do so because 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 subsections 2.3.1 to 2.3.5, we explain each step of the process for tabulating information on employer matching contributions. Appendix Sections B.0.1 and B.0.2 provide analogous details and analytics for the collection of auto-features and vesting data.

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 plan-year 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 that were extracted from plan documents 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 tabular 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 contributions.

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[5] are remarkably similar to the RA-extracted snippets (which were copy-pasted by hand from company PDFs); see Figure 6. 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.

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 $g_2(\mathbb{1})$. These 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 (or subsequently tested) using any outputs of an in-sample LLM, making training examples highly realistic and our accuracy estimates on the test set applicable to true out-of-sample tasks. The tabulators are fine-tuned and inferenced using retrieval-augmented generation prompts, described in the following section.

2.3.3 Retrieval-Augmented Generation and Plan Feature Tabulation

The LLMs that code plan features in tabular format ('tabulators') are fine-tuned in the same manner as the 'snippet extractors' 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 7 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 evident that the held-out accuracy estimate within the hand collected sample provides a reasonable estimate of accuracy in the wild, since neither f_1 nor g_1 had any retirement plans from $\mathbb{2}$ in its train set. For matching, the LLMs generate identical output to the hand-collected data in the held-out test partition 84–86% of the time.[6] This likely provides a lower bound on accuracy of our final sample since there are clerical errors

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

[6] The comparable figure for auto-features is 94.5–95.5%, and for vesting is 74–75%. These closely mirror the between-LLM agreement rates in the wild for each of the features reported in Tables 1, 5 and 7.

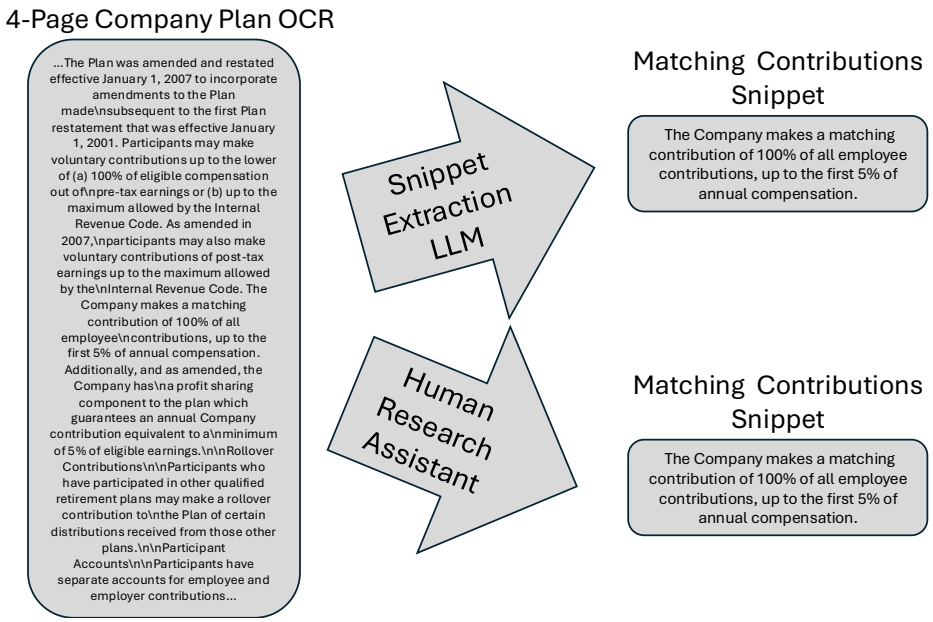
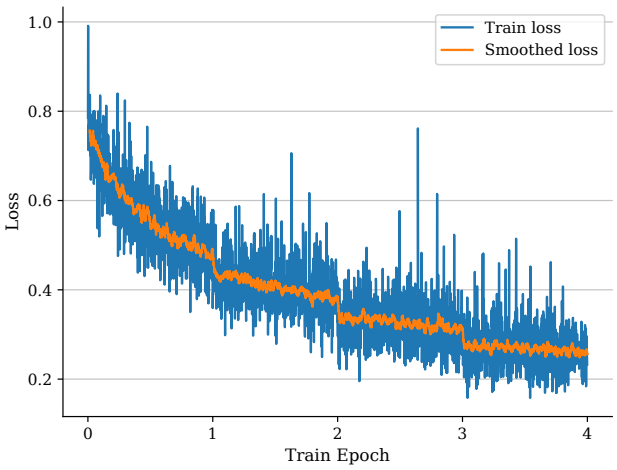


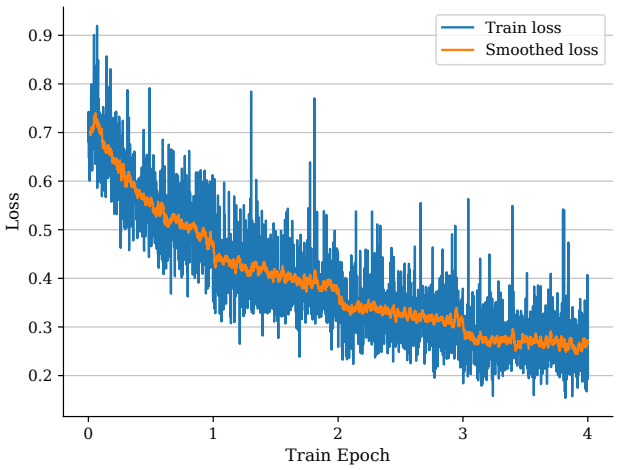
Figure 6: Demonstration of snippet extraction input and output using JetBlue's 2007 retirement plan. The hand-collected snippet is identical to the LLM-extracted snippet even though JetBlue was not a training example for the LLM.

in the hand-collected data (deflating the agreement rate in the test set) and we drop rows in the final sample where the LLMs do not produce identical output.

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.



(a) Training loss for the LLM trained on partition 𝟙.



(b) Training loss for the LLM trained on partition 𝟚.

Figure 7: 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.

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 variable, and a snippet is not considered for inclusion in the RAG prompt if a snippet with higher semantic similarity 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.

While we extracted a snippet from every plan, we include the full OCR text of the plan being analyzed in the RAG prompt. In principle we could have passed only the extracted text snippets to the prompts, but in practice we found that often they did not incorporate all relevant information, especially information necessary to confirm that the plan was simple. As a result, we passed the full OCR text while simultaneously providing the additional context from similarly phrased, previously human-tabulated plans. In initial experiments, we found that this allowed us to make use of the hand-collected extracted snippets while preserving accuracy related to confirming that a plan was simple.[7]

Another, seemingly simpler option would be to skip the snippet extraction approach entirely and only pass in the full OCR text. However, more epochs were required to lower training loss, which created a risk of overfitting. RAG prompting therefore acts as a key performance enhancer, exploiting the LLM's foundation knowledge to generalize effectively from a few relevant examples. 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.

[7]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.

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 nearly independent of idiosyncratic errors in the hand-collected data used to train the other pair.[8] This permits us to condition on whether the LLM outputs are identical for a given plan as a control against random LLM errors. We also set the temperature parameter to 0.01, which further reduces random variation.[9]

Table 1 shows that, for the double-coded out-of-sample plan-year observations before applying the agreement filter, the LLMs agreed on the distinction between 'complicated' and 'simple' in almost 91% of cases, and produced identical output describing the match schedule in 88.3% of cases. Given that the LLMs agree that the plan is simple, they produce identical tables for 96.3% of plans. Corresponding analytics for auto-features and vesting schedules are reported in Sections B.0.1 and B.0.2.

Table 1: Match schedule agreement proportions in out-of-sample universe for $f_1(x; g_1(x))$ and $f_2(x; g_2(x))$.

Criterion	Agreement Rate
All formulas	
Distinction between simple and complicated	90.9%
Distinction and numeric values	88.3%
Among formulas both LLMs classify as simple	
Exact numeric terms	96.3%

Notes: All formulas calculates agreement proportions using the entire codified universe as the denominator. 'Distinction between simple and complicated' indicates whether both models classified the formula as 'simple' or 'more complicated'. 'Distinction and numeric values' indicated whether both models classifies the formula as 'simple' or 'more complicated' and produced identical numeric values. *Among formulas both LLMs classify as simple* calculates agreement using the plans that both LLMs classified as 'simple' as the denominator. 'Exact numeric terms' indicates whether the LLMs produced identical numeric values for these simple plans.

[8] Strictly speaking, the training inputs for each tabulator *are* influenced by a snippet extractor trained on the opposite partition. For example, since f_1 is trained with $g_2(\mathbb{1})$, partition $\mathbb{2}$ *does* affect the snippet generated for each plan-year in $\mathbb{1}$ through its effect on the training of $g_2(\cdot)$.

[9] 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.

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. One of the key advantages of fine-tuning is to standardize the style of the final output across LLM queries, which significantly simplifies this process. The extracted text is compared to expected characteristics[10] and flagged if there are any errors. Since each plan is double-coded, the error-free LLM outputs are then compared to each other. If the LLMs produce identical output,[11] 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. Figure 18 in Appendix A reports the share dropped for LLM disagreement for each of the three plan features in each year.

Finally, there is one special case worth mentioning. This is if there is no mention of a 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

To benchmark the LLM output against skilled human coders, we conducted an audit using three research assistants trained to perform the coding task by hand. For a subsample of plans, the auditors were provided the exact plan text supplied to the LLM and asked to fill in a spreadsheet with the relevant plan features. Each plan was independently coded by at least two auditors; where all auditors agreed, that coding was taken as ground truth, and any disagreements were adjudicated by one of the coauthors reading the full plan text. Results are reported in Table 2, with separate panels for matching, auto-features, and vesting.

The audit sample is constructed as follows. We began by drawing 502 plan-year observations for each of matching and auto-features, selected to be representative on calendar year and classification as 'simple' or 'more complicated'. Of these, 24 were not present in our

[10] 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).

[11] "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.

Table 2: Comparison of LLM output to trained RA output on a subset of plans.

Column	N	Agree	Agreement Rate
Panel A: Matching			
Match Formula	439	427	97.3%
Given simple match formula:			
Match Rate 1	313	307	98.1%
Cap 1	313	306	97.8%
Match Rate 2	313	311	99.4%
Cap 2	313	310	99.0%
Match Rate 3	313	313	100.0%
Cap 3	313	313	100.0%
Panel B: Auto-features			
Auto-Enrollment Offered	450	445	98.9%
Auto-Increase Offered	450	442	98.2%
Given simple auto-enrollment:			
Initial Deferral	445	445	100.0%
Given simple auto-increase:			
Auto-Increase Amount	442	442	100.0%
Auto-Increase Cap	442	442	100.0%
Panel C: Vesting			
Vesting Offered	499	475	95.2%
Given simple vesting:			
Vesting Year 0	368	351	95.4%
Vesting Year 1	368	347	94.3%
Vesting Year 2	368	346	94.0%
Vesting Year 3	368	345	93.8%
Vesting Year 4	368	345	93.8%
Vesting Year 5	368	346	94.0%
Vesting Year 6	368	349	94.8%

Notes: Each row reports the number of plans coded by both the RA and the LLM (N), the number on which they agree (Agree), and the agreement rate. Conditional panels restrict to plans where the classification field took the indicated value. For plans not categorized as simple, all numeric columns are left empty, mechanically resulting in perfect agreement in numeric fields (hence we do not report them).

final dataset and are dropped, leaving an audit sample of 478.[12] We then restrict to plans

[12]These are almost always cases in which the same plan filed more than once for a given year, and our

on which the two independently trained LLM pipelines produced identical output, since our final dataset retains only double-coded agreements. This removes 39 observations for matching and 28 for auto-features, yielding final audit samples of 439 and 450, respectively. The vesting audit was carried out last, after we had our final sample, and so we sampled 499 plans, also representative on year and classification, each of which was in the final data. The number of plans entering each panel of Table 2 reflects these counts; within-panel N's fall further where the conditional rows restrict to plans the LLM classified as simple.

The audit indicates that the LLM achieves near human-level performance. For matching, the LLM and auditors agreed on the match formula classification in 97.3% of cases (427 of 439), and among plans both classified as simple, agreement on the numeric match rates and caps ranged from 97.8% to 100%. For auto-features, auto-enrollment and auto-increase classifications agreed in 98.9% and 98.2% of cases respectively, with perfect agreement on the conditional numeric fields (initial deferral, increase amount, and increase cap). For vesting, the LLM and auditors agreed on whether vesting was offered in 95.2% of cases, and on the year-by-year vesting fractions in roughly 94% of cases.

3 The Evolving Landscape of DC Retirement Plans

3.1 Matching

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

The distribution of employer matching formulas. Figure 8 illustrates the vast heterogeneity that exists in employer match schedules. Shading these match schedules by prevalence, however, reveals some measure of concentration in certain plan structures.

Table 3 gives the ten most common matching formulas in our dataset and quantifies the aforementioned concentration—there is a significant overrepresentation of just a few match schedules. 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 (17% of plans). The next most common arrangements are a 50% match up to 6% of salary (16.1%) and a 100% match up to 4% (8.9%). These three formulas alone account for over 40% of all plans offering employer contributions, suggesting that plan sponsors gravitate toward a small set of conventional matching structures.

build retained an alternative filing.

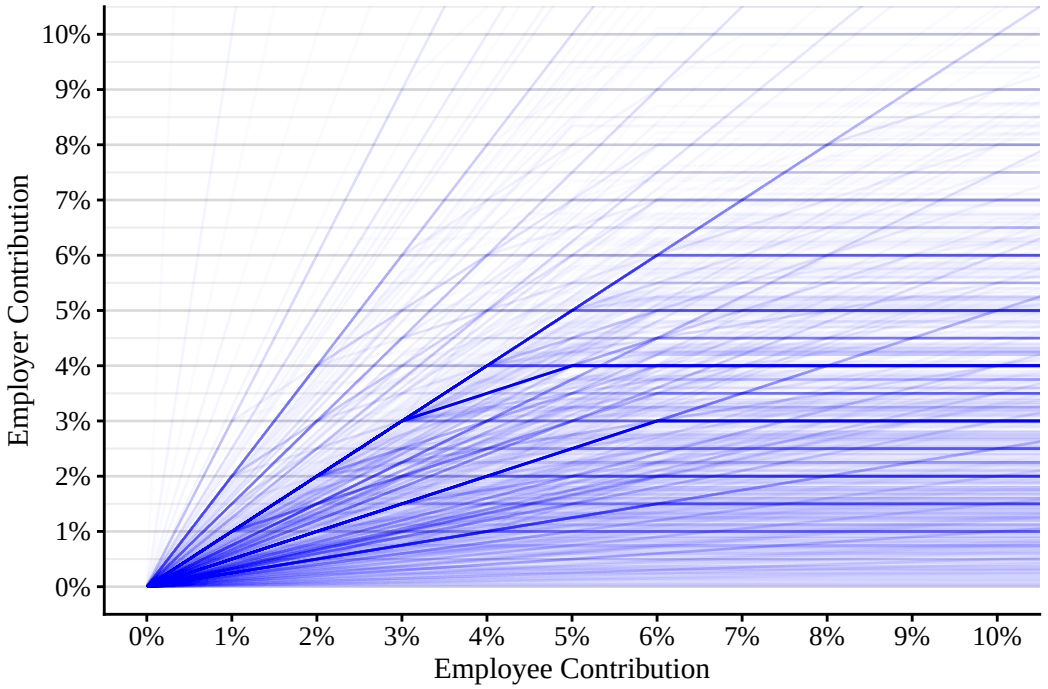


Figure 8: All nonzero match schedules in our universe shaded by prevalence.

Figure 9 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 (32% of plans), followed closely by 6% (31%) and 4% (19%). 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 62% of plans offering dollar-for-dollar matching and 27% 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.

Fact 1: Employer matches have become substantially more generous. Figure 10 tracks three measures of plan generosity from 2003 through 2023, showing raw means alongside a composition-adjusted profile.[13] 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 1.8% of salary in 2003 to 2.8% in 2023—a 56%

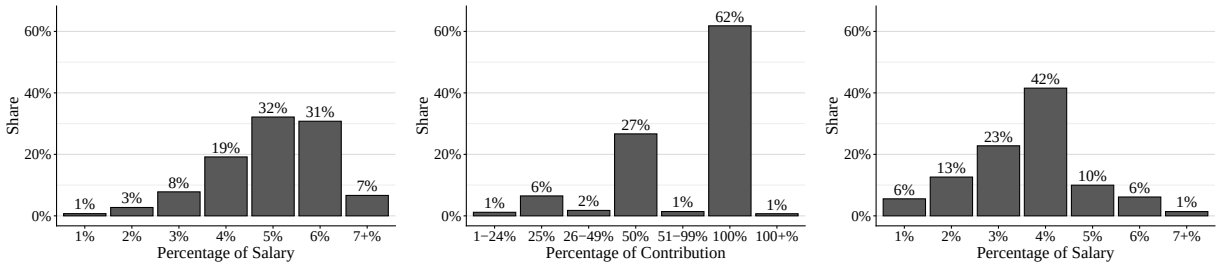
[13]The composition-adjusted profile plots the year coefficients from the fixed effects regression $y_{it} = \hat{\alpha}_i + \sum_s \hat{\beta}_s \text{Year}_s + \varepsilon_{it}$, where we set 2003 as the base year and include the average fixed effect of plans observed in that year. This approach isolates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

Table 3: Top ten most common plans.

Plan					
Match Rate 1	Cap 1	Match Rate 2	Cap 2	Proportion	Safe Harbor Match
100	3	50	5	17.0%	Yes
50	6	0	0	16.1%	No
100	4	0	0	8.9%	Yes
50	4	0	0	5.8%	No
100	3	0	0	4.6%	No
100	5	0	0	4.5%	Yes
25	6	0	0	4.4%	No
100	6	0	0	4.1%	Yes
25	4	0	0	3.4%	No
50	5	0	0	2.9%	No

Notes: This table presents the ten most common match schedules in our data. Match Rate 3 and Cap 3 are zero for all ten of these plans, so they are not reported. When calculating the proportions, we restrict our sample to plans with a match that we could encode in tabular format, which offers a nonzero contribution. The 'Plan' column follows the format of Figure 3, so the first row 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.

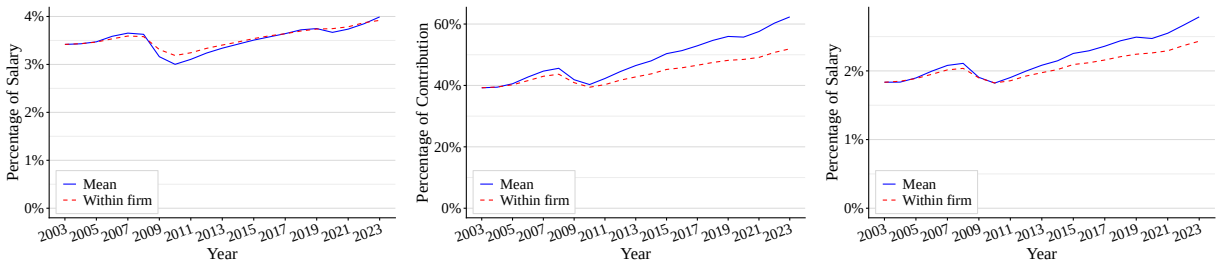
increase—and the median jumped from 1.5% to 3.0%. The composition-adjusted measures increase by less, suggesting that a substantial portion of the increase in generosity arises due to new plans entering rather than existing plans changing their formulas.



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

Figure 9: 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. The sample is all those plans we could codify which offer a nonzero matching contribution. 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.

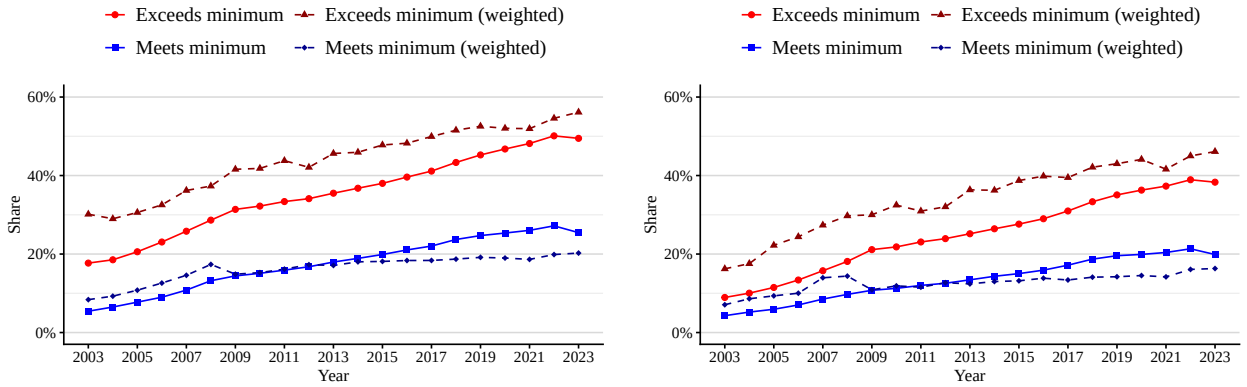


(a) Matching cap. (b) Match rate on first dollar. (c) Maximum employer match.

Figure 10: 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 $y_{it} = \hat{\alpha}_i + \sum_s \hat{\beta}_s \text{Year}_s + \varepsilon_{it}$. 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 11 documents the rise of safe harbor: the share of matching plans meeting or exceeding safe harbor matching requirements grew from approximately 18% in 2003 to nearly 50% by 2023, with most of this increase representing plans that exactly meet minimum safe harbor thresholds rather than exceeding them (Figure 11a). The share of workers covered by matching plans meeting or exceeding safe harbor requirements grew from roughly 30% in 2003 to 56% in 2023. Figure 11b plots the same series, but includes requirements on auto-enrollment and vesting for safe harbor status, reducing raw proportions yet maintaining the upward trend. Safe Harbor status offers employers relief from nondiscrimination testing, creating a strong regulatory incentive that appears to be reshaping plan design nationwide.



(a) Share of plans satisfying safe harbor matching requirements.

(b) Share of plans satisfying safe harbor matching, auto-enrollment, and vesting requirements.

Figure 11: 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. 'Meets minimum requirements' is defined as exactly following one of the safe harbor specifications (Basic or QACA). 'Exceeds minimum requirements' is defined as offering a match more generous than the minimum requirements, while still satisfying the regulatory definition of safe harbor. The weighted series calculate a weighted mean for each year, weighted by the number of plan participants.

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 1,600 distinct matching formulas observed in our data, plan design is converging toward a small number of dominant options. The top 5 formulas accounted for 42% of all plans in 2003 but 60% by 2023. The Herfindahl-Hirschman Index (HHI) rose by 48% over the same period, from 0.071 to 0.105 (Figure 12). Importantly, the number of distinct formulas in

use has remained roughly stable (around 520–580 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 9—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.

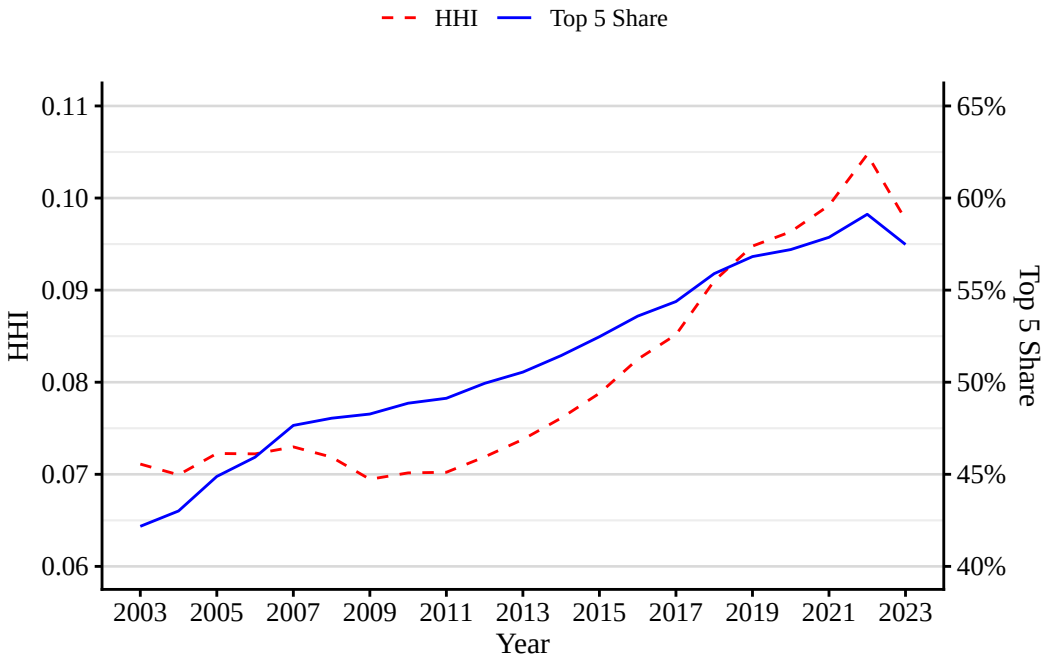
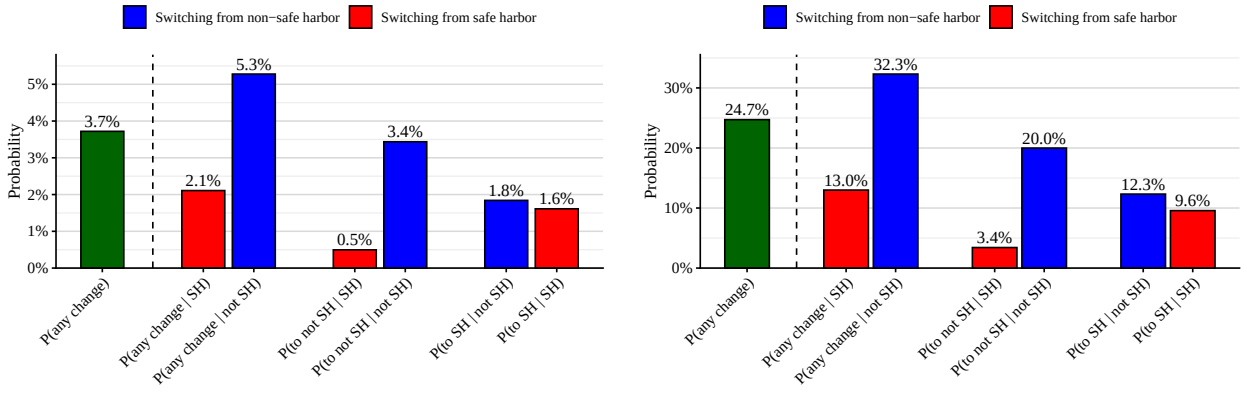


Figure 12: Concentration of matching formulas over time.

Notes: Sample restricted to those plans that we can codify that offer a nonzero matching contribution. The HHI in year t is calculated as $\text{HHI}_t = \sum_i \left(\frac{\text{Count}_{it}}{\sum_i \text{Count}_{it}} \right)^2$ where i indexes unique plan structures. The HHI (dashed red, left axis) and the cumulative market share of the top 5 formulas (solid blue, right axis) both trend upward steadily from 2003 to 2023, 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. Figure 13a reveals that in 2022 only about 3.7% of plans changed their matching formula. Comparing the eight year period from 2015 to 2023, roughly 25% of plans had a different formula by the end of the period, and 75% of plans in 2023 are still using the same formula they started with in 2015 or earlier. This

inertia helps explain the composition-adjusted finding from Figure 10: 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.[14]



(a) Probability of plan change in 2022–2023.

(b) Probability of plan change in 2015–2023.

Figure 13: Probability of changing formula conditional on Safe Harbor status.

Notes: Sample restricted to those plans that we can codify and offer a nonzero matching contribution. A plan is classified as safe harbor if it satisfies the regulatory requirements on matching for safe harbor status (regardless of auto-enrollment and vesting). Plans can be more generous than the minimum safe harbor matching requirements and maintain safe harbor status. Conditional probabilities calculated as, e.g., $\mathbb{P}(\text{to SH} \mid \text{not SH}) = \frac{\sum \text{Changed to SH in final year from non-SH in base year}}{\sum \text{non-SH in base year}}$. Safe Harbor plans (red) are far less likely to change than non–Safe Harbor plans (blue), and non–Safe Harbor plans are far likelier to change to a Safe Harbor formula than vice-versa.

Safe Harbor formulas are especially persistent. The probability of a formula change from 2022–2023 is roughly halved for safe harbor matching formulas, which had a 2.1% probability of changing in the period; non-safe harbor formulas had a 5.3% probability of changing between 2022 and 2023—about 40% higher than the aggregate change probability of 3.7% (Figure 13a). Breaking down these probabilities further, a safe harbor plan had a 0.5% chance of exiting safe harbor status by changing to a non-safe harbor match formula, whereas a non-safe harbor plan had roughly a 1.8% chance of changing to a safe harbor match formula to enter safe harbor status. That is, a non-safe harbor plan was approximately 3.6 times likelier to enter safe harbor status than a safe harbor plan was to exit, suggesting that safe harbor acts as an absorbing state.

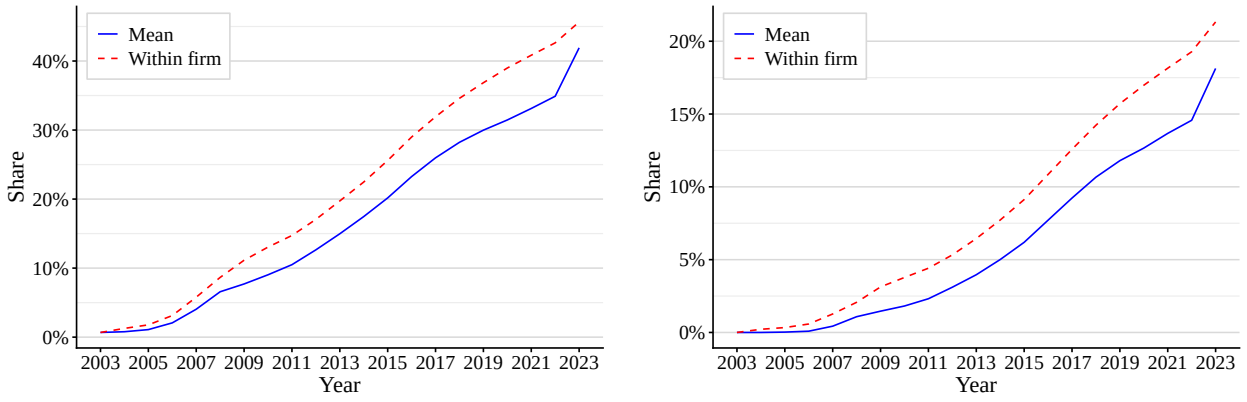
Indeed, over the eight years in Figure 13b, safe harbor plans had a 13% chance of changing their formula compared to a 32.3% probability of a formula change for non-safe harbor plans,

[14] A useful plausibility check is that the one-year probabilities roughly recover the corresponding eight year probabilities. For example, $\mathbb{P}(\text{any change from 2022–2023}) = 0.037$, so $\mathbb{P}(\text{no change from 2022–2023}) = 0.963$, and therefore roughly speaking, $\mathbb{P}(\text{any change from 2015–2023}) = 1 - 0.963^{2023-2015} \approx 26\%$. This closely matches the 24.7% estimate reported in Figure 13b.

making non-safe harbor plans more than twice as unstable. (Figure 13b). A plan that satisfied safe harbor in 2015 had a 3.4% probability of exiting safe harbor by 2023, while a non-safe harbor plan in 2015 had a 12.3% chance of entering safe harbor by 2023. The regulatory benefits of Safe Harbor status appear to have two effects. They create a strong lock-in effect: once a plan qualifies for the nondiscrimination testing exemption, there is little incentive to redesign. Moreover, they have a gravitational effect: plans are far likelier to enter safe harbor than they are to exit. Combined with the rapid adoption documented in Figure 11, this stickiness implies that the Safe Harbor formula's dominance is likely to persist and deepen in the years ahead.

3.2 Auto-enrollment

Figure 14 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 plausibly 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) Share offering auto-enrollment over time.

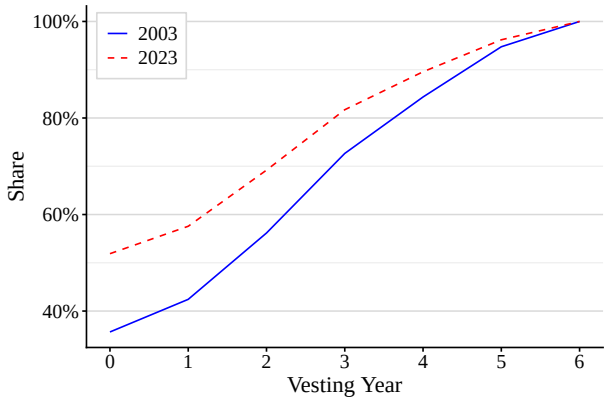
(b) Share offering auto-escalation over time.

Figure 14: Presence of auto-features over time.

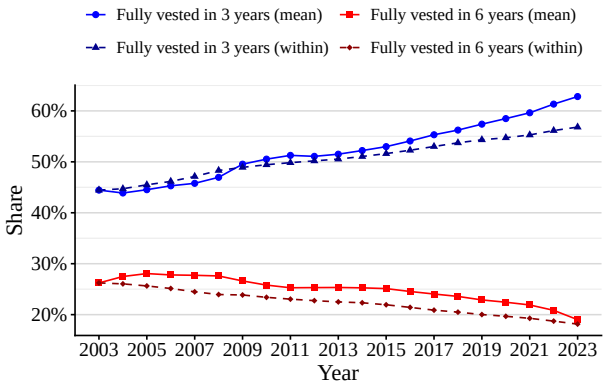
Notes: This figure summarizes the presence of auto-features for each year in our data. We do not do any filtering before calculating the proportions for the 'mean' line. The 'within firm' line plots the year coefficients in the fixed effects regression $y_{it} = \hat{\alpha}_i + \sum_s \hat{\beta}_s \text{Year}_s + \varepsilon_{it}$. 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 15 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 45% in 2003 to nearly 65% by 2023, while the share requiring six years to fully vest declined from roughly 25% 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 no more than 3 years.

Figure 15: Evolution of vesting.

Notes: Each of the 'within-firm' lines plots the year coefficients in the fixed effects regression $y_{it} = \hat{\alpha}_i + \sum_s \hat{\beta}_s \text{Year}_s + \varepsilon_{it}$. 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. 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.

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.

A Extra Exhibits

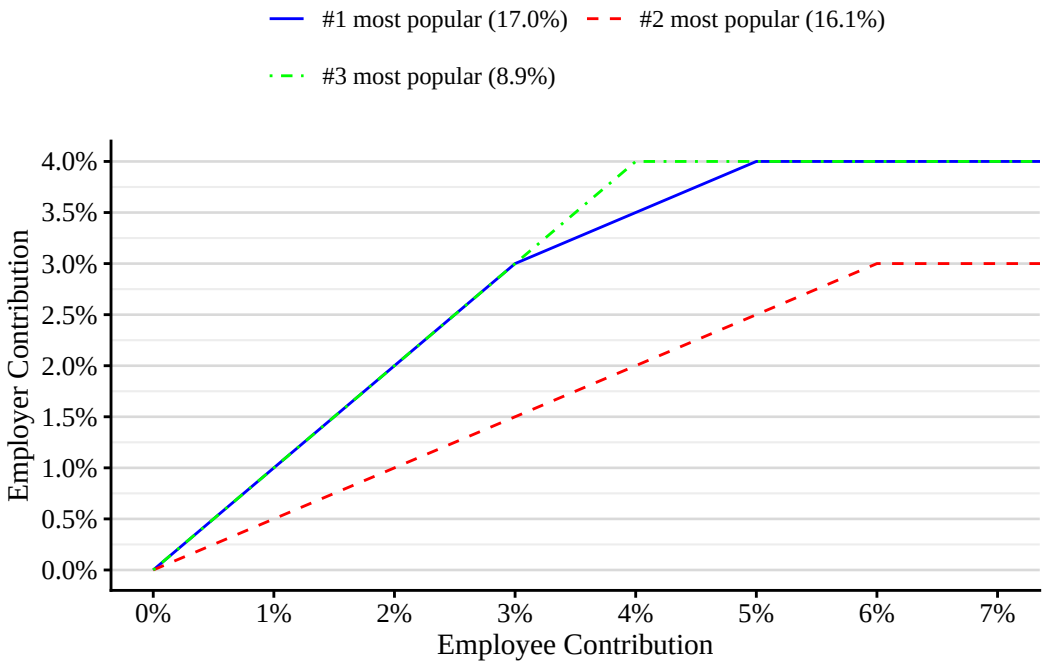
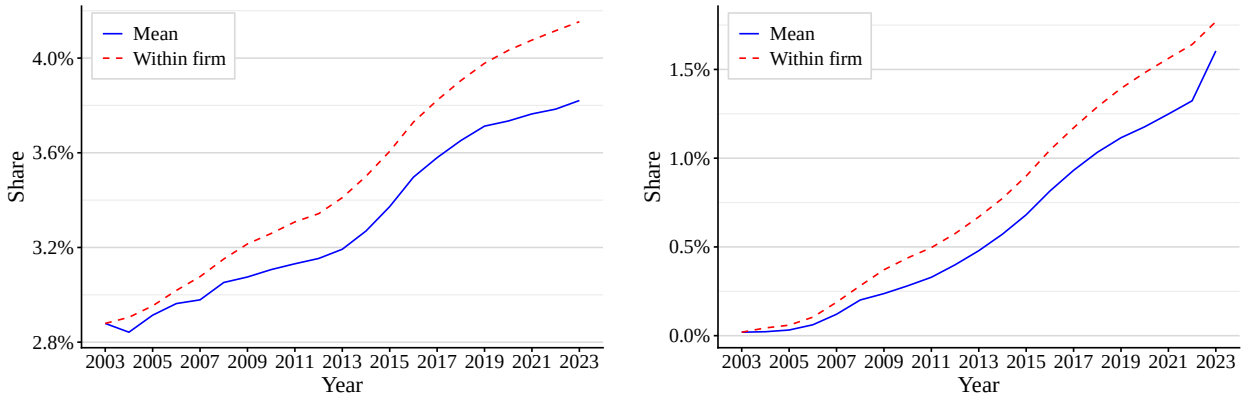


Figure 16: Top three most common match schedules.

Notes: This figure presents the three most common match schedules in our data. The proportions differ from those in Table 3 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 17: Automatic enrollment initial deferral rates over time.

Notes: This figure summarizes the average initial deferral rates 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 $y_{it} = \hat{\alpha}_i + \sum_s \hat{\beta}_s \text{Year}_s + \varepsilon_{it}$. 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.

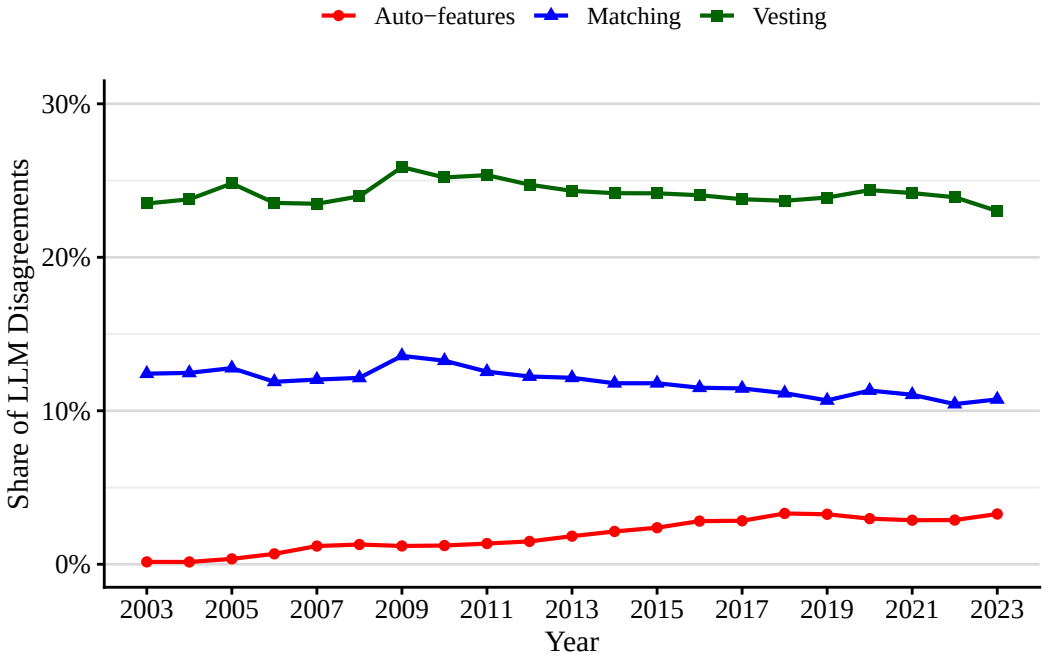


Figure 18: Proportion of the sample dropped due to LLM disagreement.

Notes: This figure presents the proportion of the sample for which OCR text extraction was successful that is dropped due to LLM disagreement at inference.

B LLM Pipeline for Other Plan Features

B.0.1 Auto-features

The LLM output for a plan that has simple auto-enrollment and auto-escalation is presented in Figure 4. 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 Figure 4. 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.

Table 4: Share of auto-features classified as more complicated vs. simple.

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

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 directionally similar to those in the hand-collected data, in which 98% of plans were adjudicated to be 'simple', though about twenty times less likely to be more complicated.

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 the auto-enrollment and auto-escalation provisions in 97.9% of cases (Table 5). Given that the LLMs agree that the plan is simple, they produce identical tables for 98.7% of plans. Table 4 breaks down the proportion of plans that the LLMs could encode in tabular format for 2004, 2023, and across the entire sample.

Table 5: Auto-features agreement proportions in out-of-sample universe for $f_1(x; g_1(x))$ and $f_2(x; g_2(x))$.

Criterion	Agreement Rate
All formulas	
Distinction between simple and complicated	99.3%
Distinction and numeric values	97.9%
Among formulas both LLMs classify as simple	
Exact numeric terms	98.7%

Notes: All formulas calculates agreement proportions using the entire codified universe as the denominator. For the purposes of Table 5, plans are classified as simple only if both autoenrollment and auto-increase are classified as simple—if either is classified as more complicated, the entire plan is classified as such. 'Distinction between simple and complicated' indicates whether both models classified the formula as 'simple' or 'more complicated'. 'Distinction and numeric values' indicates whether both models classified the formula as 'simple' or 'more complicated' and produced identical numeric values. *Among formulas both LLMs classify as simple* calculates agreement using the plans that both LLMs classified as 'simple' as the denominator. 'Exact numeric terms' indicates whether the LLMs produced identical numeric values for these simple plans.

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

The LLM output for a plan that offers a simple vesting schedule is presented in Figure 5. 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 Figure 5. 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 (Table 7). Given that the LLMs agree that the plan is simple, they produce identical tables for 81.9% of plans. Table 6 breaks down the proportion of plans that the LLMs could encode in tabular format for 2004, 2023, and across the entire sample.

Table 6: Share of vesting schedules classified as more complicated vs. simple.

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

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'.)

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

Criterion	Agreement Rate
All formulas	
Distinction between simple and complicated	88.8%
Distinction and numeric values	75.8%
Among formulas both LLMs classify as simple	
Exact numeric terms	81.9%

Notes: All formulas calculates agreement proportions using the entire codified universe as the denominator. 'Distinction between simple and complicated' indicates whether both models classified the formula as 'simple' or 'more complicated'. 'Distinction and numeric values' indicates whether both models classified the formula as 'simple' or 'more complicated' and produced identical numeric values. *Among formulas both LLMs classify as simple* calculates agreement using the plans that both LLMs classified as 'simple' as the denominator. 'Exact numeric terms' indicates whether the LLMs produced identical numeric values for these simple plans.

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 employer 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] > > >