RESEARCH-ARTICLE

Transforming Unstructured Sensitive Information into Structured Knowledge

BRAULIO C BLANCO LAMBRUSCHINI, University of Luxembourg, Esch-sur-Alzette, Luxembourg

MATS BRORSSON, University of Luxembourg, Esch-sur-Alzette, Luxembourg

Open Access Support provided by:

University of Luxembourg

Published: 14 November 2024

ICAIF '24: 5th ACM International Conference on AI in Finance
November 14 - 17, 2024
NY, Brooklyn, USA

ICAIF '24: Proceedings of the 5th ACM International Conference on AI in Finance (November 2024)
https://doi.org/10.1145/3677052.3698602
ISBN: 9798400710810

Transforming Unstructured Sensitive Information into Structured Knowledge

Braulio C. Blanco Lambruschini
SNT - SEDAN, University of Luxembourg
LU
braulio.blanco@uni.lu

Mats Brorsson
SNT - SEDAN, University of Luxembourg
LU
mats.brorsson@uni.lu

Abstract

Information is crucial in today's context, yet less than 20% of companies utilize their unstructured data due to its complexity. Information Extraction (IE) is vital for effective data use, but current IE models face four major issues. First, they often provide limited information, such as a simple entity-attribute relation. Second, they struggle with multiple languages. Models like GPT, Mistral, and Llama3 show promise but face a third issue: output reliability due to hallucinations. Fourth, there is a challenge in reducing sensitive data leakage after fine-tuning models.

This study introduces an enhanced approach for fine-tuning GPT-based models, designed to extract and assess information involving multiple entities and attributes, performing both multientity extraction (MEE) and multirelation extraction (MRE), and presenting results in a JSON format. Our methodology evaluates the impact of using synthetic data for fine-tuning to ensure reliable outcomes.

Applied to legal documents from the Luxembourg Business Registers (LBR), our findings show that replacing sensitive data with synthetic data significantly improves the fine-tuning of Llama3-based models, though not for Mistral-based models. Our top models outperform Mistral in various scenarios, requiring only 500 samples for fine-tuning and running efficiently on modest servers. This approach is suitable for multilingual Information Extraction in any domain.

CCS Concepts

• **Computing methodologies** → **Natural language processing**; **Natural language generation**; **Information extraction**.

Keywords

LLM, Finance, Information Extraction

ACM Reference Format:
Braulio C. Blanco Lambruschini and Mats Brorsson. 2024. Transforming Unstructured Sensitive Information into Structured Knowledge. In *5th ACM International Conference on AI in Finance (ICAIF '24), November 14–17, 2024, Brooklyn, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3677052.3698602

ICAIF '24, November 14–17, 2024, Brooklyn, NY, USA

ACM ISBN 979-8-4007-1081-0/24/11
https://doi.org/10.1145/3677052.3698602

1 Introduction

Despite the critical importance of business information, only 18% of companies utilize their unstructured data, which constitute about 80-90% of their total amount of data [5, 7, 9].

In order to use unstructured data, you need to extract useful information from it. Most of the methods used for *information extraction* fall in one of two main categories: *content-based* or *context-based*. Content-based methods usually use regular expressions to search for predictable patterns [10, 12, 15, 16]. Context-based methods, on the other hand, use modern technologies such as machine learning and deep learning [11, 13, 17, 19]. These methods are more adaptable to unknown rules and scenarios than content-based methods, and advances in generative techniques such as GPT have improved the task of information extraction to a great extent. Large Language Models (LLMs) such as ChatGPT[1], Bing Chat[2], and Claude GPT [3] are now common tools for various tasks, including information extraction. However, their use in business is limited by the high costs of processing large volumes of documents and the significant computing power required, necessitating resource-intensive cloud setups.

Cloud services provide scalability, cost efficiency, and enhanced accessibility, allowing businesses to quickly adapt to changing demands. However, data leakage remains a significant concern. In March 2024, Forbes reported a cyber attack on cloud-based AI platforms affecting various organizations [8]. Additionally, high costs for processing large document volumes and substantial computing power needs often necessitate resource-intensive cloud setups.

Security and cost reduction drive IT projects. While cloud solutions offer robust security, their data aggregation attracts hackers. Thus, finding affordable, effective solutions that integrate seamlessly into organizational infrastructure is crucial.

This paper presents a compact GPT-based model for deployment on small servers, designed to extract sensitive information from limited labeled samples while handling untrained languages and unexpected document formats. Figure 1 shows our evaluation setup. We compared our model's performance with ChatGPT3.5, Claude3 Haiku, and nine locally deployable models, none of which were fine-tuned. ChatGPT3.5 was the most effective cloud-based solution, while Llama3 outperformed the other local models.

Our findings indicate that fine-tuning the LLaMA-3 and Mistral-based models with real data yields significantly improved performance. However, for LLaMA-3 models, substituting some sensitive data with semi-synthetic data achieves results comparable to those of using only real data, while mitigating the risk of data leakage

[1] https://openai.com/chatgpt
[2] https://www.bing.com/chat
[3] https://claude.ai

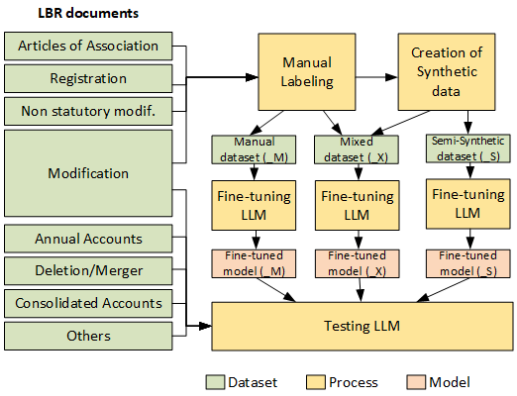

Figure 1: An overview of the evaluation process. We compare the effectiveness of information extraction using i) only labeled data, only semi-synthetic data and a mix of both.

associated with training on sensitive information. Additionally, our trained model is small enough to be deployed on a local server.

In summary, our primary contributions are:

(1) presenting a comprehensive set of objective metrics for evaluating LLM performance for Information Extraction tasks,
(2) introducing a compact free model for extracting sensitive information in a language-agnostic JSON format,
(3) proposing a methodology for generating semi-synthetic data for sensitive datasets, and
(4) analyzing the impact of using semi-synthetic data alone versus combining it with manually labeled data for fine-tuning LLMs.

Our proposed methodology and recommendations are ideal for situations where fine-tuning a large language model (LLM) is necessary, particularly when data sensitivity during training is a major concern.

2 Related work

Historically, information extraction has been extensively studied and several methods have been used to approach the problem [1–4, 21]. These methods have multiple shortcomings. Zaman et al. [21] described the following common issues: First, the solutions are not generalized for all domains; each approach is highly domain-specific. Second, they are, in general, not able to deal with the overwhelming diversity in the type and nature of documents, with hundreds and thousands of documents in different formats. Third, the methods are limited to specific types of documents due to the noise generated during digitization. Fourth, ambiguities in NLP models arise from text written in Natural Language. Fifth, different languages can be found even in the same documents, but most of the current models are language-specific and cannot handle that situation.

The Open Information Extraction (OpenIE) annotator [6, 18], is a tool for Information Extraction developed by the Stanford Natural Language Processing Group, that extracts information into relation triplets that use a logistic regression classifier. OpenIE breaks short input sentences into smaller ones to find the relationship between entities. In 2022, Yu et al. [20] proposed DragonIE as an improved OpenIE. The authors used a directed acyclic graph to generalize the model outperforming OpenIE tasks both in- and out-of-domain.

Qu et al. [14] proposed the extraction of triplets from entities based on a BiLSTM network + Graph Convolutional Networks (GCN) together with a parallel BiLSTM network + Convolutional Neural Network (CNN), combined with a softmax classifier. Examples of triplets given by the authors for this study are *born_in(John, Count Kerry)* and *educated_in(John, Saint Columbas College)*. As shown in the research, triplets are limited to linking a certain number of pairs from a small input text (50 words/tokens).

Exsense [9] extracts sensitive information from unstructured data using both context- and content-based mechanisms in a single architecture. It uses regular expressions (e-mail, addresses, securities, etc.) and a BERT-biLSTM-attention network to label other sensitive information in four categories: personal information, networking information, secrets, and credential information. However, Exsense is limited to English and is highly sensitive to OCR errors, which can result in mismatches with regular expressions. In our case, we define sensitive date as personal information such as names, addresses, birthday information and so on, likewise company information, costs and transactions and information about relationships between companies or a company and people.

Adnan and Akbar [3], present the thesis that the main limitations in the information extraction of unstructured data are: i) the lack of a defined schema, ii) the high variety of multiple formats, iii) lack of standardization, iv) the inherent noise of the text, and v) the limited availability of multilingual models.

Training GPT-3+-based Large Language Models (LLM) requires substantial processing resources and GPU memory. For instance, the Llama3 8B model, with 8 billion parameters and a space in disk of 5.5 GB, demands around 22 GB of GPU RAM for fine-tuning due to additional memory needs for gradients and optimizer state. Parameter-Efficient Fine-Tuning (PEFT) with Quantized Low-Rank Adapters (QLORA) helps reduce this memory consumption. PEFT, through techniques like Low-Rank Adapters (LoRA), stores gradients and activations in smaller matrices during fine-tuning, while QLORA further reduces memory use by employing 16-bit, 8-bit or even 4-bit quantization instead of 32-bit precision. This approach enables fine-tuning of a 5.5 GB model on a GPU with just 16 GB of RAM.

With the current technological advances, we tested the latest available OpenAI model, GPT-4o-mini, using the API[4]. The results are promising and comparable to those of GPT-3.5. While the model is more compact, it remains a paid service. Even using the JSON output capability the other problems remains.

In summary, most of the current tools are limited by language, structure, domain or triplet-based relations.Other models that can deal with those limitations, can be or expensive or not compact to be deployed in a small server.

3 Dataset

We obtained our dataset from a set of diverse documents of the Luxembourg Business Register (LBR), which are publicly available

[4]https://platform.openai.com/docs/api-reference/making-requests

for download[5]. For the current research, we split the set of documents into two distinct groups: one for fine-tuning and validation, and the other for testing. See figure 1.

As shown in Table 1, the first dataset group consists of documents in French, German, and English that we already preprocess and extracted the text. We manually labeled 1,000 pages from 145 documents. Reading the text from each page, we extract the sensitive information in JSON format. The process consisted in querying the extracted text and together with the original PDF, we create the desired JSON output.

Table 1: Distribution of labeled pages per language.

Language	# Pages	Avg. # words
French	682 (68.2%)	233
German	142 (14.2%)	237
English	176 (17.6%)	306
Total	1,000 (100%)	259

This is an example of a very basic short raw text record. We keep the attribute names in English and the values in the original language.

Statuts coordonnés de MEP Industries S.à r.l.\nAll matters not governed by these Articles shall be determined in accordance with the\nlaw of August 10, 1915 on commercial companies, as amended, and the Law.\nPour copie conforme :\nLuxembourg, le 18 août 2015\n Pour la société :\nMaître Marie CASTELLO\n (notaire)\n

And the manually labeled JSON:

```
{
   'Company': {'name': 'MEP Industries S.à r.l.'},
   'Notary': {
       'last_name': 'CASTELLO',
       'first_name': 'Marie'
    }
}
```

For statistical purposes, we classify each document according to a set of predefined classes using a deep learning model that we developed and trained for this purpose. This model predicts the type of information contained in a text page based on its content (Business, Semi-structured or Incomplete Information and Free Layout page) and achieves an F1 score of 98%. We analyzed the results of the different tests based on the type of page. As shown in Table 2, our labeled dataset predominantly consists of free-form text pages, with the second largest portion comprising semi-structured information. These semi-structured pages exhibit various formats of tabular data, often resulting in decreased legibility after OCR processing, which complicates information extraction. Pages with incomplete information are semi-structured and part of a previous page that contains the main information.

Business information includes pages that describes the company, objectives, main activities, etc. This pages tend to contain long paragraphs. Semi-Structured pages are related to pages that are based in diverse templates. Incomplete Information refers to information that are part of any of the two above but does not contains labels or categories which can be used to understand the context.

[5]https://www.lbr.lu

Table 2: Distribution of labeled pages per type of content.

Content type	# Pages	Avg. # words
Business information (BI)	43 (4.3%)	129
Semi-structured information (SS)	216 (21.6%)	165
Incomplete information (II)	35 (3.5%)	134
Free-layout page (FL)	536 (53.6%)	310
Others (OT)	170 (17.0%)	205
Total	1,000 (100%)	259

These documents correspond to four different types of business documents that are shown in Table 3 (classification derived from the types of LBR documents).

Table 3: Distribution of fine-tuning dataset per type of document and language.

Document Type	Fr	Ge	En	Total
Articles of association	66	2	44	112
Modification	373	10	73	456
Non-statutory modif. of the agents	0	119	0	119
Registration	243	11	59	313
Total	682	142	176	1000

Based on the initial dataset of 1,000 manually labeled records, we generated a semi-synthetic dataset comprising 4,000 additional records. This expanded dataset facilitated the creation of three distinct fine-tuning training configurations: (1) only manually labeled records, (2) only semi-synthetic records, and (3) a combination of both manually labeled and semi-synthetic records, as detailed in Table 4. This methodology aims to evaluate the impact of incorporating synthetic data on the performance of the fine-tuning process.

Table 4: Dataset configurations for fine-tuning.

code	Manual (_M)	Synthetic (_S)	Mixed (_X)
_500	500	500	500
_750	750	750	750
_1000	1,000	1,000	1,000
_2000	-	2,000	2,000
_3000	-	3,000	3,000
_4000	-	4,000	4,000

For testing, we ran the model with a dataset of 60 documents with 854 pages. We randomly selected 20 documents from each language and different types of documents. As shown in Table 5, we obtained the distribution of the testing pages by type of content.

Table 6 shows the distribution of the testing dataset by document type. In contrast to the fine-tuning dataset, we include other types of documents such as Annual Accounts, Deletion/Merger, and Consolidated accounts.

Finally, to assess the model's performance on previously unseen languages and formats, we evaluated the best-performing model

Table 5: Distribution of testing dataset per document type and language.

Content type	# Pages
Business information (BI)	74 (8.7%)
Semi-structured information (SS)	178 (20.8%)
Incomplete information (II)	35 (4.0%)
Free-layout page (FL)	494 (57.8 %)
Others (OT)	73 (8.7%)
Total	854 (100%)

Table 6: Distribution of testing dataset per type of document.

Document Type	# Docs	# Pages
Annual accounts (AA)	12	142
Modification (MD)	29	252
Deletion / Merger (DM)	3	20
Consolidated accounts (CA)	3	401
Others (OT)	13	39
Total	60	854

using press releases and business records. These documents contain diverse information about companies and individuals, providing a comprehensive test of the model's robustness. For the press releases, we include nine pages in Swedish and three pages in Spanish. For the business records, we include ten pages in Spanish and four pages in Swedish.

4 Methodology and Proposition

As explained in the Introduction, we are addressing four major issues while training and using the LLM. To deal with the large amount of labeled data required for fine-tuning, hallucination, and the potential leakage of sensitive training data, we are going to create a semi-synthetic dataset and then fine-tune the LLM.

To evaluate the impact of the dataset on the performance of the fine-tuned model, we propose five metrics, as explained in subsection 4.2.

4.1 Creation of a semi-synthetic dataset

For the creation of the semi-synthetic dataset, we use the manually labeled dataset and then replace the sensitive information with synthetic information generated by an LLM. To do this, we do the following steps.

- With the labeled JSONs we create a list of entities and their corresponding attributes. Some entities found were: Person (First name, last name, birthdate, birthplace, role, etc); Company (Name, Legal Tax ID, Legal Form, etc., legal address, branch information); Address (Name, number, postal code, city, country), etc.
- Using the list of entities and attributes we use ChatGPT[6] to generate several alternative lists with synthetic information. One of the biggest difficulties while generating synthetic data was the high repetition and lack of creativity of the resulting lists. To avoid this, we modified the prompt of most of the entities adding a location restriction. For example, for *Person's*: *first name* the initial prompt was:
  "Act as a Synthetic dataset generator and generate a random list of 1,000 Person's first name"
  and the regionalized prompt was:
  "Act as a Synthetic dataset generator and generate a random list of 200 Person's first names from *East Asia*."
  24 geographical regions were used to generate the different entity's attributes such as orth America, Latin America, Central Europe, Northern Africa, etc.
  After we got the list of entities we removed duplicates and consolidated them into a single database of synthetic data.
- With the already labeled data we created our random dataset generator, where we specify the dataset size and start a while loop. We select randomly an index for selecting the labeled text data and for each entity and attribute in the manually labeled JSON we get a random index and get the synthetic information. Then we proceed to replace the true information with the synthetic information.

Following the previously established methodology, we generated a semi-synthetic dataset comprising 4,000 records. Utilizing this synthetic dataset in conjunction with our original dataset of 1,000 labeled records, we will create three distinct dataset configurations of varying sizes. These configurations are detailed in Table 4 and refereed to:

- *Manually labeled dataset*: three different dataset sizes composed only of labeled data: 500, 750 and 1,000.
- *Semi-synthetic dataset*: five different dataset sizes composed only of semi-synthetic data: 500, 750, 1,000, 2,000, 3,000, and 4,000.
- *Mixed dataset*: This configuration consists of five different dataset sizes, composed of both labeled data and semi-synthetic data: 500, 750, 1,000, 2,000, 3,000, and 4,000 records. For dataset sizes less than 2,000, the proportion of labeled to semi-synthetic data is 50%. For dataset sizes of 3,000 and above, each dataset includes 1,000 manually labeled records, with the remainder being semi-synthetic data.

4.2 Baseline evaluation

In the first place, we made a proof of concept of how GPT3.5[7], Mistral 7B[8], and Llama3 8B[9] work to identify the main problems faced by the top LLM. We use the corresponding API to connect our data with these models. We use the same prompt as we use for testing our fine-tuned models which is the following:

"You are a helpful assistant designed to output only a short JSON. Do not add information which is not in the provided text. Please don't share false information. Do not hallucinate. Extract sensitive information from multiple entities in a single JSON object (one level), if there is no information available do not include the field at all. Keep the attribute names in English and the values in the original language. Extract information such as: Person (first_name, last_name, birthdate, birth_country, birth_place, city, country, address_name, address_number; postal_code, manager_type,

[6]https://chat.openai.com

[7]https://platform.openai.com/docs/api-reference

[8]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[9]https://llama.meta.com/llama3

manager_group,function, function_category, start_mandate, end_mandate, shares_number, shares_types)"

The recurrent problems from many of the models were (1) that before or after the JSON there were extra information generated by the model; (2) Attributes without value (empty); (3) Repetitive information such as several times one or several attributes; and (4) Generation of multiple JSON levels to provide the answer.

A common characteristic of all the base models was the language where the attribute names were given, Even if it was explicitly instructed in the prompt that they should be returned in English, some of them were given in the document language. Furthermore, the names and addresses were provided as a single attribute, and we would like the model to try to split into more detailed attributes. We provided the desired structures in the prompt (like in the example for the entity Person).

These previous models were included in our baseline selection together with other models that can also be considered as compact (fit into one GPU). All these models were executed in a server with Tesla V100, with 22.6 TFlops, 1 GPU, 16 GB per GPU, 20GB RAM. For testing the remaining models as is shown in Table 7 we use ollama2[10] as platform to run easily most of LLM models. Is just required to install ollama2 and pull the desired model, an REST endpoint will be available for querying.

Table 7: Baseline models used for comparison with our fine-tuned models

Type	Baseline Model	Size
General	gpt-3.5-turbo-1106 (API)	-
General	claude-3-haiku-20240307 (API)	-
General	Llama3:8b	8b
General	Mistral-7B-Instruct-v0.2	7.3b
General	Gemma (Google)	7b
Coding	CodeLlama-7b-Instruct (Based on Llama2)	7b
Coding	LLaMA-Pro-8B-Instruct (Based on Llama2)	8b
Coding	CodeGemma (Based on Gemma)	7b

We also analyzed other models, including TinyLlama-1.1B-Chat, Qwen-7B-Chat, Llama2-7B, Llama2-13B, Neural-Chat-v3, and WizardCoder-33B. However, their performance was significantly lower, leading us to exclude their results from our final considerations.

4.3 Fine-tuning

Based on the initial baseline results, we fine-tuned the two best free models using different dataset configurations. We employed QLORA and PEFT for three epochs on a T4 GPU in Google Colab[11]. The hyperparameters used were: 4-bit quantization with nf4 quantization type, LoRA dimension of 64, learning rate of 2e-4, weight decay of 0.001, and the AdamW optimizer.

After fine-tuning the models, we use FastAPI[12] to deploy the fine-tuned LLM model as a REST service. This service receives raw text, constructs the appropriate prompt, and uses the model for prediction (text generation). It then generates a response in JSON format. In addition, the service handles retries and exceptions during the prediction process.

[10]https://ollama.com
[11]https://colab.google
[12]https://fastapi.tiangolo.com

4.4 Evaluation

For the testing of the baseline and fine-tuned models we use the second dataset group, see Figure 1 and Tables 5 and 6. After we received the response from the fine-tuned API service, we measured the performance of each test case model with our five proposed metrics. The codification of the test cases for the corresponding models is shown in Table 4.

We ran each test-case scenario three times and report the average results. The standard deviation in most cases is close to zero; therefore, it is not included in the tables.

For evaluation, we propose five metrics based on the problems found while using the models in the proof of concept.

(1) % Validity of JSONs (γ): As we aim for the model to produce pure JSON responses, we measure the percentage of responses that are valid JSON without any additional information or remarks. A higher percentage indicates a better quality of the desired format. If necessary, we will create a valid JSON by removing all content before the first "{" and after the last "}". However, we do not alter this metric even if a fix is possible. The symbol ↑ denotes that higher values are preferable.
(2) % Existing-Data (ϵ): If the model's response is a valid JSON or can be extracted by removing extraneous text before and after the JSON structure, we automatically verify whether the attribute values of each entity are present in the input text. A higher percentage indicates a lower level of hallucination. The symbol ↑ denotes that higher values are preferable.
(3) % Emptiness(ϕ): Considering extractable JSON responses, we measure the number of empty attribute values in the model's output. Lower values indicate better model flexibility and adaptability to different scenarios. The symbol ↓ denotes that lower values are preferable.
(4) % Repetitions(ψ): Furthermore, when considering extractable JSON responses, we measure the number of repeated attribute names and/or attribute values in the model's output. A lower percentage of repetitions indicates higher quality of the response. The symbol ↓ denotes that lower values are preferable.
(5) % 2-Level Structure(α): This measure focuses on JSON structures that are relatively shallow and straightforward. We expect each JSON response to consist of a main object and its corresponding attributes, without nested complex objects within the main structure. The required format is exemplified in the prompt.

The metrics ϕ and ψ reflect the prevalence of repeated data and empty values in responses from LLM models. Despite extracting only a few attributes in some cases, we encountered extensive JSON objects with repeated or empty attributes. This often resulted in invalid JSON due to exceeding the maximum number of output tokens. Metric α assesses adherence to the required prompt schema.

Is important to have in zero the metrics ψ and ϕ, to be considered a consistent model, but also can be done in a subsequent cleaning phase. The most important metrics are γ and ϵ which are refered

to a correct output, easy to process and the *no hallucination*. The last metric α could be also considered important to reduce the post processing.

5 Experiments and Results

Table 8 presents the results of five metrics across various baseline models. The highest scores were achieved by GPT-3.5 (cloud-based, paid service), Mistral, and Llama3 (free, local service).

Table 8: Baseline model's performance results

Model	$\gamma\uparrow$ (%)	$\epsilon\uparrow$ (%)	$\phi\downarrow$ (%)	$\psi\downarrow$ (%)	$\alpha\uparrow$ (%)
GPT-3.5	**89.3**	**79.5**	14.9	4.3	**71.7**
Claude3	48.9	46.4	**0.0**	**3.6**	40.4
llama3:8B	0.0	30.9	61.6	1.1	11.9
mistral:7b	**47.7**	**36.0**	0.1	3.1	12.2
codellama:7b	23.8	33.0	0.4	3.0	**18.5**
llama-pro:8b	1.5	23.6	0.1	2.7	8.2
gemma:7b	0.0	29.7	65.3	1.4	10.9
codegemma:7b	38.6	38.5	43.2	1.8	8.9

For a detailed analysis, we are not analyzing the %Validity of JSONs (γ), because it can be partially solved when we remove the text before and after JSON. On the other hand, the %Emptiness (ϕ) and the %Repetitions (ψ) also can easily solved by code. For this reason, we will focus on a detailed analysis of % Existing-Data (ϵ) and % 2-Level Structure (α). These metrics are directly related to the non-hallucination capacity and the following structure requirements (Table 9).

Table 9: Baseline model's performance results by type of content: ϵ & α

Model	BI (%)		FL (%)		II (%)		SS (%)	
	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$
GPT-3.5	**100**	**100**	**79**	**70**	**50**	**50**	**95**	**97**
Claude3	79	78	44	38	50	50	89	89
llama3:8B	60	40	30	10	**50**	**50**	56	**39**
mistral:7b	67	21	**34**	12	50	34	**77**	7
codellama:7b	**83**	**83**	32	**17**	0	0	53	**39**
llama-pro:8b	54	33	23	8	50	25	22	14
gemma:7b	0	0	30	11	0	0	38	16
codegemma:7b	60	38	8	**50**	50	0	**41**	19

Based on the previous results, the two best performed free models are Llama3 [13] and Mistral 7-b [14]. These are the models that we are going to fine tune with the different dataset configurations.

Table 10 demonstrates that the best-performing models are based on Llama3. These models consistently avoid returning empty data, adhere to the two-level JSON structure, and generate valid JSON outputs. The model fine-tuned with manually labeled information (M) performed the best in terms of %Existing data (ϵ), with the Mixed model (X) following closely behind, despite being fine-tuned with only 500 samples.

Table 10: Performance of fine-tuned models

Model	$\gamma\uparrow$	$\epsilon\uparrow$	$\phi\downarrow$	$\psi\downarrow$	$\alpha\uparrow$
Mistral_M500	53.0	52.3	27.4	2.0	52.7
Mistral_M750	59.4	58.0	29.7	2.1	58.2
Mistral_M1000	**61.6**	**60.3**	**27.1**	**1.3**	**61.0**
Mistral_S500	36.7	49.1	29.3	0.8	50.3
Mistral_S750	34.9	49.2	**24.4**	0.6	51.5
Mistral_S1000	**51.5**	**50.9**	33.7	1.0	**52.3**
Mistral_S2000	5.1	11.7	38.0	**0.2**	12.6
Mistral_S3000	9.3	21.0	50.3	0.2	20.9
Mistral_S4000	0.8	14.4	47.6	0.3	15.2
Mistral_X500	**59.0**	**61.6**	**19.8**	0.6	**62.7**
Mistral_X750	46.3	50.3	24.1	1.2	52.2
Mistral_X1000	47.0	48.4	31.8	1.5	49.8
Mistral_X2000	15.0	23.0	51.8	**0.2**	22.8
Mistral_X3000	0.5	15.6	81.0	0.3	16.1
Mistral_X4000	9.1	13.4	63.8	0.3	13.5
Llama3_M500	100	**78.1**	0.0	6.7	100
Llama3_M750	100	64.4	0.0	**2.2**	100
Llama3_M1000	100	72.9	0.0	3.7	100
Llama3_S500	100	46.6	0.0	1.1	100
Llama3_S750	100	34.3	0.0	1.4	100
Llama3_S1000	100	**66.3**	0.0	1.7	100
Llama3_S2000	100	44.0	0.0	1.3	100
Llama3_S3000	100	53.4	0.0	1.3	100
Llama3_S4000	100	27.6	0.0	**0.9**	100
Llama3_X500	100	**71.8**	0.0	2.6	100
Llama3_X750	100	67.0	0.0	2.4	100
Llama3_X1000	100	51.8	0.0	2.0	100
Llama3_X2000	100	52.5	0.0	**1.8**	100
Llama3_X3000	100	32.4	0.0	1.7	100
Llama3_X4000	100	45.5	0.0	1.8	100

Table 11 presents a detailed performance analysis of the various fine-tuned models by content type, focusing on ϵ and α. Among the Llama3-based models, the Manual (M) and Mixed (S) models with 500 samples performed the best. For the Mistral-based models, those labeled manually (M) exhibited strong performance, and the Mixed (X) models with just 500 records also showed outstanding results. Notably, the model trained with Semi-Synthetic data (S) and 1,000 samples performed well, closely matching the performance of the other top models.

Table 12 shows the performance of the baseline models with the testing dataset in scenarios of unseen types of documents with respect to the fine-tunning data (Annual Accounts[AA], Consolidated Accounts [CA], Deletion/Merger [DM]).

Table 13 presents the performance of the fine-tuned models on a testing dataset featuring unseen document types, including Annual Accounts (AA), Consolidated Accounts (CA), and Deletion/Merger (DM). The results indicate that ChatGPT-3.5 outperforms Claude-3 across all unseen document types. Additionally, Mistral and CodeLlama are among the best-performing models overall. However, it is

[13] unsloth/llama-3-8b-bnb-4bit
[14] mistralai/Mistral-7B-Instruct-v0.2

Table 11: Performance of fine-tuned models by type of content: ϵ & α

Model	BI (%)		FL (%)		II (%)		SS (%)	
	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$
Mistral_M500	100	100	50	50	50	50	94	94
Mistral_M750	100	100	56	56	50	50	97	97
Mistral_M1000	**100**	**100**	**57**	**57**	**50**	**50**	**97**	**97**
Mistral_S500	83	83	47	48	**50**	**50**	**85**	**89**
Mistral_S750	**100**	**100**	47	50	50	50	79	83
Mistral_S1000	67	67	**50**	**51**	50	50	74	78
Mistral_S2000	40	40	10	11	50	50	33	33
Mistral_S3000	33	33	21	21	0	0	19	19
Mistral_S4000	33	33	15	16	0	0	4	6
Mistral_X500	83	83	**61**	**63**	**50**	**50**	**61**	**61**
Mistral_X750	83	83	51	53	50	50	44	44
Mistral_X1000	**100**	**100**	47	48	0	0	72	72
Mistral_X2000	0	0	24	24	0	0	0	0
Mistral_X3000	33	33	16	16	0	0	8	8
Mistral_X4000	13	17	14	14	0	0	11	11
Llama3_M500	**100**	**100**	**77**	**100**	50	**100**	**97**	**100**
Llama3_M750	100	100	62	100	50	100	94	100
Llama3_M1000	100	100	71	100	**100**	100	95	100
Llama3_S500	0	100	46	100	0	100	64	100
Llama3_S750	100	100	31	100	50	100	88	100
Llama3_S1000	**100**	**100**	**64**	**100**	**100**	**100**	89	**100**
Llama3_S2000	50	100	42	100	50	100	86	100
Llama3_S3000	83	100	51	100	50	100	**97**	100
Llama3_S4000	83	100	24	100	50	100	89	100
Llama3_X500	**100**	**100**	**70**	**100**	50	**100**	**97**	**100**
Llama3_X750	100	100	65	100	100	100	93	100
Llama3_X1000	83	100	49	100	100	100	96	100
Llama3_X2000	100	100	50	100	100	100	89	100
Llama3_X3000	83	100	30	100	100	100	70	100
Llama3_X4000	83	100	43	100	50	100	86	100

Table 12: Baseline models Performance by document type: ϵ & α

Model	AA (%)		CA (%)		DM (%)	
	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$
GPT-3.5	**88**	**86**	**78**	**73**	**64**	**64**
Claude3	67	60	42	38	18	18
llama3:8B	**55**	**50**	**35**	**25**	45	20
mistral:7b	26	13	13	11	**55**	**26**
codellama:7b	38	24	26	15	30	0
llama-pro:8b	30	12	15	6	25	14

worth noting that Mistral's performance on unseen documents is not particularly high.

Among our fine-tuned models, the Llama3-based model trained with only manual data (M) or mixed data (X) with 500 samples emerged as the best performer. Its results are comparable to GPT-3.5 and even surpass it in terms of (α). Consistent with previous findings, the model trained with semi-synthetic data alone also performs well, even outperforming others for DM documents.

Furthermore, we performed a manual check using the best-performing model on a sample of 26 pages, with an equal distribution of Spanish and Swedish content (50% each). Therefore, deploying the highest-performing semi-synthetic-based model as a service for extracting information from Swedish and Spanish documents yielded the results presented in Table 14.

Table 13: Fine-tuned models performance by document type: ϵ & α

Model	AA (%)		CA (%)		DM (%)	
	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$	$\epsilon\uparrow$	$\alpha\uparrow$
Mistral_M500	66	67	58	58	63	64
Mistral_M750	76	76	62	62	62	64
Mistral_M1000	**82**	**83**	**66**	**67**	**63**	**64**
Mistral_S500	59	60	45	46	55	55
Mistral_S750	**65**	**67**	45	48	**58**	**64**
Mistral_S1000	56	56	**51**	**53**	51	55
Mistral_S2000	16	16	8	9	22	22
Mistral_S3000	31	31	20	20	26	27
Mistral_S4000	16	17	18	20	9	9
Mistral_X500	**64**	**64**	**62**	**64**	27	27
Mistral_X750	57	58	54	57	53	55
Mistral_X1000	59	60	44	47	**54**	**55**
Mistral_X2000	8	9	26	25	0	0
Mistral_X3000	17	17	17	18	9	9
Mistral_X4000	21	21	12	12	18	18
Llama3_M500	**85**	**100**	**80**	**100**	**62**	**100**
Llama3_M750	80	100	62	100	63	100
Llama3_M1000	84	100	62	100	100	100
Llama3_S500	52	100	47	100	0	100
Llama3_S750	50	100	27	100	60	100
Llama3_S1000	**83**	**100**	**53**	**100**	**97**	**100**
Llama3_S2000	60	100	44	100	45	100
Llama3_S3000	65	100	49	100	72	100
Llama3_S4000	37	100	17	100	27	100
Llama3_X500	**85**	**100**	**72**	**100**	**54**	**100**
Llama3_X750	76	100	59	100	50	100
Llama3_X1000	67	100	38	100	97	100
Llama3_X2000	66	100	43	100	100	100
Llama3_X3000	44	100	18	100	100	100
Llama3_X4000	68	100	37	100	45	100

Furthermore, we performed a manual check using the best-performing model on a sample of 26 pages, with an equal distribution of Spanish and Swedish content (50% each). Therefore, deploying the highest-performing semi-synthetic-based model as a service for extracting information from Swedish and Spanish documents yielded the results presented in Table 14.

Table 14: Performance of the best performed semi-synthetic-based model with other languages and formats

Model	$\gamma\uparrow$	$\epsilon\uparrow$	$\phi\downarrow$	$\psi\downarrow$	$\alpha\uparrow$
Llama3_S1000	100	81	0	4	100

According to the results, the model's performance is excellent, nearly eliminating the percentage of repetitions (ψ) and reducing the percentage of empty fields (ϕ) to almost zero. The JSON validity percentage (γ) and the 2-Level Structure percentage (α) reached

100%, producing the expected JSON outputs. Additionally, the percentage of existing data (ϵ) is very high, approximately 81%. While α could be in fact, higher, the primary errors identified during manual checks were due to OCR typo corrections, ambiguous company or personal names, and lists of people with closely aligned first and last name columns.

6 Discussion

Fine-tuning Llama3 yields superior results in terms of output structure. For data quality, incorporating some real data is preferred for optimal performance. However, if it is necessary to exclude sensitive data from fine-tuning, training exclusively with semi-synthetic data in the same proportion as real data still produces good results. This is also evident for unseen documents.

The fine-tuned models are not biased with respect to the training data. We got significant improvements in every single category in most of the cases with respect to our base model Llama3:8B, codellama and Mistral. Our fine-tuned model can be applied to any domain where sensitive information (companies and people) can be found, regardless of whether OCR noise exists.

It is important to highlight that for documents in unseen formats and languages, the model trained exclusively with semi-synthetic data provides very good results. This allows us to share this compact fine-tuned model for use in other domains and formats where extracting sensitive information from raw text is required. Additionally, this model can be further fine-tuned for languages beyond French, German, and English, improving its ability to handle ambiguities in the target language even more effectively.

7 Conclusion and next steps

Our five proposed metrics enable objective comparison of different LLM models. These metrics can be weighted according to business needs, with the most important being the percentage of data existence (ϵ) and JSON structure (α).

Training smaller models such as Llama3-8B and Mistral 7B with limited training samples can significantly enhance the performance of Information Extraction models. Labeling the training samples and subsequently replacing them with semi-synthetic data notably improves the fine-tuned model's results, especially in the absence of data augmentation. Utilizing semi-synthetic data for fine-tuning helps maintain the model's independence from sensitive information while enhancing inference outcomes. Fine-tuning Llama3, in particular, yields superior performance compared to other models.

Given the importance of fine-tuning with semi-synthetic data to preserve sensitive information within the organization, our methodology for creating a semi-synthetic dataset ensures high-quality model performance while minimizing the need for a large number of manually labeled documents. Our best-performing semi-synthetic dataset, which can be implemented on a small GPU server, is publicly available for general use. This Llama3-based model consistently generates high-quality JSON outputs, effectively handling sensitive information from the provided text.

As the next step, we plan to anonymize the dataset and make it publicly available for broader use and research, utilizing the best fine-tuned model.

Acknowledgments

This research was funded in whole or in part by the Luxembourg National Research Fund (FNR), grant reference 15403349. For the purpose of open access, and in fulfilment of the obligations arising from the grant agreement, the author has applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

References

[1] Kiran Adnan and Rehan Akbar. 2019. An analytical study of information extraction from unstructured and multidimensional big data. *Journal of Big Data* 6, 1 (2019), 1–38.

[2] Kiran Adnan and Rehan Akbar. 2019. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management* 11 (2019), 1847979019890771. https://doi.org/10.1177/1847979019890771

[3] Kiran Adnan and Rehan Akbar. 2019. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management* 11 (2019), 1847979019890771.

[4] Kiran Adnan, Rehan Akbar, and Khor Siak Wang. 2019. Information extraction from multifaceted unstructured big data. *International Journal of Recent Technology and Engineering (IJRTE)* 8 (2019), 1398–1404.

[5] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919* (2017).

[6] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 344–354.

[7] Deloitte. 2019. *Analytics and AI-driven enterprises thrive in the Age of With.* https://www2.deloitte.com/us/en/insights/topics/analytics/insight-driven-organization.html

[8] Forbes. 2024. *Hackers Breached Hundreds Of Companies' AI Servers, Researchers Say.* https://www.forbes.com/sites/thomasbrewster/2024/03/26/hackers-breach-hundreds-of-ai-compute-servers-researchers-say/ Accessed: March 27, 2024.

[9] Yongyan Guo, Jiayong Liu, Wenwu Tang, and Cheng Huang. 2021. Exsense: Extract sensitive information from unstructured data. *Computers and Security* 102 (2021), 102156. https://doi.org/10.1016/j.cose.2020.102156

[10] Michael Hart, Pratyusa Manadhata, and Rob Johnson. 2011. Text classification for data loss prevention. In *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 18–37.

[11] Chu-Hsing Lin, Po-Kai Yang, and Yu-Chiao Lin. 2020. Detecting security breaches in personal data protection with machine learning. In *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. IEEE.

[12] Michael Meli, Matthew R McNiece, and Bradley Reaves. 2019. How bad can it git? characterizing secret leakage in public github repositories.. In *NDSS*.

[13] Umara Noor, Zahid Anwar, Asad Waqar Malik, Sharifullah Khan, and Shahzad Saleem. 2019. A machine learning framework for investigating data breaches based on semantic analysis of adversary's attack patterns in threat intelligence repositories. *Future Generation Computer Systems* 95 (2019), 467–487.

[14] Jianfeng Qu, Wen Hua, Dantong Ouyang, and Xiaofang Zhou. 2022. An efficient and effective approach for multi-fact extraction from text corpus. *World Wide Web* (2022), 1–24.

[15] Yuri Shapira, Bracha Shapira, and Asaf Shabtai. 2013. Content-based data leakage detection using extended fingerprinting. *arXiv preprint arXiv:1302.2028* (2013).

[16] Xiaokui Shu, Danfeng Yao, and Elisa Bertino. 2015. Privacy-preserving detection of sensitive data exposure. *IEEE transactions on information forensics and security* 10, 5 (2015), 1092–1103.

[17] Yan Shvartzshnaider, Zvonimir Pavlinovic, Ananth Balashankar, Thomas Wies, Lakshminarayanan Subramanian, Helen Nissenbaum, and Prateek Mittal. 2019. Vaccine: Using contextual integrity for data leakage detection. In *The World Wide Web Conference*. 1702–1712.

[18] StanfordNLP. 2020. *OpenIE.* https://stanfordnlp.github.io/CoreNLP/openie.html

[19] Slim Trabelsi. 2019. Monitoring leaked confidential data. In *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE.

[20] Bowen Yu, Zhenyu Zhang, Jingyang Li, Haiyang Yu, Tingwen Liu, Jian Sun, Yongbin Li, and Bin Wang. 2022. Towards Generalized Open Information Extraction. arXiv:2211.15987 [cs.CL]

[21] Gohar Zaman, Hairulnizam Mahdin, Khalid Hussain, and Atta Rahman. 2020. Information extraction from semi and unstructured data sources: A systematic literature review. *ICIC Express Letters* 14, 6 (2020), 593–603.