ASC-PIE: An Evaluation Framework for PII-Aware Named-Entity Recognition

Mohamed Nader Mohamed Hafez

A Thesis Submitted to the Faculty of Graduate Studies In Partial Fulfillment of the Requirements for the Degree of Master of Arts

Graduate Program in Information Systems & Technology

York University

Toronto, Ontario

April 2026

Abstract

The robust extraction of Personally Identifiable Information (PII) is essential for privacy protection in modern text-processing systems, where users often share sensitive details in prompts, emails, chat logs, and support tickets. As PII categories and deployment domains evolve, updating extraction models can improve coverage but may also cause catastrophic forgetting of previously learned types.

This thesis investigates how PII extraction can remain accurate, reliable, and maintainable as task scope expands. It introduces ASC-PIE, a unified English corpus and evaluation framework that combines public datasets with a synthetic component to improve coverage of rare and challenging PII cases without using real personal data. Using ASC-PIE, the thesis compares encoder-based, encoder-decoder, and decoder-only models under supervised fine-tuning and in-context prompting. It also proposes SPRINT-PP, a privacy-safe continual-learning method that mitigates forgetting without storing raw historical examples. The evaluation covers extraction quality, robustness, output validity, computational efficiency, and knowledge retention.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Marin Litoiu, for his invaluable guidance, continuous support, and encouragement throughout my master's journey. His mentorship, patience, and insightful feedback have been instrumental in shaping this research and in helping me grow both academically and personally. I am truly grateful for the opportunity to work under his supervision and for the trust he placed in me throughout this work.

I would also like to extend my sincere thanks to Manar Jammal for her support, guidance, and encouragement during this journey. Her help and thoughtful advice have been greatly appreciated.

My heartfelt thanks also go to my lab mates, Amr, Farhoud, Sara, Hong, Komal, Chen, Abed, Victor, Yar, Farshid, Ghadeer, and Lamisa, for their support, discussions, and the positive environment they created in the lab. Sharing this journey with such a supportive group made the experience both productive and enjoyable.

I am also grateful to my friends, especially Ali, Etman, and Sandi, as well as all my other friends, for their encouragement, kindness, and support throughout this journey. Their presence, understanding, and friendship helped make the difficult moments easier and the good moments more meaningful.

I am deeply grateful to my parents and my brother for their unconditional love, sacrifices, and constant encouragement. Their support has always been my foundation and has given me the strength to continue through every challenge.

Finally, I would like to express my special and deepest thanks to my fiancée, Ghada, for her patience, love, and unwavering support. Her constant encouragement, understanding, and belief in me have meant more than words can express, and they have been a source of strength throughout this entire journey.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

We address a practical and high-impact challenge: the robust extraction of Personally Identifiable Information (PII) from unstructured text, coupled with the preservation of this capability as system requirements evolve [1, 2]. Organizations routinely process heterogeneous text from diverse sources such as support tickets, chat logs, email correspondence, and internal forms. Because failures in identifying PII weaken privacy safeguards and elevate analytical risks, accurate extraction serves as a foundational component of data governance within real-world pipelines.

PII extraction has evolved through several methodological stages. Early approaches relied on rule-based and statistical techniques, which were often effective in narrow domains but difficult to generalize across datasets and changing annotation requirements [3]. Later, deep learning methods, especially neural sequence labeling models, improved contextual modeling and reduced the need for manual feature engineering [4, 5]. More recently, pretrained Transformer models and instruction-following large language models have become central to PII extraction because they provide stronger language representations, support flexible output formats, and can be adapted to diverse extraction settings [6–9]. Despite these advances, operational environments remain inherently dynamic. PII inventories expand as new identifiers become relevant, annotation guidelines are revised, and data distributions drift across domains over time. Updating an extraction model to accommodate these shifts is therefore a necessary operational requirement. However, such updates can degrade performance on previously learned PII categories, a phenomenon widely studied in continual learning and commonly referred to as catastrophic forgetting [10–12]. This degradation directly undermines system reliability and reflects the broader stability-plasticity trade-off in evolving machine learning systems [12, 13].

Simultaneously, the advent of modern large language models (LLMs) introduces alterna-

tive extraction paradigms, including instruction-driven prompting and in-context learning [8, 14, 15]. While these generative approaches provide flexible interfaces for specifying tasks through natural language, they also raise critical new questions regarding output validity, constraint adherence, reproducibility, and computational cost [8, 16]. In parallel, catastrophic forgetting mitigation in privacy-sensitive settings introduces an additional constraint: replay can be effective, but it requires storing historical examples that may contain raw identifiers, while privacy-safe strategies such as distillation do not explicitly distinguish between genuinely new supervision and old-entity tokens that have been relabeled as `O` under staged annotation [2, 17, 18]. Together, these operational realities motivate a unified evaluation framework and a privacy-conscious incremental-learning study that compare disparate extraction strategies under consistent data handling, staged task growth, and shared reporting criteria.

1.2 Problem Statement

The core problem addressed in this thesis can be formalized as follows. Let an input text be represented as a sequence

$$X = (x_1, x_2, \ldots, x_n),$$

where each x_i denotes a token in the text. The goal is to identify a set of spans

$$\mathcal{S} = \{(b_j, e_j, y_j)\}_{j=1}^{m},$$

where b_j and e_j denote the begin and end positions of the j-th extracted span in X, and $y_j \in \mathcal{Y}$ is its assigned label from a predefined set of PII types under a consistent tagging scheme [7]. In other words, the task is to recover all text spans corresponding to PII and assign each span its correct semantic category.

This extraction setting is challenging because PII appears in diverse surface forms, is highly context-dependent, and occurs in noisy, informal, and domain-specific language [19].

Because operational requirements evolve through expanding PII categories, revised guidelines, and shifting domains, extraction systems require repeated adaptation rather than a single static training cycle [12, 20].

Formally, the learning objective is to estimate a function $f_\theta : X \mapsto \mathcal{S}$ that predicts the set of labeled PII spans for any input sequence.

The central methodological challenge is to maintain extraction performance as the task scope expands, particularly in a class-incremental learning setting where the label space grows across stages [18, 20, 21]. As models are updated with new data to introduce additional entity types or distributions, they frequently suffer from catastrophic forgetting, a phenomenon wherein performance on previously learned PII categories degrades [12, 13]. In privacy-sensitive deployments, this challenge is compounded by the fact that one of the strongest continual-learning baselines, replay, requires retaining earlier training examples that may contain raw identifiers, whereas privacy-safe strategies such as distillation apply preservation signals more broadly and do not explicitly distinguish old-entity tokens that have been relabeled as `O` under class-incremental training [12, 13, 17, 18]. The problem therefore requires methods and evaluation protocols that quantify both the learning of new capabilities and the retention of prior ones while remaining compatible with privacy-aware operating constraints.

Furthermore, this challenge spans multiple modeling paradigms that formulate PII extraction differently. Encoder-based models typically treat the task as token-level sequence labeling, encoder-decoder models formulate extraction as structured generation, and decoder-only instruction-following models perform extraction through prompted generation with optional parameter updates [7, 15, 22]. These paradigms differ in their training interfaces, output constraints, and failure modes, including invalid or partially structured outputs for generative approaches [8, 23]. Consequently, the need to maximize extraction quality while ensuring output validity, robustness under realistic noise, and practical efficiency motivates a unified experimental setting that supports consistent comparisons across model families.

1.3 Research Questions

We address the following research questions (RQs), each targeting a core aspect of PII extraction under evolving task requirements. These questions are motivated by prior work on continual and class-incremental Named Entity Recognition (NER), as well as recent instruction-based and LLM-driven extraction paradigms [8, 20, 21].

1. **RQ1 (Comparison of model families under supervised fine-tuning):** Under a consistent dataset, label schema, and evaluation protocol, how do encoder-based token classification, encoder-decoder structured prediction, and decoder-only instruction-following models compare when trained via supervised fine-tuning in terms of extraction quality, robustness, output validity, and computational efficiency?

2. **RQ2 (Fine-tuning versus in-context prompting):** For the same extraction task and evaluation protocol, how does in-context prompting compare to supervised fine-tuning across the considered model families, and how do key prompting factors, such as the number of shots, affect performance, structural reliability, and inference cost?

3. **RQ3 (Class-incremental learning, forgetting, and mitigation):** When the label space and data distribution expand across stages, to what extent do fine-tuned models exhibit catastrophic forgetting, how do replay and distillation behave under a unified protocol, and can the proposed privacy-safe SPRINT-PP method better balance the learning of new PII capabilities with the retention of previously learned ones?

4. **RQ4 (Effect of corpus design and synthetic augmentation):** How do the choices made when building the ASC-PIE corpus, including which public datasets are included, how labels are normalized under a unified schema, and how synthetic examples are added to increase coverage, affect extraction performance, particularly for rare or challenging PII types, and how do these effects vary across model families and adaptation strategies?

1.4 Contributions

The primary contributions are as follows:

1. **The ASC-PIE Framework for cross-paradigm evaluation.** We develop a unified experimental and evaluation framework for comparing three model families, encoder-based token classification, encoder–decoder structured generation, and decoder-only instruction-following models, under consistent data splits, constraints, and metrics. This contribution supports **RQ1** by enabling direct supervised fine-tuning comparison across model families, and supports **RQ2** by enabling a controlled comparison between supervised fine-tuning and in-context prompting, including the analysis of prompting factors such as shot count, output validity, structural reliability, and inference cost.

2. **SPRINT-PP for privacy-safe continual learning.** We formalize a three-stage class-incremental setting in which the label space and data distribution evolve over time, and we evaluate baseline sequential fine-tuning together with replay and distillation under shared continual-learning metrics. Within this setting, we propose SPRINT-PP (Selective Prototype-Routed Incremental NER Triage with Privacy Preservation), a privacy-safe mitigation method that combines selective teacher-guided correction, prototype-based anchoring, and classifier-head stabilization to reduce catastrophic forgetting without storing raw historical PII. This contribution addresses **RQ3**.

3. **ASC-PIE corpus and synthetic augmentation.** We introduce ASC-PIE (Augmented Synthetic Corpus for PII and Incremental Evaluation), which consolidates heterogeneous English PII and NER resources into a single dataset and normalizes all sources to a unified 19-type label schema for consistent training and evaluation. We further design and integrate a purpose-built synthetic component to improve coverage of long-tail, high-structure, and otherwise under-supported PII patterns without using real personal data, enabling controlled analysis of difficult entity types and corpus-design effects. This contribution addresses **RQ4**.

4. **Reproducible artifacts.** We provide standardized exported artifacts, including dataset manifests, configurations, evaluation summaries, per-type reports, and prediction traces, to support transparent reporting, implementation traceability, and replication of the study.

1.5 Scope and Assumptions

This thesis investigates PII extraction from English unstructured text within a unified framework for corpus construction, model comparison, and evaluation. The study covers three model families: encoder-based token classification, encoder-decoder structured prediction, and decoder-only instruction-following extraction. It examines supervised fine-tuning across these families and, for decoder-only models, in-context prompting under controlled prompt settings. In addition, the thesis considers a staged class-incremental setting in which the label space expands over time, enabling the study of catastrophic forgetting together with privacy-aware mitigation.

ASC-PIE is constructed by consolidating heterogeneous public resources with a thesis-specific synthetic component, all normalized to a unified privacy-oriented label schema. The target inventory is defined as privacy-relevant PII types, consistent with standard definitions used in privacy practice [1]. The study is restricted to English to maintain controlled comparability across datasets and model families, although the framework itself can be extended to other languages given suitable corpora, schema mappings, and preprocessing resources. Throughout the thesis, supervision is treated in span-based form and normalized to a consistent tagging and evaluation scheme [24].

To preserve tractable and comparable evaluation, the thesis adopts the following assumptions:

- **Unified span-based representation.** PII mentions can be represented as labeled spans and normalized to a common schema, enabling valid comparison across heterogeneous datasets and model families.

- **Staged task growth as an operational proxy.** Incremental expansion of the label space provides a meaningful proxy for deployment settings in which new PII categories become relevant over time and models must be updated without reconstructing a fully joint training set.

- **Synthetic augmentation as controlled coverage expansion.** Synthetic data can be used to improve coverage of rare and highly structured PII patterns without exposing real personal data, and its value is assessed empirically rather than assumed.

- **Shared evaluation under privacy-aware constraints.** Outputs from token classifiers and generative models can be normalized to a common entity representation and compared under shared scoring criteria, while continual-learning methods should also be assessed with respect to privacy-sensitive operating constraints.

1.6 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 introduces key terminology and background concepts relevant to PII-aware extraction, output validity, and catastrophic forgetting under incremental updates. Chapter 3 reviews related work on privacy-oriented datasets, generative NER architectures, continual learning methods, and the literature gaps that motivate the proposed study. Chapter 4 presents the ASC-PIE framework, including corpus construction, staged incremental design, model-family formulations, adaptation strategies, evaluation protocol, and reproducibility considerations. Chapter 5 reports the empirical results for the research questions, analyzing supervised architectural comparisons, prompting strategies, continual-learning behavior, and the effect of corpus design and synthetic augmentation. Finally, Chapter 6 concludes the thesis by summarizing the main findings, discussing limitations, and outlining future directions, with supplementary materials provided in the appendices.

2 Terminology & Background

2.1 Key Terms

Table 2.1 summarizes the core terminology used throughout this thesis. These definitions establish a consistent interpretation of dataset construction, task formulations across model families, evaluation criteria, and incremental learning concepts. The terms are referenced across chapters to support comparability and to reduce ambiguity when reporting experimental results.

Table 2.1: Key terms and definitions used throughout the thesis.

Term	**Definition**
Personally Identifiable Information (PII)	Information that can identify an individual directly or indirectly, either alone or in combination with other attributes. In this thesis, PII refers to privacy-relevant identifiers expressed in text, such as names, contact details, addresses, government identifiers, and account-related identifiers [1, 2].
PII extraction	The task of locating all PII mentions in an input text and assigning each mention a type from a predefined inventory. We treat extraction as span-based: the output is a set of entity spans paired with PII types [27, 28].
Named Entity Recognition (NER)	A sequence labeling task that identifies entity mentions in text and assigns each mention a category from a predefined label space [3, 24].
PII-aware NER	A privacy-oriented specialization of NER in which the entity inventory is defined by PII types and evaluation emphasizes correctness, reliability, and practical utility for redaction and governance [27, 28].
Entity mention and entity span	An entity mention is a specific occurrence of an entity in text. An entity span is the exact contiguous character range corresponding to that mention, used as the reference representation for extraction and evaluation [24].
Entity type	A category of PII (e.g., `EMAIL` or `PHONE`) that specifies how an extracted span should be interpreted under the unified schema.

Continued on next page

Term	Definition
Label schema	The complete specification of the entity types and the canonical label names used across datasets and model families. A unified label schema enables consistent training, decoding, and evaluation across heterogeneous sources.
Token, subword, and tokenization	A token is the unit consumed by a model, often produced by a subword tokenizer that may split a word into multiple pieces. Tokenization determines how text is segmented, and it affects how character-level spans are aligned to token-level representations [29, 30].
Label alignment	The procedure that maps span annotations to the tokenized input representation. For token classification, correct alignment is required to ensure that token labels correspond to the intended entity spans after tokenization.
IOB2 tagging	A standard token-level tagging scheme where `B-TYPE` marks the first token of an entity of type `TYPE`, `I-TYPE` marks subsequent tokens in the same entity, and `O` marks tokens outside any entity [24, 31].
Span-level versus token-level evaluation	Span-level evaluation treats a multi-token entity as a single decision, while token-level evaluation scores each token independently. Span-level metrics are prioritized because they directly reflect extraction quality for redaction and auditing [24, 27].
Span-level precision, recall, and F1	Precision is the fraction of predicted entity spans that are correct, recall is the fraction of ground-truth spans recovered, and F1 is their harmonic mean. In this thesis, span-level F1 is the primary performance measure for extraction [24].
Model families	We consider three families that implement extraction differently: encoder-based token classification, encoder-decoder structured prediction, and decoder-only instruction-following extraction. Each family differs in its interface, constraints, and failure modes [7, 8, 32].
Encoder model	A bidirectional Transformer encoder adapted for extraction via a token classification head, producing a label per token [6, 7].
Encoder-decoder model	A sequence-to-sequence Transformer that encodes an input sequence and generates an output sequence, enabling extraction through structured generation conditioned on the input text [32, 33].

Continued on next page

Term	Definition
Decoder-only model	An autoregressive Transformer that generates output tokens sequentially. In this thesis, decoder-only models perform extraction through prompted generation, optionally combined with parameter-efficient fine-tuning under a constrained output format [9, 14].
Supervised fine-tuning	Task-specific training that updates model parameters using labeled data for extraction [7].
Parameter-efficient fine-tuning (PEFT)	Adaptation methods that update a small set of additional or selected parameters while keeping most of the base model fixed, reducing memory and compute requirements relative to full fine-tuning [25, 26].
LoRA	A PEFT method that introduces low-rank trainable matrices in selected layers, typically within attention modules, enabling efficient adaptation while keeping base weights fixed [25].
QLoRA	A PEFT configuration in which the base model weights are quantized, commonly to 4-bit precision, while LoRA adapters are trained in higher precision. This reduces memory usage and enables the fine-tuning of larger models under limited hardware [26].
In-context learning (ICL)	A prompting paradigm in which task instructions and a small number of labeled examples are included in the input context at inference time. The model performs extraction without updating its parameters [14].
Shot and number of shots	A shot is a labeled example included in the prompt. The number of shots refers to how many examples are provided, which influences performance, prompt length, and inference cost [14].
Prompt template	A structured prompt design that specifies task instructions, optional examples, and the expected output format. Prompt templates enable controlled comparisons across models and shot regimes [8].
Structured output	A machine-readable representation of extracted entities produced by generative models. In this thesis, structured outputs are normalized to a common representation of entities, such as span text and type, with optional offsets when supported [8, 22].
Output validity	The degree to which a produced output satisfies structural constraints and uses only allowed label names. Validity checks distinguish extraction errors from formatting violations in generative outputs [8].

Continued on next page

Term	Definition
Constrained decoding	A decoding approach that restricts generation to satisfy a required structure and a legal label set, improving the reliability of structured extraction outputs [23].
Robustness	The stability of extraction behavior under realistic noise and perturbations, such as typos, casing variation, spacing and punctuation changes, and domain-specific formatting differences [34, 35].
Computational efficiency	The resource cost of training and inference, including runtime, memory usage, and throughput. Efficiency is reported to support practical comparison across model families and adaptation strategies [36].
Distribution shift and domain drift	Changes in the input distribution across time, sources, or domains that can alter the surface forms and contexts in which PII appears. These shifts motivate repeated adaptation and staged evaluation [37].
Class-incremental learning	A learning setting in which the label space expands over time, typically through discrete stages that introduce new entity types and additional data [12, 20].
Stage	A discrete step in an incremental protocol. A stage defines the available training data, the active label set, and the evaluation scope at that point in the sequence [20].
Catastrophic forgetting (CF)	Performance degradation on previously learned entity types after adapting a model to new types or new data. Forgetting is assessed by comparing performance on earlier stages before and after adaptation [10, 12].
Knowledge retention	The extent to which a model preserves performance on previously learned entity types after incremental updates. Retention complements the learning of new capabilities when evaluating the stability-plasticity trade-off [12].
Knowledge distillation (KD)	A training strategy that encourages a student model to match the output behavior of a teacher model, often used to preserve performance on earlier categories during updates [38].
Stability-plasticity trade-off	The tension between preserving prior behavior (stability) and acquiring new behavior (plasticity). In this thesis, the trade-off is quantified by jointly measuring the learning of newly introduced types and the retention of previously learned types [12, 13].

2.2 Personally Identifiable Information and PII-Aware NER

PII refers to information that can identify an individual directly or indirectly, either on its own or when combined with other attributes. In operational privacy contexts, this includes identifiers such as names, contact details, government-issued numbers, and account-related identifiers, as well as location-related attributes when they enable linkage to a specific person [1, 2]. In unstructured text, PII appears in diverse surface forms, may be embedded within informal language, and often co-occurs with contextual cues that affect how an extracted span should be interpreted.

PII extraction can be formulated as a specialization of named entity recognition (NER), where the target entity inventory is defined by privacy-relevant types rather than generic categories. Classical NER is typically framed as sequence labeling over tokenized text, with entity boundaries and types predicted under an agreed label space [24]. PII-aware NER inherits this formulation but emphasizes practical reliability for privacy workflows such as redaction, auditing, and compliant data handling. Formally, given an input sentence

$$S = (x_1, x_2, \ldots, x_n),$$

the goal is to recover a set of labeled spans

$$\mathcal{E} = \{(b_i, e_i, y_i)\}_{i=1}^{m},$$

where b_i and e_i denote the begin and end token positions of the i-th extracted span, and $y_i \in \mathcal{Y}$ is its assigned label from a predefined PII type inventory. Under this formulation, valid extraction requires both correct span boundaries and consistent mapping to a well-defined schema, ensuring that results remain comparable across datasets and modeling approaches [3].

Figure 2.1 illustrates this formulation with a compact synthetic example. The sentence contains multiple privacy-relevant spans, each of which must be identified and assigned the correct semantic type. This example also highlights why PII-aware extraction is more than keyword spotting: the system must determine not only which tokens correspond to identifiers, but also which category each span belongs to under the target schema.

PII-aware extraction is closely related to automated de-identification in clinical and administrative text, where protected health information and related identifiers must be detected and removed before downstream use. Neamatullah et al. [39] showed that de-identification is challenging because identifier expressions vary substantially and real-world narratives are noisy. Stubbs et al. [27] further established clinical de-identification as a practical benchmark setting. Dernoncourt et al. [28] later demonstrated that neural approaches improve extraction quality, while boundary errors and rare identifier patterns remain persistent challenges.

In this thesis, we study PII extraction on English text that reflects informal and heterogeneous user-generated content. We build a unified corpus by consolidating public, privacy-oriented resources and normalizing them under a single label schema. Furthermore, we augment coverage through a synthetic component designed to represent rare and edge-case PII patterns without exposing real personal data [19, 40]. This design supports controlled comparisons across modeling paradigms and enables consistent evaluation under the staged settings studied in later chapters.

Sentence:
`John Smith emailed john.smith@example.com from Toronto on 12 May 2024.`

Extracted PII spans:
`(John Smith, NAME), (john.smith@example.com, EMAIL), (Toronto, CITY), (12 May 2024, DATE)`

Figure 2.1: Illustrative synthetic example of PII-aware extraction as labeled span identification. Given an input sentence, the task is to recover all privacy-relevant spans together with their semantic types under the target schema.

Table 2.2 provides a compact snapshot of the unified PII types used in this thesis, categorized by semantic group alongside representative synthetic examples. The full label schema, mapping tables, and IOB2 tag list are provided in Appendix Section B.

Table 2.2: Snapshot of unified PII types grouped by semantic category, with representative synthetic examples.

Semantic Category	**PII Type**	**Example Mention (synthetic)**
Personal Attributes	NAME	John Smith
	GENDER	female
Temporal Data	DATE	12 May 2024
	TIME	3:45 PM
Contact Information	EMAIL	john.smith@example.com
	PHONENUM	+1 000 000 0000
Professional Details	ORGANIZATION	York University
	JOBTITLE	Software Engineer
Government & Financial IDs	SIN	000 000 000
	TAXNUM	TAX-000000
	NATIONALID	ID-00000000
	PASSPORT	P0000000
	DRIVER-LICENCE	DL-0000000
	CREDITCARD	0000 0000 0000 0000
Geographic Data	LOCATION	downtown Toronto
	CITY	Toronto
	STATE	Ontario
	COUNTRY	Canada
	ZIPCODE	M5V 0A0

2.3 Neural Architectures for NER

Neural architectures for named entity recognition (NER) typically model extraction as sequence labeling, where entity boundaries and types are predicted over a tokenized input under a fixed label space [3, 24]. Prior to Transformers, strong neural baselines combined contextual word representations with character-level features and structured decoding, often using a conditional random field (CRF) layer to encourage globally consistent label sequences [4, 5]. Figure 2.2 illustrates a representative sequence labeling architecture that integrates character-level signals with contextual encoding and structured prediction.

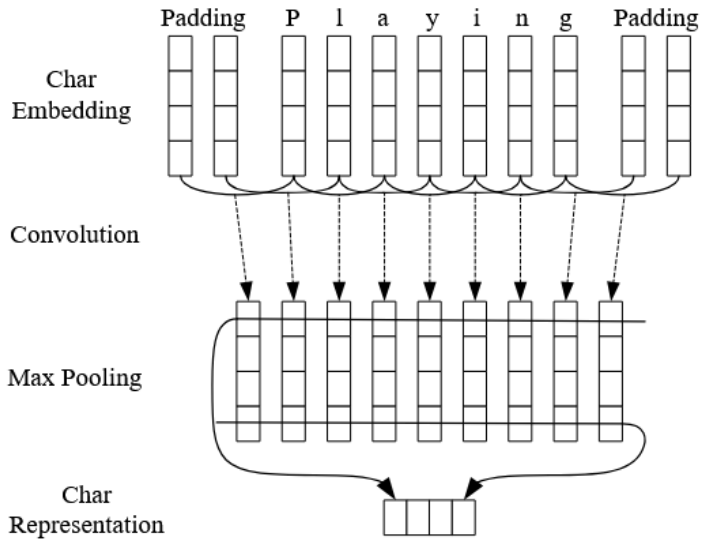

Figure 2.2: A neural sequence labeling architecture for NER that combines character-level features with contextual encoding and CRF decoding. Adapted from [5].

Transformer-based encoders later became the dominant backbone for NER due to their ability to model long-range context using self-attention [6, 7]. Encoder-based models typically attach a token-classification head to contextual token representations and predict one label per token, after which token labels are converted into entity spans for evaluation. This formulation aligns naturally with span-level evaluation because predicted token sequences can be deterministically mapped to spans under schemes such as IOB2 [24].

Figure 2.3 shows the canonical Transformer architecture with an encoder stack and a decoder stack [6]. Encoder-only models for NER correspond to using the encoder stack as the contextualizer, while encoder-decoder models perform text-to-structure generation by conditioning the decoder on encoded representations. Decoder-only models correspond to using a decoder stack with causal self-attention, and they perform extraction through prompted generation, often within instruction-following interfaces [9, 14].

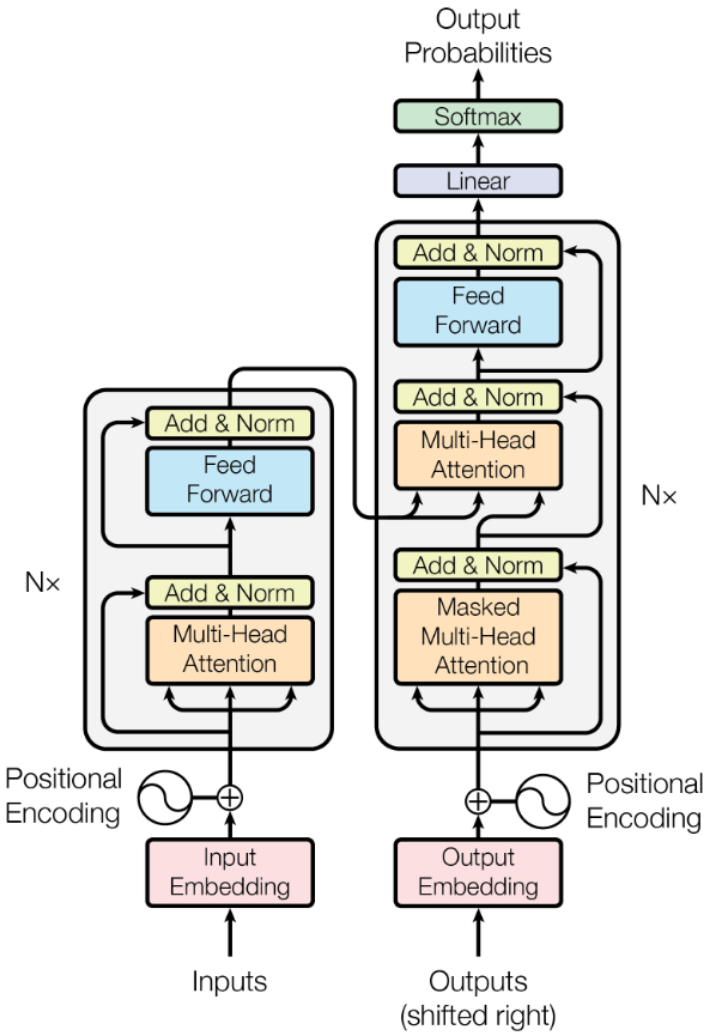

Figure 2.3: The Transformer encoder-decoder architecture. Adapted from [6].

Encoder-decoder architectures provide an alternative formulation in which extraction is treated as structured generation. Instead of predicting token tags, the model generates an output sequence that encodes extracted entities and their types [22, 32, 33].

Decoder-only instruction-following models extend the generative paradigm by performing extraction through prompt templates and in-context examples, optionally combined with parameter-efficient fine-tuning when full fine-tuning is impractical [8, 15, 25, 26]. Because generative approaches may produce malformed or partially structured outputs, later chapters incorporate explicit validity checks and, where applicable, constrained decoding mechanisms to support reliable evaluation [23].

2.4 Fine-Tuning and Catastrophic Forgetting

Supervised fine-tuning adapts a pretrained language model to a downstream task by updating model parameters using labeled examples and a task-specific objective [7]. For token classification, fine-tuning typically optimizes a cross-entropy loss over token labels, while sequence-to-sequence formulations optimize the negative log-likelihood of a target output sequence that encodes the desired structure [32, 33]. Fine-tuning is therefore a general adaptation mechanism that applies across the model families studied in this thesis, even though the task interface and output representation differ.

In operational extraction systems, requirements evolve over time. New PII types become relevant, label inventories expand, and the underlying text distribution shifts across domains. We model this evolution as staged learning, where a model is adapted sequentially across stages that introduce new labels or additional data while previous requirements remain valid [12, 20]. This setting is often referred to as class-incremental learning when the label space grows, making retention a primary concern alongside performance on newly introduced types.

A central challenge in staged fine-tuning is catastrophic forgetting, where adapting a model to new supervision degrades its performance on previously learned categories [10, 13]. This introduces the fundamental stability-plasticity trade-off: stability preserves earlier extraction capabilities, while plasticity enables the acquisition of new ones as the label space and data distributions expand. Prior continual NER studies show that this degradation is not merely theoretical. For example, Zhang et al. report that stronger mitigation can

improve over earlier continual NER baselines by an average of 6.3% in Micro-F1 and 8.0% in Macro-F1, with gains reaching 14.96% and 18.89% in some settings [18]. Improvements of this scale indicate that forgetting and semantic shift can materially distort incremental extraction performance if they are not explicitly addressed.

Several high-level mitigation strategies are commonly used to reduce forgetting. Regularization based methods constrain parameter updates to preserve earlier behavior, for example by penalizing changes to parameters estimated to be important for prior tasks [11]. Distillation-based approaches transfer knowledge from an earlier model checkpoint by matching outputs or intermediate behavior during updates, which can preserve earlier decision boundaries without requiring full access to historical data [38, 41]. Replay-based approaches rehearse a small buffer of earlier examples during later-stage training, often providing strong retention with limited memory [17, 42]. In continual NER, recent work adapts these principles to span and boundary decisions and studies retention strategies tailored to extraction settings [18, 21]. Qualitative evidence also illustrates the nature of forgetting errors. In the examples reported by Zhang et al. [18], earlier methods fragment or partially suppress long entity spans such as *Chinese Mentally Impaired and Blind Sports Associations*, incorrectly assigning internal tokens to unrelated categories or to the non-entity class. Such examples show that forgetting in NER is not only a drop in aggregate F1, but also a failure to preserve coherent boundaries and type consistency for previously learned entities.

Table 2.3: Common strategy families for mitigating catastrophic forgetting.

Strategy Family	**Mechanism**	**Representative Work**
Regularization-based	Constrains parameter updates so that weights important for earlier stages change less, preserving prior behavior without explicitly storing old data.	EWC [11]
Distillation-based	Uses a previous checkpoint as a teacher and adds a loss that encourages the updated model to match prior outputs, helping preserve decision boundaries while learning new supervision.	LwF [41]
Replay-based (rehearsal)	Mixes a small buffer of earlier examples into later-stage training so the model repeatedly rehearses prior patterns while learning new ones.	iCaRL, TinyEM [17, 42]
Background-correction / selective correction	Identifies old-class tokens that are relabeled as background or non-entity in later stages and selectively corrects or reinterprets them to reduce poisoned gradients and semantic drift.	MiB, CPFD [18, 43]

A remaining gap in the literature is that many strong mitigation strategies either rely on storing raw historical examples or apply preservation signals too broadly to isolate the structural source of forgetting in class-incremental NER. EWC slows parameter drift through global weight regularization but does not distinguish which updates are driven by old-entity tokens that have been relabeled as `O`, while distillation uses a teacher-based preservation signal that remains broad relative to the specific tokens responsible for poisoned gradients [11, 41]. The closest conceptual prior is MiB, which identifies the old-class-to-background relabeling problem in class-incremental semantic segmentation and motivates the idea that mislabeled background tokens can be corrected with teacher guidance [43]. However, MiB

operates on independently processed pixels, whereas continual NER introduces additional constraints because the mislabeled positions correspond to entity-bearing token sequences under a structured tagging scheme rather than independent pixels [43]. These limitations are especially important in privacy-oriented NER, where retaining raw historical PII may be undesirable and where effective mitigation must preserve old knowledge without raw-example replay. In this thesis, we address that gap in two steps. First, we evaluate replay and distillation under a unified ASC-PIE protocol. Second, we introduce SPRINT-PP, a privacy-safe mitigation method that combines selective teacher-guided correction of suspected old-entity tokens with prototype-based anchoring and classifier-head stabilization, enabling stronger retention without storing raw historical examples. Later chapters formalize this method and evaluate it against the baseline mitigation strategies.

2.5 In-Context Learning (ICL)

In-context learning (ICL) is a prompting paradigm in which a model performs a task by conditioning on a prompt that includes task instructions and a small set of labeled demonstrations, without updating model parameters [14]. Compared to supervised fine-tuning, ICL enables rapid iteration without retraining, making it particularly attractive for settings where requirements evolve or new label inventories must be supported on demand. Recent analyses also study ICL as a form of implicit learning that emerges from how Transformers process demonstrations in the input context [44].

However, ICL performance is highly sensitive to prompt design choices. The selection of demonstrations and their ordering can significantly affect outcomes, and prompt formats can introduce biases that influence predictions [45, 46]. These sensitivities are especially relevant for extraction tasks, where the model must simultaneously infer the target label semantics and correctly identify boundaries. As a result, practical ICL systems typically standardize prompt templates, control the number of shots under a context-length budget, and apply post-processing to interpret outputs consistently across inputs.

For NER-style extraction, recent work recasts the task as text generation; for example, InstructionNER leverages natural language instructions and auxiliary supervision to improve few-shot performance [8]. Subsequent methods incorporate in-context demonstrations more explicitly within the input to enhance low-resource extraction in a unified generation framework [16]. In parallel, prompt-based large language model approaches transform NER into a constrained generation problem and introduce strategies to reduce spurious extractions, highlighting the importance of validity checks and self-consistency when outputs are generated rather than token-labeled [15].

Despite this progress, the existing ICL literature still leaves important limitations for the problem studied in this thesis. Many ICL-based extraction studies are evaluated on general NER benchmarks, low-resource settings, or model-specific prompting configurations, often with heterogeneous output formats and task assumptions [8, 15, 16]. As a result, their findings are difficult to compare directly across model families and do not fully address the practical requirements of PII extraction, where label inventories may evolve, output validity is critical, and adaptation must remain feasible under resource constraints. In parallel, much of the broader PII and extraction literature continues to rely on parameter-updating approaches such as supervised fine-tuning, which can deliver strong performance but also introduce retraining cost, dependence on labeled data, and repeated model maintenance when schemas change [7, 32, 33]. These characteristics make rapid adaptation more difficult in operational privacy settings.

In this thesis, we therefore use ICL as a standardized parameter-free baseline for PII extraction within the unified ASC-PIE protocol. Rather than treating prompting as an ad hoc or model-specific procedure, we define fixed prompt templates that specify the task, constrain predictions to the unified label schema, and include a controlled number of demonstrations under a consistent context budget. Model outputs are normalized to a shared structured representation and evaluated using the same span-level metrics used for fine-tuned models, with additional validity checks for generative outputs.

This design allows us to assess prompt-based adaptation under the same evaluation framework used throughout the thesis, thereby providing a clearer comparison between parameter-free inference and parameter-updating approaches for PII-aware extraction. The concrete templates and examples used for ICL are reported in the experimental setup and included in the appendices for reproducibility.

Figure 2.4 illustrates the standard structure of an in-context prompting setup for extraction tasks. The prompt is composed of (i) an instruction that specifies the target label inventory, (ii) a set of demonstrations that exemplify the desired input-to-output mapping, and (iii) a new query text for which the model must produce extractions in the same format. This structure aligns with how we operationalize ICL in later chapters: we standardize prompt templates, control the number of shots, and normalize generated outputs to a shared structured representation for evaluation under the unified schema.

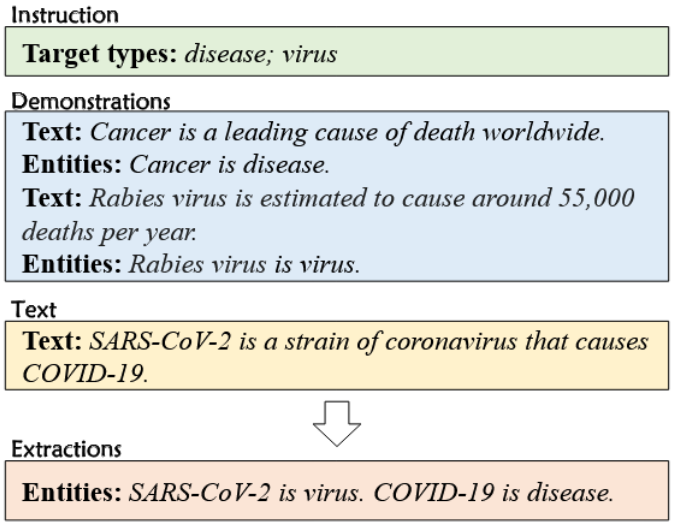

Figure 2.4: Illustration of in-context prompting for extraction. Adapted from [47].

2.6 Evaluation, Validity, and Ethical Considerations (Conceptual Overview)

PII-aware extraction is evaluated not only by whether a system predicts the correct entity types, but also by whether it identifies correct boundaries and produces outputs that are usable in privacy workflows. In this thesis, we prioritize span-level evaluation, where each predicted entity is treated as a single decision defined by its span and type. Span-level precision, recall, and F1 therefore capture both boundary and type correctness directly and are widely used in extraction and de-identification evaluation [24, 27]. In generative settings, models are not required to output explicit character offsets; instead, extracted entity strings are aligned back to the input text during post-processing to recover spans. This strict emphasis on boundary correctness is important for privacy operations such as redaction and auditing.

For generative formulations, correctness alone is insufficient if outputs are not reliably parsable or do not adhere to the allowed label inventory. We therefore distinguish extraction quality from output validity. Output validity refers to whether the produced structure conforms to the required format and uses only legal label names under the unified schema. This distinction is especially important for instruction-following and text-to-structure approaches, where models may produce malformed outputs or hallucinated labels [8, 15]. Validity checks and constrained decoding can reduce such failures [23].

Robustness is another important consideration because real-world text is noisy and varies across domains [34, 35]. We also report practical efficiency indicators, since model families and adaptation strategies differ substantially in training and inference cost [36]. In the incremental setting, retention becomes an additional evaluation dimension, reflecting how well previously learned capabilities are preserved after later-stage updates. Table 2.4 summarizes these evaluation dimensions at a conceptual level.

Ethical considerations remain central because the task concerns sensitive information. We

follow established definitions of PII and treat privacy protection as a primary motivation for extraction [1, 2]. To reduce the risk of exposing personal data, the corpus design emphasizes public resources and synthetic augmentation rather than real personal identifiers, together with transparent dataset documentation practices [48].

Table 2.4: Evaluation dimensions used in this thesis (conceptual overview).

Dimension	**What is assessed**
Extraction quality	Span-level precision, recall, and F1 for entity boundaries and types.
Output validity	Conformance to required structure and legal label names for generative outputs.
Robustness	Sensitivity to realistic perturbations and domain formatting differences.
Efficiency	Training and inference cost indicators (runtime, memory, throughput).
Retention	Performance preservation on earlier stages after incremental updates.

2.7 Summary

This chapter established the core terminology and conceptual background required for the remainder of the thesis. We defined PII-aware extraction as the span-based identification of privacy-relevant entities under a unified label schema, reviewing how tokenization, tagging schemes, and span-level evaluation connect model outputs to entity-level performance. We then outlined the principal modeling paradigms studied in this work, including encoder-based token classification, encoder-decoder structured prediction, and decoder-only instruction-following extraction, highlighting how these families differ in their interfaces, constraints, and typical failure modes.

We also framed the operational challenge of evolving requirements through staged learning and catastrophic forgetting. To address this, we summarized practical mitigation strategies, including regularization, replay, and distillation, and situated them within the broader stability-plasticity trade-off. Finally, we introduced in-context learning as a parameter-free

alternative to fine-tuning and outlined evaluation dimensions beyond standard accuracy, including output validity, robustness, computational efficiency, and the ethical considerations necessary when working with sensitive information.

At the same time, the chapter has highlighted two broader shortcomings in the current literature. First, existing studies often evaluate datasets, model families, learning paradigms, and forgetting mitigation strategies in isolation, with different label inventories, output formats, and evaluation assumptions. This fragmentation makes it difficult to compare results fairly across encoder-based, encoder-decoder, and decoder-only approaches, and it limits the practical interpretation of prior findings for privacy-oriented extraction settings in which label schemas evolve and reliability constraints are strict. Second, continual NER mitigation methods do not yet provide clear guidance for privacy-sensitive deployment settings in which raw historical examples cannot always be retained and prior knowledge must be preserved under staged adaptation.

This thesis addresses these shortcomings through the ASC-PIE framework, which provides a unified schema, a shared evaluation protocol, and a common experimental setting for comparing model families, adaptation strategies, and incremental learning behavior in PII-aware extraction. It further studies staged continual adaptation under a unified protocol by evaluating replay and distillation and by introducing SPRINT-PP as a targeted privacy-safe mitigation method for catastrophic forgetting. Therefore, the concepts introduced in this chapter not only provide background, but also motivate the central contribution of this thesis: a more consistent and practically grounded basis for studying extraction quality, generalization, and retention under evolving privacy requirements. These foundations lead directly to the ASC-PIE framework and the experimental protocol developed in the following chapters.

3 Related Work & Literature Review

This chapter reviews the literature most directly related to the four research questions guiding this thesis. Rather than surveying prior work solely by model architecture or application domain, the discussion is organized around the comparative and methodological problems addressed in this study. This structure enables an evaluation of prior work in terms of what has already been established, what remains insufficiently resolved, and how the present thesis positions itself relative to those gaps.

The chapter therefore proceeds in the same order as the research questions introduced in Chapter 1. First, it reviews work relevant to the supervised comparison of encoder-based, encoder-decoder, and decoder-only extraction models under a shared evaluation setting. Second, it examines prior studies on fine-tuning and in-context prompting, emphasizing whether existing work has compared these adaptation modes under consistent constraints and scoring rules. Third, it surveys research on catastrophic forgetting, class-incremental named entity recognition, and mitigation strategies such as replay and distillation, before positioning the proposed SPRINT-PP method relative to that literature. Finally, it reviews privacy-oriented datasets, NER resources, and synthetic augmentation strategies to clarify how ASC-PIE differs from and builds upon prior datasets.

Throughout the chapter, the emphasis is not only on summarizing representative studies, but also on contrasting them with the scope of this thesis. In each section, the literature is used to identify the limitations that motivate the corresponding part of the ASC-PIE framework, including unified cross-family evaluation, controlled comparison between fine-tuning and prompting, privacy-aware continual adaptation, and consolidated corpus construction with targeted synthetic augmentation.

3.1 Cross-Family Supervised Comparison

3.1.1 Encoder-based NER

A substantial part of the NER literature formulates extraction as token-level sequence labeling. Early neural systems by Lample et al. [4] and Ma and Hovy [5] established strong performance using recurrent architectures with sequence decoding, while Devlin et al. [7] showed that pretrained Transformer encoders can substantially improve token-level labeling under standard fine-tuning. In privacy-oriented settings, Dernoncourt et al. [28] further demonstrated that neural sequence labeling is effective for clinical de-identification, although boundary errors and rare identifiers remain persistent challenges. These studies establish encoder-based tagging as a strong supervised paradigm; however, they primarily evaluate models within the token-classification setting rather than against alternative extraction formulations under shared conditions.

3.1.2 Generative encoder-decoder extraction

A second line of work formulates extraction as constrained sequence generation. Yan et al. [22] recast NER as sequence generation, showing that entity structures can be linearized into textual outputs rather than decoded from token-level tags. Lu et al. [57] generalized this perspective through a unified text-to-structure framework in which multiple extraction tasks are expressed through structured generation under a common interface. More broadly, Raffel et al. [33] and Lewis et al. [32] established the encoder-decoder architectures that make such structured generation practical. This literature shows that encoder-decoder models can perform extraction without relying on tag-based decoding, but the primary emphasis is typically on validating the generative formulation itself, rather than comparing it directly with encoder-based and decoder-only alternatives under a single dataset, schema, and evaluator.

3.1.3 Decoder-only instruction-following extraction

Recent work has also explored decoder-only and instruction-based extraction. Wang et al. [8] showed that instruction-guided formulations can support NER without traditional tag decoders, while Zhang et al. [16] studied instruction tuning for entity extraction with large language models. Brown et al. [14] demonstrated more broadly that large autoregressive models can perform downstream tasks through prompt conditioning, while Ouyang et al. [9] showed that instruction tuning improves structured task execution in generative systems. These developments motivate decoder-only extraction as a realistic supervised and prompted paradigm. However, they also introduce an issue less central to token classification: extraction quality must be considered alongside structural validity and output compliance when the model generates JSON or other constrained formats.

3.1.4 Positioning with respect to RQ1

Taken together, these lines of work show that encoder-based tagging, encoder-decoder structured generation, and decoder-only instruction-following each provide viable supervised extraction paradigms. What remains less common in the literature is a controlled comparison across all three families under the same corpus, label schema, split design, and evaluation protocol. Prior studies often compare models within a single paradigm, or they evaluate different paradigms under different data sources, schemas, and output formats. As a result, it is often difficult to determine whether reported differences arise from the model family itself or from variations in supervision format, schema design, or evaluation criteria.

This limitation is especially important in privacy-oriented extraction, where label inventories vary across datasets and generative models may require explicit validity-aware scoring irrelevant to standard BIO tagging. A fair cross-family comparison therefore requires more than applying several architectures to loosely related data. It requires shared schema normalization, aligned dataset representations, consistent split construction, common entity-level scoring, and explicit treatment of structural validity for generative outputs.

Prior work establishes the individual strengths of encoder-based NER, encoder-decoder structured extraction, and instruction-based generative extraction, but these paradigms are usually evaluated separately or under different data and scoring conditions. In contrast, this thesis addresses **RQ1** by comparing all three model families in a privacy-oriented extraction setting under a unified corpus, a shared 19-type schema, synchronized data representations, and a common validity-aware evaluation protocol. This design makes it possible to attribute observed differences more directly to the modeling paradigm rather than to incompatible dataset construction or evaluation assumptions.

3.2 Fine-Tuning versus In-Context Prompting

3.2.1 Prompt-based extraction and ICL

In-context learning (ICL) treats extraction as a prompting problem rather than a parameter-update problem. Brown et al. [14] showed that large autoregressive models can perform a wide range of downstream tasks by conditioning on instructions and a small number of demonstrations, establishing the broader foundation for prompt-based adaptation. In extraction settings, Wang et al. [8] demonstrated that named entity recognition can be reformulated through explicit instructions rather than conventional tag decoding, while Zhang et al. [16] studied prompt-based and instruction-oriented formulations for few-shot entity extraction. Wang et al. [15] further showed that large language models can perform NER through generative prompting, reinforcing the feasibility of extraction without task-specific retraining.

This literature is significant as it suggests that extraction behavior can be elicited through prompt design alone, especially when the model has already acquired broad linguistic and task-following capabilities during pretraining or instruction tuning. Ouyang et al. [9] showed that instruction tuning can substantially improve the ability of generative models to follow structured task requirements, which is directly relevant for extraction settings requiring machine-readable outputs. However, most prompt-based extraction studies focus on demonstrating that ICL is feasible or competitive in selected settings, rather than systematically comparing it with supervised fine-tuning under the same schema, output constraints, and evaluation protocol.

3.2.2 Factors affecting ICL reliability

Prior work also suggests that ICL performance is highly sensitive to prompt design. The number of demonstrations, the structure of the output format, the clarity of the instruction, and the consistency of example selection can all affect extraction quality. In generative extraction, these factors determine not only whether the correct entity is recovered but also whether the output remains structurally valid. This issue is particularly important in PII extraction, where malformed outputs may be unusable even if they contain partially correct entity content. Studies on instruction-based extraction therefore motivate evaluating not only span recovery but also the reliability with which prompted models satisfy explicit output constraints [8, 15, 16].

Another limitation in the existing literature is that fine-tuning and prompting are often evaluated under different assumptions. Some studies report prompted extraction without an aligned fine-tuned baseline on the same dataset, while others compare models across different prompt formats, label inventories, or post-processing rules. As a result, the literature does not yet provide a sufficiently controlled basis for determining when prompting is genuinely competitive with fine-tuning and when apparent differences instead reflect changes in formatting constraints, demonstration choices, or evaluation design.

3.2.3 Positioning with respect to RQ2

Taken together, prior work establishes that ICL and instruction-based prompting can support extraction without parameter updates and that prompt design plays a substantial role in determining reliability. However, the literature remains limited in two ways that are central to this thesis. First, direct comparisons between fine-tuning and ICL are often not conducted under a single unified schema, shared dataset split, and common evaluator. Second, prompt-based studies do not always separate extraction quality from structural compliance, even though output validity is critical in privacy-oriented extraction workflows.

This thesis addresses **RQ2** by comparing supervised fine-tuning and in-context prompting under a single privacy-oriented schema, aligned test subsets, shared scoring rules, and controlled shot settings, so that the comparison captures not only extraction quality but also validity, structural reliability, and inference cost.

3.3 Catastrophic Forgetting and Mitigation in NER

3.3.1 Continual and class-incremental NER

Catastrophic forgetting has been widely studied in continual learning, where a model must acquire new knowledge without severely degrading on previously learned tasks or classes. De Lange et al. [12] survey this problem across continual-learning settings and show that the central challenge is the stability-plasticity trade-off between preserving prior knowledge and learning new information. In NER, this issue becomes particularly acute because supervision is assigned at the token level, while correctness is ultimately judged at the span level. Monaikul et al. [20] show that sequentially introducing new entity types leads to severe forgetting of previously learned types, even when the model remains effective under static training. This makes class-incremental NER a distinct and difficult continual-learning setting rather than a straightforward transfer of ideas from classification.

The problem is especially natural in privacy-sensitive PII extraction. In realistic deployment settings, new PII categories may become relevant over time, and new annotated data may arrive from different owners, domains, or governance conditions. Under such conditions, continual adaptation is operationally necessary, but retaining historical raw data may be legally or institutionally constrained. Privacy regulations such as the GDPR, PIPEDA, CCPA place explicit restrictions on the retention and reuse of personal data [2]. As a result, class-incremental learning is not only a technical formulation for PII extraction, but also a practically motivated deployment setting.

3.3.2 Replay, distillation, and semantic-shift correction

A common response to catastrophic forgetting is replay. In continual learning more broadly, exemplar-based methods such as iCaRL [17] show that retaining representative past examples can strongly improve retention. In continual NER, Xia et al. [59] likewise show that reviewing synthetic samples can strengthen the preservation of old entity types. Replay remains a strong empirical baseline because it directly reintroduces correct supervision for previously learned classes. However, in privacy-oriented PII extraction, replay is problematic because the retained buffer may contain raw identifiers. This makes replay attractive empirically, but less compatible with privacy-sensitive deployment constraints.

A second family of methods attempts to preserve prior knowledge without storing raw historical examples. Kirkpatrick et al. [11] propose elastic weight consolidation as a parameter-regularization approach, while Li and Hoiem [41] and Hinton et al. [38] motivate teacher-based preservation through learning without forgetting and knowledge distillation. In NER, distillation-based methods remain appealing because they avoid replay buffers, but they also tend to apply preservation signals broadly rather than distinguishing between genuinely new tokens and previously learned entity tokens that have been relabeled as `O`. This limitation is particularly important in staged NER, where forgetting is often driven by mislabeled old spans rather than by the simple absence of old examples.

Recent continual NER work has begun to address this issue more directly. Zheng et al. [58] show that the miscellaneous other-class can play a central role in forgetting, and that continual NER is affected not only by representation drift but also by the way the outside class absorbs old entities during incremental training. This observation is closely related to the central problem addressed in this thesis, namely that old entity tokens relabeled as `O` do not merely fail to reinforce prior knowledge, but actively push the model away from it.

A closely related idea appears in MiB, proposed by Cermelli et al. [43] for class-incremental semantic segmentation. MiB shows that forgetting can also arise when previously learned classes are relabeled as background in later training stages, and it addresses this problem through teacher-based background correction. Although MiB is developed for semantic segmentation rather than NER, it is relevant here because it identifies a structural source of forgetting that goes beyond generic parameter drift and highlights the importance of correcting mislabeled background instances rather than treating all background positions uniformly.

3.3.3 Positioning with respect to RQ3

Taken together, prior work shows that catastrophic forgetting is a serious problem in continual NER, and that replay, distillation, and regularization each provide important mitigation baselines. The literature also suggests that contamination of the outside or background class is a central structural source of forgetting, with MiB offering the closest conceptual formulation of this problem. However, prior methods do not fully address the combination of requirements that arises in privacy-oriented class-incremental NER: explicit handling of old entities relabeled as `O`, preservation of prior knowledge under staged adaptation, and retention support without replaying raw historical PII.

This thesis addresses **RQ3** by evaluating replay and distillation under a shared staged NER protocol and by introducing SPRINT-PP, a privacy-safe mitigation method that combines selective teacher-guided correction of suspected old-entity tokens with prototype-based anchoring and classifier-head stabilization, without raw-example replay. The practical consequence is that replay-based methods may be difficult to justify in production PII extraction systems operating under data-protection regimes such as the GDPR (European Union), PIPEDA (Canada), CCPA (California), and related legislation, because these methods require retaining raw historical examples that may contain personal identifiers.

3.4 PII Datasets, Corpus Design, and Synthetic Augmentation

3.4.1 Privacy-oriented and de-identification corpora

PII-aware extraction depends on labeled corpora in which privacy-relevant spans, such as names, dates, addresses, identifiers, or other sensitive attributes, are explicitly annotated for detection, masking, or de-identification. In clinical and privacy-preserving settings, such corpora are often released only after identifiers are removed, transformed, or governed under controlled access conditions. Neamatullah et al. [39] demonstrated that automated de-identification is feasible but challenging because identifiers vary substantially in form and because residual disclosure risk remains when spans are incompletely removed. Stubbs et al. [27] later provided shared-task resources that helped establish clinical de-identification as a standard evaluation setting, and Dernoncourt et al. [28] showed that neural sequence labeling improves extraction quality while leaving rare identifiers and boundary errors as persistent difficulties.

These corpora are foundational for privacy-oriented extraction because they show that span-level de-identification is both operationally important and technically non-trivial. However, they are typically domain-specific, often governed by strong release constraints, and

usually tied to clinical document styles and schema conventions that differ from the heterogeneous, multi-domain text considered in this thesis. As a result, they are valuable benchmarks, but they do not by themselves provide a general basis for controlled comparison across multiple extraction paradigms in broader PII settings.

3.4.2 Synthetic and community PII resources

Several publicly accessible resources address this limitation through synthetic or semi-synthetic construction. AI4Privacy [19] and Gretel [40] provide large-scale privacy-oriented corpora with diverse PII categories in English text spanning multiple domains. These resources are useful because they support public training and evaluation without directly exposing real personal data. Similarly, nbroad et al. [51] introduced the PII-DD community dataset, which reflects a broader trend toward using instruction-following language models to generate privacy-oriented training material.

This literature shows that synthetic generation can be a practical mechanism for expanding coverage when real data cannot be shared. At the same time, synthetic resources vary in label taxonomies, output format, formatting realism, and the degree of templating. Some emphasize broad coverage, while others emphasize benchmark construction or prompt-generated diversity. These differences make direct aggregation difficult unless explicit schema mapping and normalization rules are defined.

3.4.3 NER resources and schema remapping

Classic NER corpora remain relevant because they provide stable span annotations for entity types that overlap substantially with privacy-oriented extraction, especially people, organizations, and locations. Sang and De Meulder [24] introduced the CoNLL-2003 shared task, which remains a canonical benchmark for sequence labeling. Zihan Wang et al. [50] later proposed CoNLL++ as a resource for robust sequence-labeling evaluation. Although these corpora are not designed around privacy-specific taxonomies, they provide high-quality su-

pervision for core entity categories that can be remapped into privacy-oriented schemas when consistent mapping rules are available.

This possibility is important because it suggests that privacy-oriented extraction datasets need not be constructed only from explicitly privacy-labeled sources. Instead, a broader corpus can be built by combining privacy-focused datasets with NER resources, provided that the resulting label inventory, boundary conventions, and evaluation protocol are carefully unified. This idea is central to the construction of ASC-PIE.

3.4.4 How ASC-PIE differs from prior datasets

The datasets used in this thesis are drawn from this broader landscape, but they are not used in isolation. Instead, they are incorporated into ASC-PIE through schema normalization, controlled preprocessing, and synchronized export across model families. Concretely, ASC-PIE builds on AI4Privacy [19] and Gretel [40] as privacy-oriented sources, a CoNLL++-style NER resource [50], the PII-DD community synthetic resource [51], and thesis-specific synthetic data introduced to improve coverage of rare identifiers and edge cases. This construction addresses several limitations of using any one source in isolation, including inconsistent label taxonomies, narrow domain coverage, differences in supervision format, and limited support for rare or highly structured PII types.

By consolidating these resources under a unified 19-type schema, ASC-PIE provides a more controlled basis for comparing extraction models across paradigms while also broadening coverage beyond what is typically available in a single dataset. For example, Luhn [53] introduced the checksum procedure commonly used to validate credit card numbers, enabling the generation of non-operational values that satisfy basic structural validity constraints. Name-like entities and broader profile-style records are generated using controlled generators whose usage constraints are respected [54]. The resulting corpus therefore builds on existing datasets, but extends them through schema unification, targeted synthetic augmentation, and a shared evaluation design.

For completeness, the literature also contains privacy-oriented datasets that are not used in our experiments but illustrate additional domains and design choices. The BigCode PII dataset [49] targets PII in source code, where surface forms and surrounding context differ substantially from natural language and require code-aware preprocessing and evaluation. Savkin et al. [52] proposed SPY as a synthetic benchmark that studies LLM-driven generation of PII detection examples as a benchmark design choice. We do not include these datasets in our empirical study because ASC-PIE is restricted to English natural-language text drawn from heterogeneous communication and document sources. Incorporating code-centric or benchmark-specific constructions would introduce modality-specific confounds that are not directly comparable under the unified extraction schema and evaluation protocol adopted in this thesis.

Table 3.1 summarizes representative datasets discussed in this section, distinguishing sources used directly in ASC-PIE from additional resources cited for context.

Table 3.1: Datasets discussed in this section, separating sources used directly in ASC-PIE from additional resources cited for context.

Dataset / Source	**Role**	**Notes**
AI4Privacy Open PII Masking [19]	Used	Privacy-oriented labels in assistant-like, multi-domain English text.
Gretel PII Masking EN v1 [40]	Used	Privacy masking with diverse passages and PII/PHI categories across domains.
CoNLL++ [50]	Used	NER-style span annotations supporting core entity types that can be remapped.
PII-DD Mistral generated [51]	Used	Community synthetic dataset generated using an instruction-following LLM.
Thesis synthetic augmentation	Used	Controlled synthetic PII, including rare identifiers and broader common-PII coverage [53, 54].
CoNLL-2003 Shared Task [24]	Reference	Canonical NER benchmark widely used in extraction research.
Clinical de-identification corpora [27, 28, 39]	Reference	Domain-specific de-identification benchmarks motivating span-level evaluation.
BigCode PII dataset [49]	Reference	PII in source code; different surface forms and context than natural language.
SPY synthetic benchmark [52]	Reference	Synthetic benchmark proposal illustrating LLM-driven dataset construction.

ASC-PIE is not a direct intersection of prior datasets, since it is not limited to the labels shared by all sources. It is also not a simple union of existing corpora, because the included datasets differ in annotation style, taxonomy, supervision format, and domain. Instead, ASC-PIE is a curated consolidation and augmentation of multiple privacy-oriented and NER-style resources under a unified 19-type schema. Existing datasets are deterministically remapped, normalized, deduplicated, and exported into aligned representations for different model families, while thesis-specific synthetic data are added to improve coverage of rare, high-structure, and otherwise under-supported PII types.

3.4.5 Positioning with respect to RQ4

Taken together, prior work provides valuable privacy-oriented corpora, synthetic resources, and classical NER datasets, but these resources are usually tied to different label inventories, supervision formats, domains, and benchmarking assumptions. This makes it difficult to study corpus design choices systematically, especially when the goal is to compare multiple model families under a shared protocol. The literature therefore motivates not only the use of existing datasets, but also the need for a unified corpus construction strategy that makes those datasets jointly usable.

This thesis addresses **RQ4** by constructing ASC-PIE as a unified and augmented corpus rather than relying on a single existing dataset. By combining privacy-oriented and NER-style resources under a common schema and adding targeted synthetic augmentation, the thesis makes it possible to study how dataset composition, schema normalization, and synthetic coverage affect extraction performance, especially for rare and structurally difficult PII types.

3.5 Summary

This chapter showed that the literature provides strong individual foundations for PII extraction, prompting, continual learning, and privacy-oriented datasets, but does not yet provide a unified basis for studying these problems under consistent assumptions. Prior work establishes the viability of encoder-based, encoder-decoder, and decoder-only extraction paradigms, yet these families are usually evaluated under different datasets, schemas, and output conventions, limiting direct comparison. Likewise, in-context prompting has emerged as a flexible alternative to supervised fine-tuning, but existing studies rarely compare the two under a shared schema, controlled prompting setup, and common evaluation protocol.

The review also confirmed that catastrophic forgetting is a central challenge in class-incremental NER and that replay, distillation, and regularization provide important mitigation baselines. However, privacy-sensitive settings remain insufficiently addressed, since replay may be undesirable and broader preservation strategies do not explicitly isolate old-entity tokens relabeled as `O` during staged adaptation. At the same time, existing privacy-oriented and NER resources differ substantially in taxonomy, domain, and supervision format, complicating controlled corpus design and cross-paradigm evaluation.

These gaps motivate the central design of this thesis: the ASC-PIE framework as a unified basis for cross-family supervised comparison, controlled fine-tuning versus prompting evaluation, privacy-aware continual adaptation, and schema-aligned corpus construction with targeted synthetic augmentation. These findings lead directly to Chapter 4, which formalizes the framework, staged protocol, adaptation strategies, and evaluation design used in the empirical study.

4 The ASC-PIE Framework

This chapter presents the ASC-PIE framework for evaluating and comparing PII extraction systems under a shared schema and evaluation protocol, together with the ASC-PIE reference corpus used in this thesis as its primary instantiation. The ASC-PIE reference corpus consolidates heterogeneous public resources spanning privacy-oriented masking corpora and general NER supervision, and augments coverage with a purpose-built synthetic component designed to represent rare and highly structured identifiers without using real personal data. The framework is constructed to support principled comparison across model families under a shared schema, common splits, and consistent evaluation criteria, while also enabling staged task growth that reflects how operational PII inventories and deployment conditions evolve over time.

Figure 4.1 presents an abstract view of the ASC-PIE framework as a reusable workflow for evaluating and comparing PII extraction systems under a shared schema and evaluation protocol. The framework begins with the **ASC-PIE Reference Corpus and Schema**, which provide a common benchmark setting for privacy-oriented extraction. Although the framework is instantiated in this thesis using the ASC-PIE corpus, it is not limited to that specific dataset. A user may apply the same framework to another corpus, provided that the data are mapped to the ASC-PIE unified schema and, for encoder-based evaluation, represented with the corresponding BIO-style tagging conventions so that model outputs remain comparable under the same label inventory, task assumptions, and scoring rules.

The next phase is **Model Family Selection**, where the user chooses the extraction paradigm to be evaluated, such as encoder-based token classification, encoder-decoder structured prediction, or decoder-only instruction-following extraction. This is followed by **Adaptation Protocol Selection**, which specifies how the selected model family will be used under the framework. In this thesis, the main protocols are supervised fine-tuning, in-context prompting, and staged continual updating. The **Model Execution** phase then

carries out the corresponding operation: training and inference in supervised fine-tuning settings, prompted inference in in-context learning settings, and sequential stage-wise updating and evaluation in continual-learning settings. Although the resulting outputs differ across paradigms, they are subsequently normalized into a common evaluation space.

These outputs are processed through a **Shared Evaluation** phase, which is the core of the framework. Here, predictions are normalized to a common structured representation and assessed under the same validity-aware and span-level scoring procedure. This shared evaluation design allows differences across model families and adaptation modes to be interpreted under consistent conditions rather than through family-specific post-processing or incompatible metrics. The resulting scores are then used in the **Comparison and Reporting** phase, where models can be ranked, contrasted, and analyzed through per-type results, validity measures, and aggregate comparative summaries.

The framework also includes an optional **Continual Learning Evaluation** branch for settings in which the label space expands over time. In this branch, the selected model is updated sequentially across stages and evaluated for both new-class acquisition and retention of previously learned entity types. The **SPRINT-PP** phase belongs to this staged branch. Its role is to support privacy-aware continual adaptation by identifying suspected old-entity tokens that have been relabeled as `O` in later stages, applying selective teacher-guided correction to those tokens, and anchoring prior knowledge through prototype-based memory and classifier-head stabilization rather than raw replayed examples. The updated model is then evaluated under the same shared procedure, so that forgetting mitigation can be assessed within the same comparative framework as the static extraction experiments.

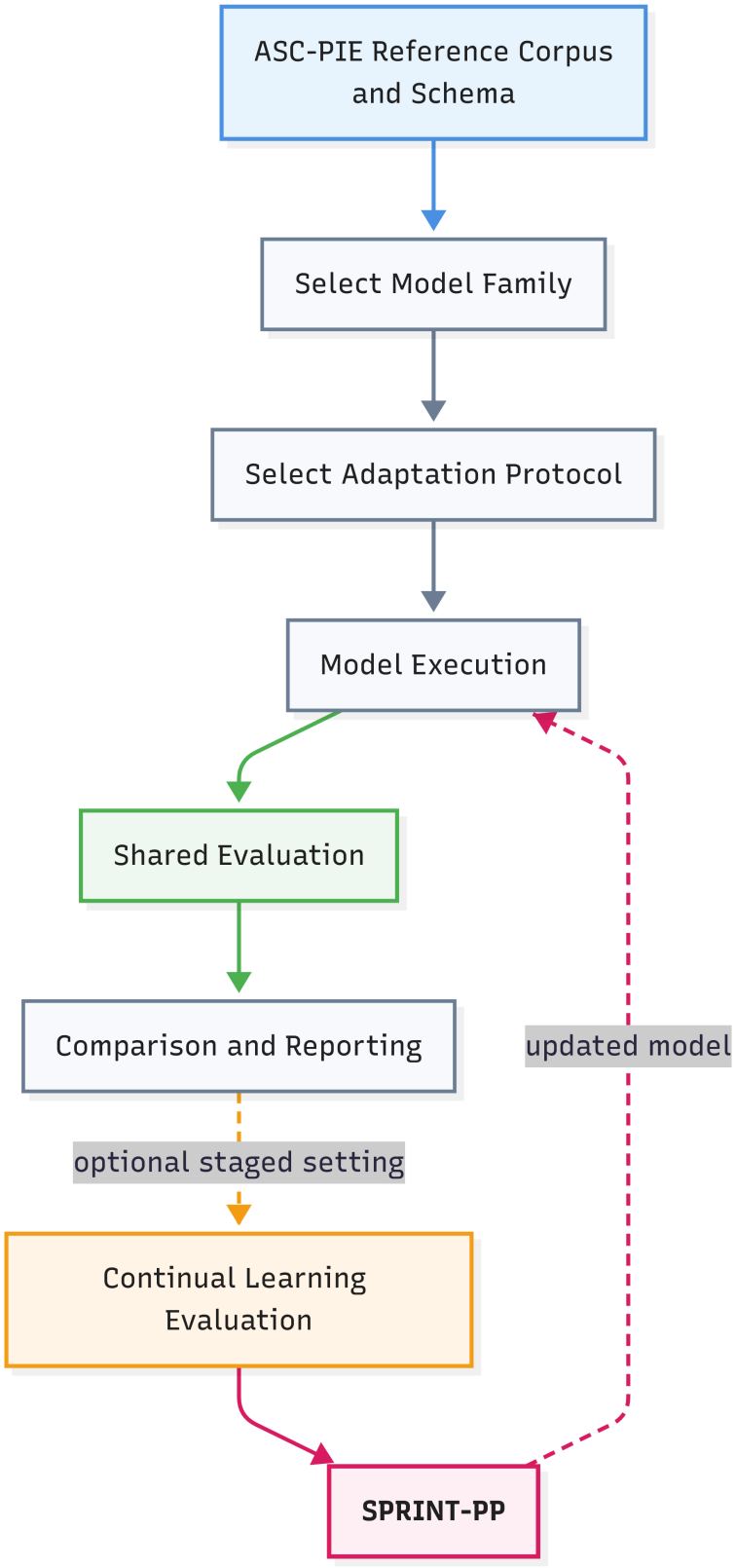

Figure 4.1: Abstract workflow of the ASC-PIE framework, showing schema-aligned input, model-family and adaptation selection, shared evaluation, comparative reporting, and the optional continual-learning branch with SPRINT-PP.

Concretely, the ASC-PIE reference instantiation used in this thesis specifies a unified 19-type label schema, deterministic mapping rules from each source taxonomy, and a consistent preprocessing pipeline with validation checks, BIO consistency repair for token labeling, deduplication, and leakage control across splits. To support cross-paradigm comparison, the corpus is exported in synchronized representations aligned with each family's interface: token-level BIO labeling for encoder-based models, structured extraction targets for encoder-decoder models, and instruction-following examples for decoder-only models. Model outputs are then normalized to a common entity representation and evaluated with a shared harness that distinguishes extraction quality from output validity under both strict and normalized matching.

The remainder of this chapter details the ASC-PIE framework and its reference instantiation in this thesis. We first state the design goals guiding corpus construction, then define the unified label schema, source datasets, and preprocessing rules. Next, we formalize the task formulations and adaptation strategies used for each model family, including supervised fine-tuning, in-context prompting, and the staged continual-learning setting. Finally, we describe the evaluation protocol, reporting criteria, reproducibility practices, and ethical safeguards that govern data handling and analysis in this work.

4.1 Design Goals

ASC-PIE is designed as a reusable framework for evaluating and comparing PII extraction systems under a shared schema, common task interfaces, and a unified evaluation protocol. The workflow in Figure 4.1 is organized to satisfy five design goals. Each goal is reflected in a corresponding part of the framework, from the schema-aligned reference corpus, to model-family and adaptation selection, to shared evaluation, comparative reporting, and the optional continual-learning branch.

G1: Shared schema and benchmark compatibility. The framework should support comparison under a single privacy-oriented label inventory and a consistent task definition. This goal is realized through the *ASC-PIE reference corpus and schema* in Figure 4.1, which provide a common basis for aligning heterogeneous sources and, when needed, for mapping external datasets into the same schema. This supports fair comparison under shared entity definitions.

G2: Cross-family and cross-protocol comparability. The framework should enable meaningful comparison across extraction paradigms and adaptation modes without confounding the results through incompatible interfaces or scoring assumptions. This goal is reflected in the *Select Model Family*, *Select Adaptation Protocol*, and *Shared Evaluation* phases of Figure 4.1.

G3: Coverage of rare and highly structured PII types. The reference instantiation of the framework should provide sufficient coverage for long-tail and structurally constrained identifiers that are often underrepresented in individual public corpora. This goal is supported by the *ASC-PIE reference corpus and schema* component, which combines public resources with thesis-specific synthetic augmentation.

G4: Validity-aware and operationally evaluation. The framework should distinguish between extraction quality and output usability, especially for generative models whose predictions may be structurally malformed even when partially correct. This goal is realized through the *Shared Evaluation* and *Comparison and Reporting* phases of Figure 4.1.

G5: Staged continual-learning evaluation under privacy-sensitive constraints. The framework should support sequential updating when the label space expands over time. This goal is reflected in the optional *Continual Learning Evaluation* branch of Figure 4.1, where models are updated across stages and assessed for both new-class acquisition and retention of previously learned types. Within this branch, replay, distillation, and SPRINT-PP can be evaluated under a shared protocol, with SPRINT-PP specifically designed as a targeted privacy-safe mitigation strategy that avoids raw-example replay.

Taken together, these design goals define the role of ASC-PIE as both a reusable evaluation framework and a reference benchmark instantiation. G1 and G2 provide the foundation for **RQ1** and **RQ2** by enforcing a shared schema, aligned task interfaces, and common evaluation rules across model families and adaptation protocols. G3 primarily supports **RQ4** by improving coverage of rare and highly structured PII types through controlled corpus construction and augmentation. G4 supports **RQ1** and **RQ2** by making output validity and operational reliability explicit dimensions of comparison. G5 directly supports **RQ3** by enabling staged continual-learning evaluation and controlled comparison of forgetting mitigation strategies, including SPRINT-PP.

4.2 ASC-PIE Reference Corpus and Schema

The ASC-PIE reference corpus combines heterogeneous public resources with a thesis-specific synthetic component under a unified 19-type privacy-oriented schema. Its purpose is not only to broaden coverage of privacy-relevant entities but also to make otherwise incompatible sources jointly usable for controlled cross-family comparison. In prior work, privacy-oriented corpora, synthetic resources, and general NER datasets often differ in taxonomy, annotation style, supervision format, and domain coverage, making direct comparison difficult. ASC-PIE addresses this problem through a structured construction process rather than simple dataset collection.

Figure 4.2 summarizes this workflow through four phases. The **Data Sources** phase combines public corpora with synthetic augmentation. The **Schema and Mapping** phase aligns these inputs under the unified ASC-PIE inventory through deterministic mapping and label normalization. The **Corpus Construction** phase then applies cleaning, validation, deduplication, and the construction of fixed train, validation, and test splits. Finally, the **Synchronized Exports** phase produces aligned views for encoder-based, encoder-decoder, and decoder-only models. Together, these phases ensure that later experiments can vary the

modeling paradigm without changing the underlying corpus content, label inventory, or split structure.

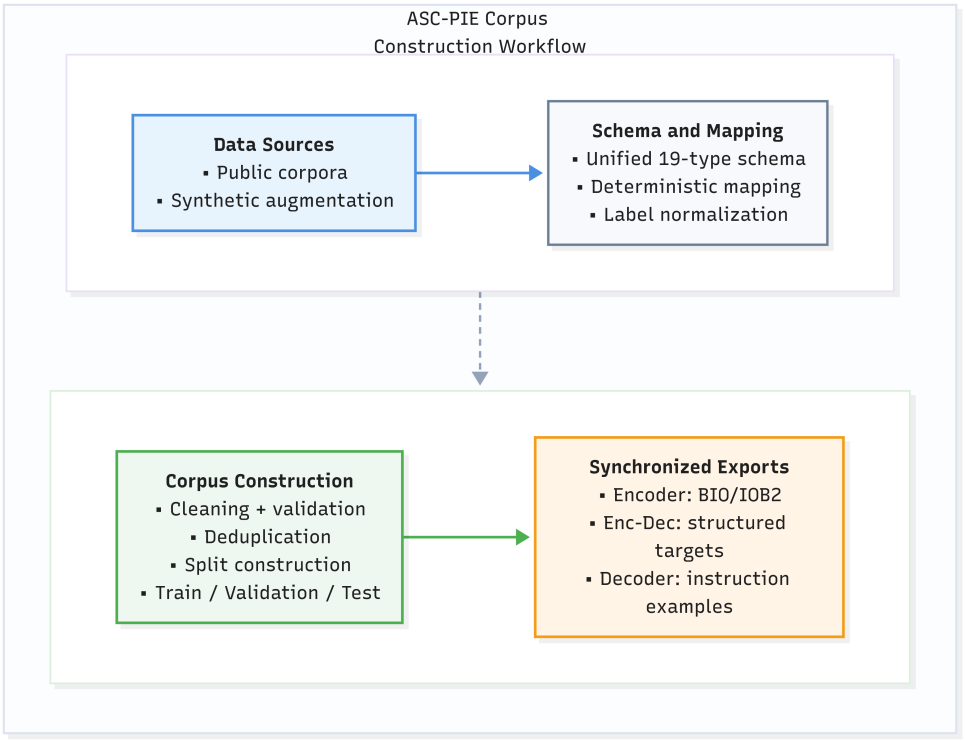

Figure 4.2: ASC-PIE corpus construction workflow, organized into data sources, schema and mapping, corpus construction, and synchronized exports.

The discussion below follows the same four phases shown in Figure 4.2: **Data Sources**, **Schema and Mapping**, **Corpus Construction**, and **Synchronized Exports**. After presenting these stages, the subsection concludes with descriptive statistics that characterize the resulting ASC-PIE corpus.

A central principle of ASC-PIE is that corpus construction is treated as schema-guided harmonization rather than simple dataset aggregation. Heterogeneous supervision formats, label taxonomies, and domain conventions are reconciled before evaluation, ensuring that subsequent differences across model families and adaptation protocols can be interpreted primarily as modeling differences rather than as artifacts of incompatible source annotations or inconsistent preprocessing. In this sense, the corpus functions as a controlled reference instantiation of the broader ASC-PIE framework.

4.2.1 Data Sources

ASC-PIE draws on both privacy-oriented masking corpora and NER-style span corpora. Privacy-oriented sources provide diverse PII categories in assistant-like and multi-domain English text, while NER corpora provide high-quality supervision for core entity categories that overlap with common PII. The primary public resources used in this thesis include AI4Privacy and Gretel for privacy masking [19, 40], CoNLL++ for NER-style spans [50], and community synthetic PII corpora derived from instruction-following generation [51]. In addition, ASC-PIE incorporates a thesis-specific synthetic component designed to improve coverage of rare and high-structure identifiers without using real personal data. This component includes both controlled template-based generation for structured identifiers and a large synthetic tabular resource of approximately `100,000` rows constructed using Fake Name Generator, which provides broad coverage of common PII fields under its stated usage constraints [54]. For example, structured identifiers such as credit card numbers can be generated using non-operational values together with validity rules such as the Luhn checksum [53], while name-like and profile-style entities can be produced through controlled synthetic generation and then normalized to the unified ASC-PIE schema.

4.2.2 Schema and Mapping

Unified Label Schema. ASC-PIE adopts a unified, privacy-oriented label schema consisting of 19 PII entity types. This schema serves as the common semantic layer of the framework and is used consistently throughout corpus construction, prompting, model adaptation, and evaluation. It is designed to cover common identifiers that arise in operational text-processing pipelines, to support deterministic remapping from heterogeneous public datasets with differing taxonomies, and to remain compatible with both token-level sequence labeling and structured generation interfaces. The canonical type set used throughout the experiments is: `NAME, GENDER, DATE, TIME, EMAIL, PHONENUM, SIN, TAXNUM, NATIONALID, PASSPORT, DRIVER-LICENCE, CREDITCARD, LOCATION, CITY, STATE, COUNTRY, ZIPCODE,`

JOBTITLE, ORGANIZATION.

For encoder-based models, ASC-PIE represents this schema using IOB2 tagging, with O for non-entity tokens, B- tags for all 19 entity types, and I- tags only for types modeled as potentially multi-token entities. In the implemented label map, GENDER, NATIONALID, and PASSPORT omit I- tags because they are realized predominantly as single-token entities in the constructed corpus. This encoder-specific inventory preserves standard token-classification training while ensuring deterministic conversion from token tags to entity spans during evaluation. For generative model families, by contrast, the same canonical type names are used directly in structured outputs. This shared type inventory allows predictions from all model families to be normalized into a common entity representation and compared under the same evaluation protocol.

Illustrative example. To make the schema more concrete, consider the following normalized ASC-PIE entry:

> **Text:** Account emails arrive at KristenJSmith@dayrep.com . I am Kristen Smith and I live in Barrie Ontario . Weekly tasks as a Linguistic anthropologist at Matrix Design . Roots are in Barrie Ontario . Payment card 5310 6911 6261 4190 expires 27 - Aug sin 688335389 . Updates can go to KristenJSmith@dayrep.com . National ID is 618 449 813 . Leave a message on 705 - 728 - 6948 .
>
> **Canonical entities:** EMAIL: KristenJSmith@dayrep.com; NAME: Kristen Smith; CITY: Barrie; STATE: Ontario; JOBTITLE: Linguistic anthropologist; ORGANIZATION: Matrix Design; CITY: Barrie; STATE: Ontario; CREDITCARD: 5310 6911 6261 4190; DATE: 27 - Aug; SIN: 688335389; EMAIL: KristenJSmith@dayrep.com; NATIONALID: 618 449 813; PHONENUM: 705 - 728 - 6948.

This example illustrates how a single ASC-PIE entry is normalized under the unified schema. The same canonical type inventory is then reused across the encoder, encoder-decoder, and decoder-only representations, even though each family consumes supervision in a different form.

Because ASC-PIE consolidates sources with different taxonomies and naming conventions, each source is mapped to the canonical schema through deterministic preprocessing rules. Equivalent type strings are also normalized across sources and model outputs. For example, the schema uses `DRIVER-LICENCE` as the canonical type name, while the evaluation pipeline maps harmless aliases such as `DRIVERLICENCE` to that form to avoid type-string artifacts during cross-family comparison.

Normalization and Preprocessing overview. After source acquisition, each dataset passes through a common normalization pipeline that converts its original annotations and label taxonomy into the ASC-PIE schema and the aligned representations used by the supported model families. In practice, the normalization pipeline includes source-specific ingestion, schema mapping, token-level normalization, and consistency repair before the data are passed to later corpus-construction and export stages. Table 4.1 summarizes the role of each source, its original annotation format, and the normalization procedure applied before integration into the unified corpus.

Table 4.1: ASC-PIE source ingestion and normalization strategies.

Source	**Original annotation format**	**Normalization approach**
AI4Privacy [19]	Span offsets and types	Deterministic schema mapping; convert spans to token-level IOB2 tags via token-offset overlap.
Gretel [40]	Entity strings and types	Deterministic schema mapping; recover mentions in text via string matching and map to token-level IOB2 tags.
CoNLL++ [50]	Token-level BIO tags	Remap entity types to privacy schema; retain token-level BIO representation.
PII-DD (community synthetic) [51]	Tokens and labels	Remap labels to the unified schema; retain token-level BIO representation.
Thesis synthetic augmentation [54]	Controlled templates and Fake Name Generator-based synthetic records	Generate rare and high-structure identifiers together with broad-coverage common PII records; normalize to the unified schema and export to all representations.

The source-specific preprocessing rules summarized in Table 4.1 convert heterogeneous annotation formats into the canonical ASC-PIE schema. In general, source labels are deterministically mapped to the closest matching ASC-PIE types, while mentions outside the target inventory are excluded by mapping them to `O`. This procedure ensures that all model families are trained and evaluated against the same label set while avoiding the introduction of source-specific artifacts that do not belong to the unified schema.

Tokenization and BIO construction. ASC-PIE uses token-level IOB2 supervision for encoder models. For sources with span offsets, we tokenize the text and assign `B-` and `I-` tags based on character overlap. If offsets are missing, we first recover mentions via deterministic string matching before applying this labeling. Finally, a consistency repair pass is applied to ensure well-formed IOB2 sequences, correcting illegal transitions such as isolated `I-` tags.

4.2.3 Corpus Construction

After per-source normalization, all samples are combined into a single corpus with source metadata retained for later analysis. To reduce redundancy and prevent cross-split leakage, ASC-PIE applies deduplication before constructing the final train, validation, and test partitions. The resulting corpus contains `333,109` examples in total, with fixed and disjoint splits of `312,085` for training, `10,512` for validation, and `10,512` for testing, as summarized in Table 4.2. These splits are reused across all model families and adaptation settings so that performance differences reflect modeling and training choices rather than variation in data partitioning.

Table 4.2: ASC-PIE fixed split sizes used across all experiments.

Split	**Number of examples**
Train	`312,085`
Validation	`10,512`
Test	`10,512`
Total	`333,109`

4.2.4 Synchronized Exports

To support cross-family comparison under a shared protocol, ASC-PIE is exported into three synchronized representations aligned with the modeling interface of each family while preserving the same underlying examples and split membership. Table 4.3 summarizes these representations. This design allows encoder, encoder–decoder, and decoder-only models to operate under family-appropriate supervision while remaining comparable through the shared evaluation harness.

Table 4.3: ASC-PIE dataset representations used across model families.

Representation	**Format**	**Key contents**
Encoder (token labeling)	Hugging Face DatasetDict	Word tokens, token-level labels under the unified IOB2 schema, and source metadata.
Encoder–decoder (structured extraction)	CSV	Instruction-style input text paired with constrained target extraction strings, together with raw text and source metadata.
Decoder-only (instruction following)	JSONL	Canonical text together with gold entity annotations including type, text, and character offsets, plus source metadata.

In practice, the encoder representation supports token classification through word-level IOB2 supervision, the encoder–decoder representation supports constrained key–value generation, and the decoder-only representation supports strict JSON-based instruction following. Together, these aligned exports allow the same corpus to be reused across model families without changing the underlying data split or entity schema.

4.2.5 Corpus Characteristics

Label distribution. ASC-PIE contains `2,025,878` entity mentions across all splits. The label distribution is intentionally imbalanced, reflecting realistic long-tail behavior in privacy inventories. High-support categories include `NAME` (`445,477`), `EMAIL` (`240,384`), and `DATE` (`221,217`), while rare categories such as `COUNTRY` (`914`) and `TAXNUM` (`3,869`) remain much lower-support. Figure 4.3 summarizes the resulting entity counts by type.

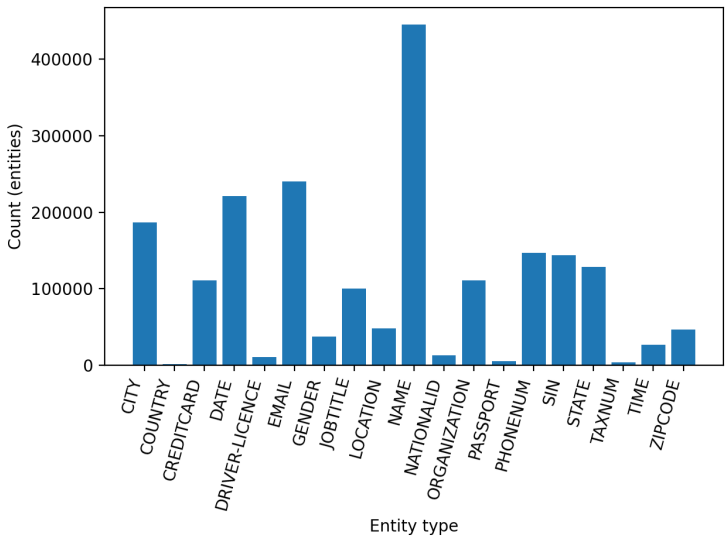

Figure 4.3: ASC-PIE entity counts by type (all splits combined).

Split stability. The label distribution remains broadly stable across the fixed train, validation, and test splits. As shown in Figure 4.4, the training split naturally dominates in absolute count because it is much larger, but the relative profile of entity types is preserved across all three partitions. High-support categories such as `NAME`, `EMAIL`, `DATE`, `CITY`, and `PHONENUM` remain consistently prominent, while lower-support categories such as `COUNTRY`, `TAXNUM`, `DRIVER-LICENCE`, and `PASSPORT` remain comparatively rare in each split. This indicates that the split construction procedure preserved the long-tail structure of the corpus rather than introducing major distortions between training and held-out data. Such stability is important for fair evaluation, because it ensures that later performance differences are less likely to be artifacts of split-specific label imbalance and more likely to reflect genuine modeling differences under a representative privacy-oriented extraction setting.

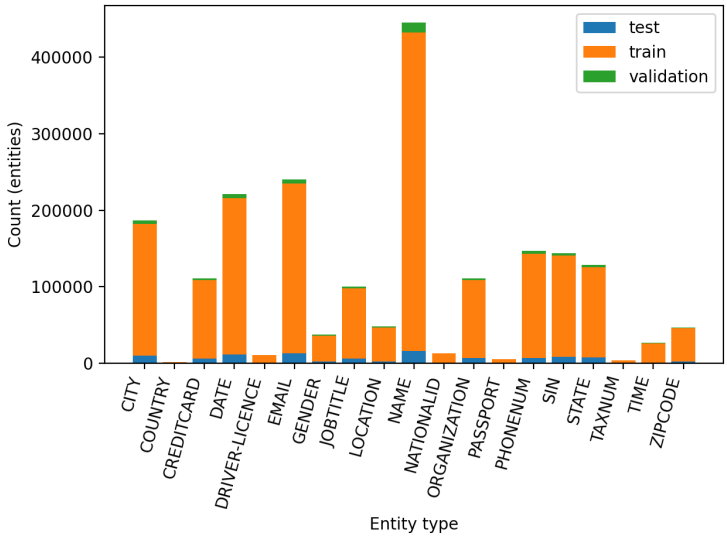

Figure 4.4: ASC-PIE entity-type counts by split.

Source composition. The final training corpus is dominated by privacy-oriented resources and the thesis synthetic component, while classic NER supervision and community synthetic corpora provide smaller but important contributions. This composition is shown in Figure 4.5, which highlights that ASC-PIE is not sourced from a single dataset but from a deliberately consolidated mixture of public privacy corpora, remapped NER data, and synthetic augmentation.

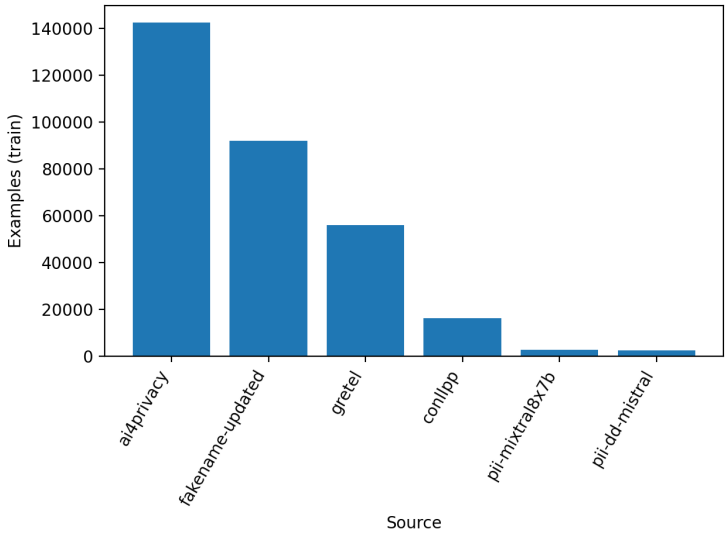

Figure 4.5: Training-set example counts by source for ASC-PIE.

Entity density per example. At the example level, ASC-PIE contains approximately 6.08 entities per example on average. However, this average hides substantial variation, as some examples contain only one or two entities while others contain dense clusters of identifiers. Figure 4.6 visualizes this distribution and illustrates that the corpus includes both simple and highly populated extraction cases.

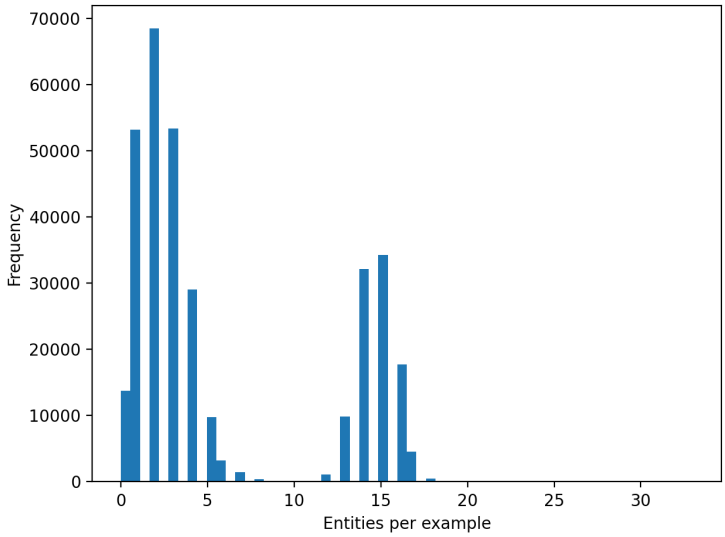

Figure 4.6: Distribution of entity counts per example in ASC-PIE.

Sequence length characteristics. Token-length statistics computed over all splits show a mean of 50.24 encoder tokens per example and a median of 23, indicating that many ASC-PIE instances are relatively short. At the same time, the corpus exhibits a pronounced long tail, with sequences reaching up to 1,846 tokens and a 99th percentile of 757. As shown in Figure 4.7, most examples are relatively short, but the corpus also includes a smaller number of substantially longer inputs. This right-skewed distribution is important for the design of the framework because it motivates the maximum sequence-length controls adopted later for family-specific training and inference, while also showing that ASC-PIE is not limited to short snippets and includes more demanding long-context extraction cases.

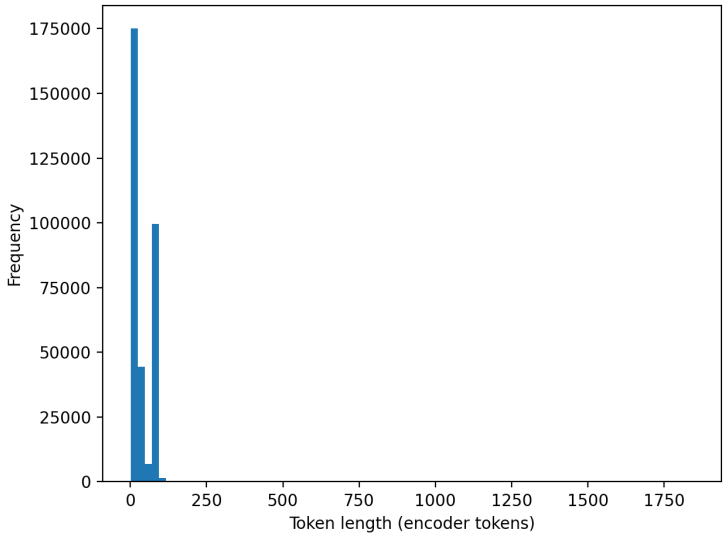

Figure 4.7: Histogram of ASC-PIE token lengths in the encoder representation.

Implications for modeling. Taken together, the summary statistics in Table 4.4 show that ASC-PIE is large, heterogeneous, and intentionally imbalanced. The synchronized multi-representation design supports fair cross-family comparison, while the long-tail label distribution, variable entity density, and right-skewed sequence lengths create a realistic extraction setting that is challenging for both supervised adaptation and continual learning.

Table 4.4: ASC-PIE Corpus summary statistics.

Property	**Value**
Canonical entity types	19
Source datasets / components	5
Supported representations	3 (encoder, encoder–decoder, decoder-only)
Total examples	333,109
Train / validation / test	312,085 / 10,512 / 10,512
Total entity mentions	2,025,878
Average entities per example	6.08
Mean token length	50.24
Median token length	23
Maximum token length	1,846
Highest-support type	NAME (445,477)
Lowest-support type	COUNTRY (914)

4.3 Model Families and Task Formulations

ASC-PIE evaluates three model families that operationalize PII extraction through different task interfaces: encoder-based token classification, encoder-decoder structured prediction, and decoder-only instruction-following extraction. These families differ in how supervision is expressed, how outputs are produced, and which types of prediction errors or structural violations arise in practice. To support fair comparison, ASC-PIE defines a family-specific task formulation for each paradigm while keeping the underlying label schema and the train, validation, and test splits fixed.

Despite these interface differences, all families are evaluated through a shared entity-level representation,

$$E = \{(\tau_m, x_m)\}_{m=1}^{M}, \tag{1}$$

where τ_m is the canonical ASC-PIE entity type and x_m is the extracted text span. Encoder outputs are reconstructed from IOB2 tag sequences, whereas generative outputs are parsed from constrained structured predictions. The family-specific subsections below describe how each paradigm produces outputs that are converted into this common representation. The shared scoring procedure, including strict and normalized matching, type canonicalization, and output-validity checks, is defined in Section 4.5.

4.3.1 Encoder Models (Token Classification)

Encoder-based models formulate PII extraction as token-level sequence labeling under the unified ASC-PIE schema. Given a tokenized input sequence, the model predicts an IOB2 tag for each token, where entity types are encoded as `B-TYPE` and `I-TYPE` tags and non-entity tokens are labeled `O`. In our implementation, a pretrained Transformer encoder is augmented with a token classification head that maps contextual token representations to the ASC-PIE tag set, following the standard fine-tuning paradigm established for BERT-style token-level prediction. Under the exported encoder label map, this formulation yields 36 labels in total,

including `O`; `GENDER`, `NATIONALID`, and `PASSPORT` omit `I-` tags because they are realized predominantly as single-token entities in the constructed corpus.

Illustrative example. Using the running example introduced in the corpus construction subsection, the encoder input is the word-token sequence itself, while the target output is an IOB2 label for each word token. For example, a portion of the sentence may be represented as

> **Tokens:** `[Account, emails, arrive, at, KristenJSmith@dayrep.com, ., I, am, Kristen, Smith, and, I, live, in, Barrie, Ontario, ., ...]`
>
> **IOB2 labels:** `[O, O, O, O, B-EMAIL, O, O, O, B-NAME, I-NAME, O, O, O, O, B-CITY, B-STATE, O, ...]`

Input representation and label alignment. ASC-PIE stores encoder supervision at the word-token level. Let $W = [w_1, w_2, \ldots, w_n]$ denote the input sequence of word tokens, and let $Y = [y_1, y_2, \ldots, y_n]$ denote the corresponding word-level IOB2 label sequence. Because the encoder backbones used in this thesis operate on subword tokenization rather than raw word tokens, these word-level annotations must be aligned to the subword sequences consumed by the model. This step is required by the architecture itself, although it is especially relevant for PII extraction because identifiers such as emails, document numbers, and hyphenated forms are often segmented into multiple subword pieces. After tokenization, each word w_i is mapped by a tokenizer function f to a sequence of k_i subword pieces, such that $f(w_i) = [s_{i,1}, s_{i,2}, \ldots, s_{i,k_i}]$. Concatenating these outputs over all words yields the full subword sequence $S = [s_{1,1}, \ldots, s_{1,k_1}, s_{2,1}, \ldots, s_{n,k_n}]$.

To preserve compatibility with the pretrained encoder tokenizers used in BERT-style token classification while maintaining word-level supervision, labels are assigned only to the first subword of each word. This follows the standard practice in encoder-based NER, where the representation of the first subtoken is used for sequence labeling. Formally, for each word w_i that is tokenized into subwords $[s_{i,1}, \ldots, s_{i,k_i}]$, the aligned subword-level label assignment

is defined as

$$\hat{y}_{i,j} = \begin{cases} y_i, & \text{if } j = 1, \\ \varnothing, & \text{if } j > 1, \end{cases} \tag{2}$$

where $\varnothing$ denotes the ignore index used during training, implemented as `-100` in practice. As a result, non-initial subwords do not contribute to the training loss.

Training objective. Given the subword sequence S, the encoder produces contextual representations

$$h_t = \text{Encoder}(S)_t, \qquad t = 1, \ldots, |S|. \tag{3}$$

A token classification layer then maps each representation to label logits, $z_t = W_c h_t + b_c$, followed by a softmax distribution over the ASC-PIE tag set, $p(y_t \mid S) = \text{softmax}(z_t)$.

Encoder models are trained using supervised fine-tuning with a weighted token-level cross-entropy objective:

$$\mathcal{L}_{\text{enc}} = -\sum_{t \in \mathcal{M}} \alpha_{\hat{y}_t} \log p(\hat{y}_t \mid S), \tag{4}$$

where $\mathcal{M}$ is the set of supervised token positions, that is, the first subword of each word, and $\alpha_{\hat{y}_t}$ is the class weight associated with label $\hat{y}_t$. The underlying fine-tuning setup follows the standard encoder-based token-classification formulation used for the pretrained encoder backbones in this thesis, while the use of class-dependent weighting is motivated by the broader literature on learning under class imbalance and long-tailed label distributions. In our implementation, these weights are derived from label frequencies in the training split so that lower-support entity types receive greater emphasis during optimization. The exact inverse-frequency weighting heuristic and encoder training schedule used in the experiments are reported in Appendix A.

Span reconstruction for shared evaluation. At inference time, the encoder predicts a word-level IOB2 tag sequence $\tilde{Y} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n]$ over the input token sequence. This sequence is scanned from left to right and converted into entity spans using standard IOB2 reconstruction rules: a new entity begins at a `B-TYPE` label and is extended by subsequent contiguous `I-TYPE` labels of the same type. The corresponding entity text is reconstructed from the original word tokens covered by that span. For example, the prediction sequence `[B-NAME Kristen, I-NAME Smith, O, O, B-CITY Barrie, B-STATE Ontario]` yields the entity pairs `(NAME, Kristen Smith)`, `(CITY, Barrie)`, and `(STATE, Ontario)`. The resulting entity set is then expressed in the shared representation defined in Section 4.5, enabling direct comparison with generative approaches.

Typical extraction errors. Encoder-based taggers commonly exhibit boundary-related errors in sequence labeling settings. These errors often appear as boundary fragmentation, where a multi-token entity is recovered only partially, or boundary drift, where adjacent punctuation or nearby context words are incorrectly absorbed into the predicted span. In addition, encoder-based NER systems may produce type confusion among semantically related categories or categories with overlapping surface forms.

4.3.2 Encoder-Decoder Models (Structured Prediction)

Encoder-decoder models formulate PII extraction as the conditional generation of a structured output sequence. Given an input text X, the model generates a target string Y that encodes the extracted entities and their types under the unified ASC-PIE schema. This formulation follows the standard sequence-to-sequence view of conditional generation, aligning well with text-to-text modeling frameworks such as T5 and BART. In this thesis, the encoder-decoder family is instantiated with two representative backbones, FLAN-T5-base and BART-base, under the same ASC-PIE extraction interface.

Illustrative example. Unlike the encoder formulation, the encoder-decoder setup in ASC-PIE does not predict IOB2 tags. Instead, it consumes an instruction-style input and

generates a structured extraction string. Using the same running example, the source side contains the input text together with the extraction instruction, while the target side is a constrained key-value sequence such as

> **Target output:** `EMAIL: KristenJSmith@dayrep.com; NAME: Kristen Smith; CITY: Barrie; STATE: Ontario; JOBTITLE: Linguistic anthropologist; ORGANIZATION: Matrix Design; CITY: Barrie; STATE: Ontario; CREDITCARD: 5310 6911 6261 4190; DATE: 27 - Aug; SIN: 688335389; EMAIL: KristenJSmith@dayrep.com; NATIONALID: 618 449 813; PHONENUM: 705 - 728 - 6948.`

This formulation preserves the same entity inventory as the encoder setting, but expresses supervision as conditional generation rather than token-level labeling.

Input prompting and constrained output format. For ASC-PIE, we use an instruction style input that specifies the extraction task, enumerates the allowed entity types, and provides an explicit output format. Targets are expressed as a semicolon-separated list of key-value pairs: `TYPE: value; TYPE: value; ...` where `TYPE` must be one of the canonical ASC-PIE labels and `value` is the extracted entity surface form. If no entities are present, the target is the fixed string `No PII found.` Formally, for an input X, the target sequence can be written as $Y = [y_1, y_2, \ldots, y_T]$, where each token y_t belongs to the output vocabulary and the full sequence encodes the ordered structured extraction result. The use of a structured output string is consistent with prior work that casts information extraction and NER as text generation rather than token classification.In the experiments, both source and target sequences are truncated to a maximum length of 512 tokens in order to bound memory usage while accommodating heterogeneous texts and multi-entity outputs.

Training objective and generation. Models are trained via supervised fine-tuning with teacher forcing to maximize the conditional likelihood of the target sequence given the input, following the standard encoder-decoder sequence generation objective. The conditional generation objective is

$$P(Y \mid X) = \prod_{t=1}^{T} P(y_t \mid y_{<t}, X), \tag{5}$$

and the corresponding negative log-likelihood loss is

$$\mathcal{L}_{\text{seq2seq}} = -\sum_{t=1}^{T} \log P(y_t^* \mid y_{<t}^*, X), \tag{6}$$

where y_t^* denotes the gold output token at position t. At inference time, the model generates an output sequence $\hat{Y} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{\hat{T}}]$, under fixed decoding settings, and the resulting text is parsed into a list of extracted entities. Because generation can introduce formatting deviations, ASC-PIE evaluates both extraction quality and output validity, where validity refers to compliance with the required structure and use of allowed label names.

Output parsing for shared evaluation. Generated encoder-decoder outputs are parsed with a strict key-value parser that recovers entity pairs from the predicted structured sequence. When parsing succeeds, the output is converted into the shared entity representation defined in Section 4.5. Outputs that do not satisfy the required key-value structure are treated as invalid under the shared evaluation protocol. This design allows encoder-decoder predictions to be compared directly with encoder and decoder-only outputs under a common entity-level representation.

Typical extraction errors. Encoder-decoder extraction can fail through omission, where entities present in the input are not generated; this behavior appears as a recall error in generation-based extraction settings. It can also fail through hallucination, where unsupported entities or labels are generated. Structural errors are another failure mode, arising when the output violates the required key-value format or uses unsupported labels. These behaviors motivate reporting validity rates in addition to extraction accuracy under the shared evaluation protocol.

4.3.3 Decoder-Only Models (Instruction-Following Extraction)

Decoder-only models perform PII extraction through autoregressive generation conditioned on an instruction prompt. In contrast to encoder-only taggers and encoder-decoder structured predictors, decoder-only instruction-following models treat extraction as a response generation task, where the model must both follow formatting constraints and recover entity mentions from the input text. This paradigm supports parameter-free inference through prompting and parameter-efficient adaptation through supervised fine-tuning. In this thesis, the decoder-only family is instantiated with two open-weight instruction-tuned backbones, `Qwen2.5-7B-Instruct` and `Meta-Llama-3.1-8B-Instruct`.

Illustrative example. Using the same running example, the decoder-only formulation presents the model with an instruction prompt and requires a strict JSON response rather than an IOB2 sequence or a semicolon-separated target string. For the input text above, the expected output has the form


```
{"entities":[ {"type":"EMAIL","text":"KristenJSmith@dayrep.com"},
{"type":"NAME","text":"Kristen Smith"}, {"type":"CITY","text":"Barrie"},
{"type":"STATE","text":"Ontario"},
{"type":"JOBTITLE","text":"Linguistic anthropologist"},
{"type":"ORGANIZATION","text":"Matrix Design"},
{"type":"CREDITCARD","text":"5310 6911 6261 4190"},
{"type":"DATE","text":"27 - Aug"}, {"type":"SIN","text":"688335389"},
{"type":"NATIONALID","text":"618 449 813"},
{"type":"PHONENUM","text":"705 - 728 - 6948"} ]}
```


This example highlights the central requirement of the decoder-only setting: the model must recover the correct entities while also satisfying the required machine-readable output schema.

Prompting interface and structured JSON outputs. ASC-PIE adopts a strict, machine-readable JSON output schema for decoder-only models:

```
{"entities": [{"type": "...", "text": "..."}, ...]}
```

The system prompt specifies the allowed ASC-PIE entity types and requires the model to output JSON only, with an empty list when no entities are present. The aligned decoder representation stores a canonical input text together with gold entity annotations including `type`, `text`, and character offsets, although supervised generation predicts only `type` and `text`. Let X denote the full prompt, including the system instruction and the user-provided input text, and let $Y = [y_1, y_2, \ldots, y_T]$, denote the target response sequence representing the required JSON output.

Under this formulation, the model generates the response autoregressively according to

$$P(Y \mid X) = \prod_{t=1}^{T} P(y_t \mid X, y_{<t}), \tag{7}$$

which is the standard causal language modeling factorization used in decoder-only large language models. This design aligns evaluation with operational needs, as downstream privacy workflows require reliable structure and legal label names. As with the encoder-decoder formulation, outputs are parsed into canonical `(type, text)` pairs and evaluated under the shared protocol.

Parameter-efficient fine-tuning. For large decoder-only models, full fine-tuning can be computationally expensive. We therefore use parameter-efficient fine-tuning via LoRA adapters, optionally combined with low-bit quantization, to adapt instruction-following models to the ASC-PIE extraction format while keeping the majority of base parameters fixed. Given a pretrained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, LoRA replaces full updating with a low-rank adaptation

$$W = W_0 + \Delta W, \qquad \Delta W = BA, \tag{8}$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and $r \ll \min(d, k)$ is the adapter rank. In this way, only the low-rank matrices are optimized while the base model remains frozen. When QLoRA is used, the frozen base model is loaded in low-bit quantized form, and training is applied only to the adapter parameters. This enables practical supervised adaptation on available hardware while preserving a consistent interface for generation and evaluation. The exact QLoRA adapter configuration used in the experiments is reported in Appendix A.

Training objective and generation. Given the prompt X and the gold response sequence $Y^* = [y_1^*, y_2^*, \ldots, y_T^*]$, supervised adaptation minimizes the causal language modeling loss

$$\mathcal{L}_{\text{dec}} = -\sum_{t=1}^{T} \log P(y_t^* \mid X, y_{<t}^*), \tag{9}$$

which corresponds to teacher-forced next-token prediction under the instruction-conditioned prompt. At inference time, the model generates an output sequence $\hat{Y} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{\hat{T}}]$, under fixed decoding settings. Because the output must satisfy both semantic and structural constraints, evaluation distinguishes extraction errors from format violations.

Output parsing for shared evaluation. Decoder-only outputs are evaluated through a strict JSON parser that accepts only responses conforming to the required schema and legal ASC-PIE labels. When parsing succeeds, the predicted entities are converted into the shared representation defined in Section 4.5. Outputs that fail JSON parsing or schema validation are treated as invalid under the shared evaluation protocol. This allows decoder-only predictions to be compared directly with encoder and encoder-decoder outputs under the same entity-level representation.

Typical extraction errors. Decoder-only extraction can fail through semantic errors such as omission, type confusion, and hallucination, where unsupported entities or labels are generated. It can also fail through structural errors, such as malformed JSON, extra commentary, or unsupported label names, which motivate explicit output verification in generation-based extraction. These errors motivate joint reporting of extraction quality and validity, and are analyzed later together with latency and throughput.

4.4 Adaptation Protocols

ASC-PIE compares adaptation approaches that reflect how PII extractors are deployed and maintained in practice. We treat supervised fine-tuning as the primary adaptation mechanism for all model families, instantiated using family-appropriate objectives and output constraints. We also define an in-context learning (ICL) setup as a parameter-free baseline for instruction-following extraction. Finally, for the staged incremental setting, we evaluate update strategies designed to reduce catastrophic forgetting while remaining feasible under privacy and large-model constraints.

4.4.1 Supervised Fine-Tuning

Supervised fine-tuning is the primary parameter-updating adaptation mode in ASC-PIE. Under this protocol, pretrained models are adapted to the extraction task using labeled examples while keeping the unified ASC-PIE schema and fixed data splits constant. The exact training objective and output interface are family-specific: encoder models use token classification under the IOB2 schema, encoder-decoder models generate constrained key-value extraction strings, and decoder-only models generate structured outputs under the ASC-PIE instruction-following interface. This protocol provides the main basis for cross-family comparison under shared evaluation rules. Exact model choices and training configurations are reported in the experimental setup and Appendix A.

4.4.2 In-Context Learning (ICL) Setup

In-context learning performs extraction without parameter updates by conditioning a model on a prompt that includes task instructions and a small number of demonstration examples. In ASC-PIE, ICL is used as a parameter-free baseline for decoder-only instruction-following extraction, enabling comparison with supervised fine-tuning under the same label schema, output constraints, and evaluation rules.

ASC-PIE supports two prompting protocols for this setting. The first uses a strict JSON schema aligned with the supervised decoder-only interface, requiring machine-readable outputs under the ASC-PIE label inventory. The second uses a lighter key–value (KV) schema that expresses extracted entities as semicolon-delimited type–value pairs. In both cases, prompts enumerate the allowed ASC-PIE entity types and require structured outputs consistent with the target protocol. The prompt protocol overview and the corresponding template formats are provided in Appendix B, while the exact prompting formats, shared controls, and decoding settings used in the experiments are summarized in Appendix A, Table A.2.

4.4.3 Continual Learning and Incremental Staging

ASC-PIE is designed to evaluate PII extraction not only as a static supervised task, but also as an evolving capability that must be maintained as requirements change. In realistic deployments, the PII inventory expands as new identifiers become relevant and models are updated repeatedly rather than trained once and frozen. We therefore adopt a staged class-incremental protocol that models task expansion explicitly and measures both the acquisition of new capabilities and the retention of previously learned ones.

The continual-learning benchmark is defined as a three-stage class-incremental sequence-labeling problem. Figure 4.8 shows the three-stage design used in this thesis. Stage 0 contains the core entity types {`NAME`, `EMAIL`, `PHONENUM`, `DATE`, `TIME`, `LOCATION`, `ORGANIZATION`}. Stage 1 introduces identity-document and structured-ID types {`CREDITCARD`, `NATIONALID`, `PASSPORT`, `DRIVER-LICENCE`, `SIN`, `TAXNUM`}, and Stage 2 introduces geographic and role-

related types {CITY, STATE, COUNTRY, ZIPCODE, JOBTITLE, GENDER}. During training, each stage exposes only its newly introduced types, while all other entity tokens are relabeled to O, thereby creating the poisoned-gradient setting used to study forgetting. For evaluation, we construct both stage-restricted test views, which retain only the labels associated with a given stage, and cumulative mixed test views, which retain all labels introduced up to that stage. This separation allows the protocol to distinguish isolated stage performance from more realistic continual-learning retention under mixed-label evaluation.

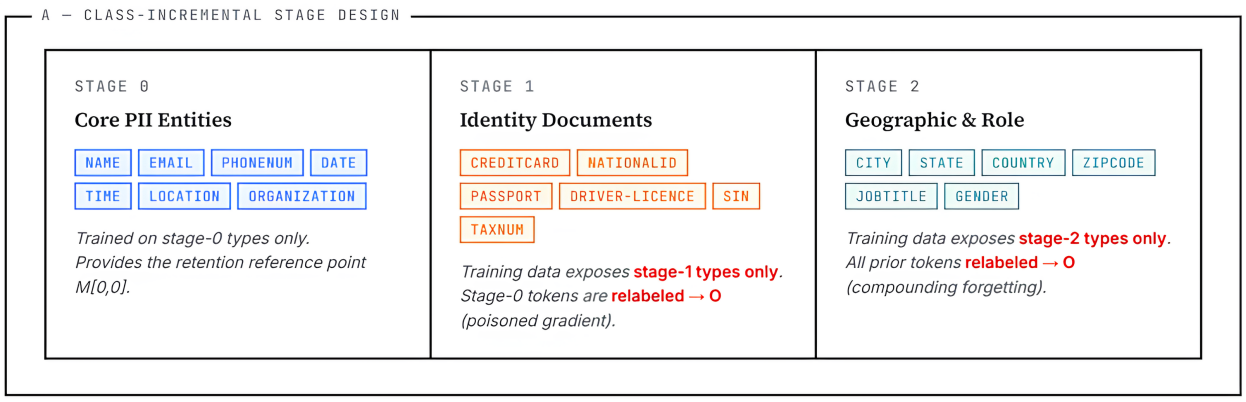

Figure 4.8: Three-stage class-incremental design used in ASC-PIE.

Stage definition. A stage corresponds to a discrete update step in which the active label set is expanded relative to previous stages. In this thesis, the protocol is class-incremental: new entity types are introduced over time while previously introduced types remain part of the overall task definition. The protocol preserves disjoint train, validation, and test splits throughout, so retention is always measured on held-out data rather than on replayed training examples.

Evaluation across stages. After each stage update, the model is evaluated on both the current stage and the earlier stages using two complementary views. First, stage-restricted evaluation measures performance on the entity types associated with a single stage in isolation, providing a clean view of current-stage learning and per-stage retained capability. Second, cumulative mixed evaluation measures performance on all entity types introduced

up to a given stage, providing the primary basis for assessing catastrophic forgetting and retained utility under a realistic mixed-label setting. Together, these views distinguish whether a model can still perform well when a stage is considered alone and whether that knowledge remains usable once old and new entity types are evaluated jointly. In additional to supporting analysis of the stability-plasticity trade-off by distinguishing isolated stage performance from retention under joint old-and-new label evaluation.

4.4.4 Mitigating Catastrophic Forgetting

In the staged incremental setting, sequential fine-tuning can lead to catastrophic forgetting, where adapting a model to new supervision degrades performance on previously learned entity types. Following the unified comparison protocol, we first identify a representative best-performing model within each family and then select an overall best-performing model under shared criteria. The continual-learning experiments are subsequently conducted on this selected encoder model so that all update strategies are compared on the same architecture, the same data splits, and the same stage design. Figure 4.9 summarizes the four strategies compared in the incremental setting.

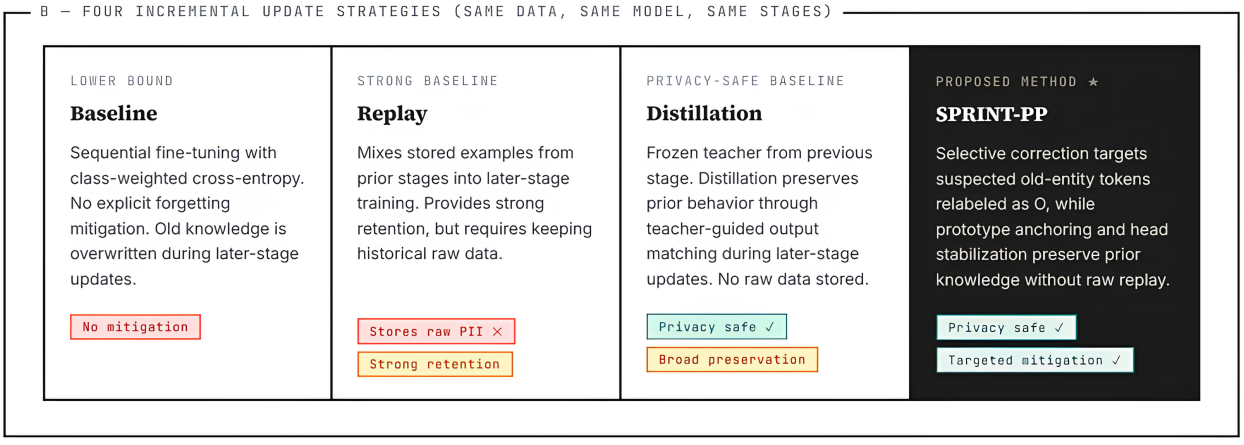

Figure 4.9: Incremental update strategies compared under the same model, data, and stage design: baseline sequential fine-tuning, replay, distillation, and the proposed SPRINT-PP method.

All four strategies are evaluated under the same shared encoder training configuration so that differences in retention can be attributed to the update mechanism itself rather than to changes in architecture, data partitioning, or stage construction.

Baseline sequential fine-tuning. The baseline applies the weighted token-classification trainer sequentially across the three stages with no explicit anti-forgetting mechanism. At Stage 0, the model is trained on the core entity set only. At later stages, the same model is further fine-tuned on new-stage data where old entity tokens have been relabeled to `O`. This provides a lower-bound continual-learning reference by exposing the model directly to the poisoned-gradient setting without replay or auxiliary constraints.

Replay implementation. Following rehearsal-based continual learning and continual NER baselines, replay is implemented by sampling a fixed buffer of `5,000` examples from each prior stage and concatenating these examples with the current-stage training set. In this thesis, the replay buffer is oversampled to approximately 25% of the new-stage data volume so that previously seen entity patterns are not overwhelmed by the larger current-stage set. After mixing, the combined dataset is shuffled, re-tokenized, and used to recompute class weights before training. Replay serves as a strong empirical baseline, but because it stores raw prior training examples, it does not satisfy the privacy-preserving design goals emphasized elsewhere in the thesis.

Distillation implementation. Distillation is implemented by loading the checkpoint from the previous stage as a frozen teacher and adding a knowledge-distillation objective during training on the current stage. The updated model is encouraged to preserve prior behavior through teacher-guided output matching while also learning the newly introduced supervision. In this thesis, distillation is included as a privacy-safe baseline because it does not require storing raw historical examples, although its preservation signal is applied more broadly than the targeted corrective mechanism introduced later in SPRINT-PP.

SPRINT-PP motivation. SPRINT-PP (Selective Prototype-Routed Incremental NER Triage with Privacy Preservation) is motivated by a specific failure mode in class-incremental NER. At a later stage, the trainingal NER. At a later stage, the training data contain three conceptually different kinds of tokens: tokens belonging to newly introduced entity types, tokens that are genuinely outside any entity, and tokens belonging to previously learned entity types that are now relabeled as `O`. The third case is the most problematic. These tokens do not merely fail to reinforce old knowledge; they actively produce poisoned gradients that train the model to suppress entity predictions it had previously learned. In this sense, forgetting in staged NER is driven not only by missing supervision, but also by incorrect supervision on old entity tokens.

Limitations of replay and distillation. Replay provides a strong empirical baseline because it reintroduces correct supervision for older classes. However, it requires storing raw historical examples, which is undesirable in privacy-sensitive PII extraction because those examples may contain personal identifiers. Distillation avoids this problem by using a frozen teacher model rather than a replay buffer, but its preservation signal is broad: it encourages the updated model to remain close to prior behavior across supervised positions without explicitly isolating the old entity tokens that have been relabeled as `O`. SPRINT-PP is designed to address this gap by preserving prior knowledge in a more targeted way while remaining privacy-safe.

Relation to MiB. The closest conceptual prior is MiB [43], proposed for class-incremental semantic segmentation. MiB shows that forgetting can arise when old classes are relabeled as background in later stages and addresses this problem through teacher-based correction of those mislabeled regions. SPRINT-PP adopts the same intuition: old knowledge is damaged not only by omission, but also by relabeling. However, the NER setting is different because it operates on token sequences rather than pixels, is privacy-oriented, and extends teacher-guided correction with prototype-based anchoring and classifier-head stabilization. Thus, MiB provides the conceptual starting point, whereas SPRINT-PP is a task-specific

extension for privacy-aware continual NER.

SPRINT-PP implementation overview. SPRINT-PP follows the same staged training framework as the other continual-learning baselines, but modifies later-stage updates in a targeted manner. Stage 0 is trained using plain weighted cross-entropy, since no previously learned entity types must yet be preserved. After each completed stage, the method stores two compact summaries of prior knowledge: prototype centroids extracted from labeled entity representations, and a snapshot of the classifier-head rows associated with the learned labels. These summaries are retained instead of raw historical examples.

At Stages 1 and 2, SPRINT-PP loads the previous-stage checkpoint as a frozen teacher. During training on the new stage, it examines only tokens currently labeled as `O`. If the teacher is sufficiently confident that such a token still belongs to one of the old labels, that token is routed away from the standard task loss and into a corrective path. In this path, the student is encouraged to stay close to the teacher's old-class behavior, to remain near the stored prototype of the corresponding old type in hidden space, and to preserve the classifier-head rows learned in previous stages. To further reduce optimization drift and training cost, the token embeddings and lower encoder layers are kept frozen during later-stage SPRINT-PP updates, while the upper encoder layers and task head remain trainable.

This design makes SPRINT-PP different from both replay and distillation. Unlike replay, it does not retain raw historical training examples. Unlike distillation, it does not apply a broad preservation signal uniformly across all supervised positions. Instead, it first identifies the tokens most likely to correspond to old entities that were relabeled as `O`, and then applies a targeted corrective update only to those positions.

Prototype memory and routed correction. For each entity type c learned up to a given stage, SPRINT-PP stores a prototype centroid

$$\mu_c = \frac{1}{|H_c|} \sum_{h \in H_c} h, \tag{10}$$

where H_c denotes the set of final-layer hidden representations associated with labeled tokens of type c. These prototypes provide a compact privacy-safe summary of prior knowledge and are updated stage by stage without storing raw historical text.

During a later stage t, SPRINT-PP examines only tokens currently labeled as `O`. For each such token, the frozen teacher estimates whether it still resembles one of the labels learned in earlier stages. Concretely, we compute

$$q_{b,i}^{\text{old}} = \max_{c \in \mathcal{Y}_{\text{old}}^{(t)}} p_{\text{teacher}}(c \mid x_{b,i}), \tag{11}$$

which is the teacher's highest confidence over the previously learned labels. Tokens whose score exceeds a threshold τ_{triage} are treated as suspected old-entity tokens and handled by the corrective branch of SPRINT-PP rather than by the standard `O`-label supervision.

Objective function. The implemented SPRINT-PP objective combines the standard task loss with targeted correction and two privacy-safe retention terms:

$$L_{\text{SPRINT-PP}} = L_{\text{task}} + L_{\text{SLC}} + \lambda_{\text{PP}}^{(t)} L_{\text{PP}} + \lambda_{\text{HRA}}^{(t)} L_{\text{HRA}}. \tag{12}$$

Here, L_{task} is the standard supervised loss applied to non-routed tokens, L_{SLC} is a selective soft-label correction term that aligns student and teacher outputs only on suspected old tokens, L_{PP} pulls routed hidden representations toward the stored prototype of the corresponding old type, and L_{HRA} stabilizes the classifier-head rows associated with previously learned labels.

To keep the retention terms proportional to stage difficulty, SPRINT-PP uses stage-dependent weights

$$\lambda_{\text{PP}}^{(t)} = \lambda_{\text{PP}}^{\text{base}} \sqrt{\frac{|\mathcal{Y}_{\text{old}}^{(t)}|}{|\mathcal{Y}_{\le t}|}}, \qquad \lambda_{\text{HRA}}^{(t)} = \lambda_{\text{HRA}}^{\text{base}} \sqrt{\frac{|\mathcal{Y}_{\text{old}}^{(t)}|}{|\mathcal{Y}_{\le t}|}}, \tag{13}$$

where $|\mathcal{Y}_{\text{old}}^{(t)}|$ is the number of labels learned before stage t and $|\mathcal{Y}_{\le t}|$ is the total number of

labels available up to that stage. These dynamic coefficients prevent the retention terms from remaining fixed as the old-to-total label ratio changes across stages.

Overall, SPRINT-PP can be understood as a targeted privacy-safe alternative to broad teacher preservation and raw replay. It first asks which `O`-labeled tokens are most likely to correspond to forgotten old entities, then corrects only those tokens, and finally preserves prior knowledge through compact prototype memory and classifier-head stabilization rather than through stored historical examples.

4.5 Shared Evaluation Protocol and Metrics

ASC-PIE evaluates PII extraction using a unified protocol designed to support fair comparisons across model families and adaptation paradigms. Because encoder models produce token tags while generative models produce structured text, ASC-PIE normalizes all predictions into a shared entity representation and applies a single evaluation harness across families. This section defines the shared entity representation, normalization rules, extraction metrics, validity criteria, and stage-wise continual-learning metrics used throughout the thesis.

Canonical entity representation. All models are evaluated on a canonical entity representation under the ASC-PIE schema:

$$E = \{(\tau_m, x_m)\}_{m=1}^{M}, \tag{14}$$

where τ_m is the canonical ASC-PIE entity type and x_m is the extracted text span. For encoder models, predicted IOB2 tag sequences are converted into entity mentions by reconstructing spans from contiguous `B-`/`I-` labels. For encoder-decoder models, generated key-value outputs are parsed into entities of the form `TYPE: value`. For decoder-only models, generated outputs are parsed from a strict JSON schema of the form `{"entities":[{"type":"...", "text":"..."}, ...]}`. In all cases, type strings are canonicalized to ASC-PIE label names

using an alias map so that harmless naming variation does not distort evaluation; for example, variants such as `DRIVERLICENCE` are mapped to the canonical form `DRIVER-LICENCE` before scoring.

Strict and normalized text matching. Entity-level scoring is performed by matching predicted and gold entities using extracted text spans. We report two matching modes. *Strict* scoring requires exact equality of the extracted entity text. *Normalized* scoring applies simple normalization, including whitespace collapse, case-folding, and trimming of surrounding punctuation, to reduce sensitivity to minor formatting variation while preserving entity identity. Reporting both views helps distinguish true extraction failures from superficial formatting differences common in generated outputs and token-based reconstruction.

Micro-averaged extraction metrics and error decomposition. Extraction quality is reported using micro-averaged precision, recall, and F1 over entities. Let TP, FP, and FN denote the total numbers of true positives, false positives, and false negatives, respectively, aggregated over all evaluation examples. The micro-averaged metrics are then defined as

$$P_{\text{micro}} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad R_{\text{micro}} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad F1_{\text{micro}} = \frac{2P_{\text{micro}}R_{\text{micro}}}{P_{\text{micro}} + R_{\text{micro}}}. \tag{15}$$

To support privacy-relevant failure analysis, ASC-PIE also reports error counts that distinguish omissions, corresponding to false negatives, from hallucinations, corresponding to false positives. When a predicted entity text matches a gold entity text but the type differs, the error is treated as type confusion and counted as one false positive for the predicted type and one false negative for the gold type. Matching is performed greedily within each example to ensure consistent counting in the presence of duplicate mentions.

Output validity. ASC-PIE distinguishes extraction quality from output validity across model families. *Validity* measures whether a prediction conforms to the required format and, where applicable, uses only legal canonical label names. For encoder models, predictions must consist only of legal ASC-PIE token labels. For encoder-decoder models, outputs must

be either a well-formed key-value list or the null output `No PII found.` For decoder-only models, outputs must parse as JSON and satisfy the required schema, including legal canonical types and non-empty entity texts. We report validity rate as the proportion of examples whose outputs satisfy these constraints. Under strict validity scoring, invalid outputs are treated as empty predictions, reflecting the operational requirement that malformed outputs cannot be consumed safely in privacy pipelines.

Stage-restricted evaluation view. For class-incremental experiments, ASC-PIE first reports a *stage-restricted* matrix $R[t, s]$. Each cell records the F1 score on the restricted stage-s test view after training through stage t, where only the entity types assigned to stage s are counted. This view provides a clean measure of isolated stage performance. As shown in Figure 4.10, the diagonal of $R[t, s]$ shows how well the model learns each stage when it is first introduced, while the entries below the diagonal show how well earlier stages remain recoverable when they are evaluated alone.

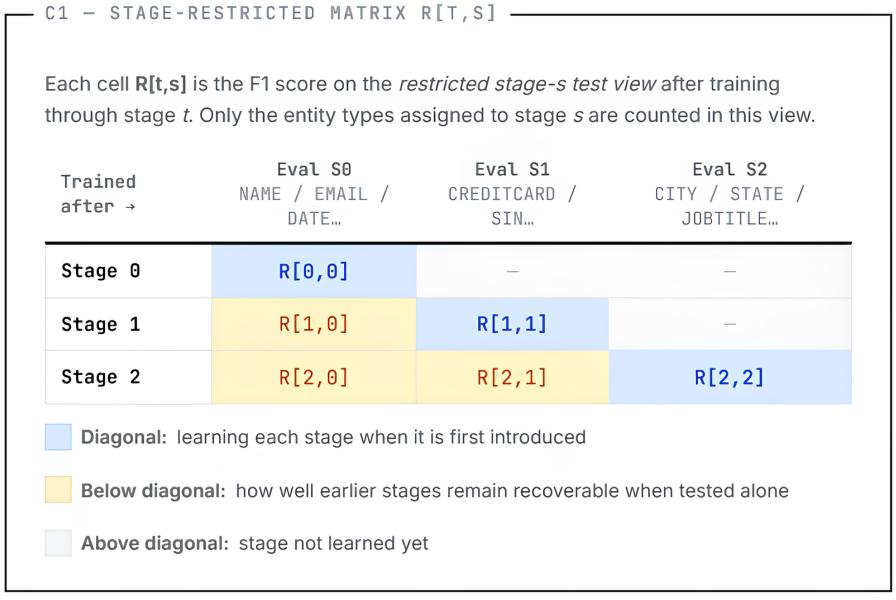

Figure 4.10: Stage-restricted evaluation matrix $R[t, s]$. Each cell records F1 on the restricted stage-s test view after training through stage t, so the matrix provides a clean view of isolated stage learning and earlier-stage recoverability.

Cumulative mixed evaluation view. ASC-PIE also reports a *cumulative mixed* matrix $C[t, s]$. In this view, each cell records the F1 score on the cumulative test view up to stage s after training through stage t. This means that the evaluation includes all entity types introduced from Stage 0 through Stage s. As illustrated in Figure 4.11, the diagonal shows performance when each cumulative stage is first reached, while the entries below the diagonal show whether earlier knowledge remains usable once old and new labels are evaluated together. This cumulative view is the primary basis for retention, forgetting, and final continual-learning comparison because it reflects the more realistic mixed-label setting.

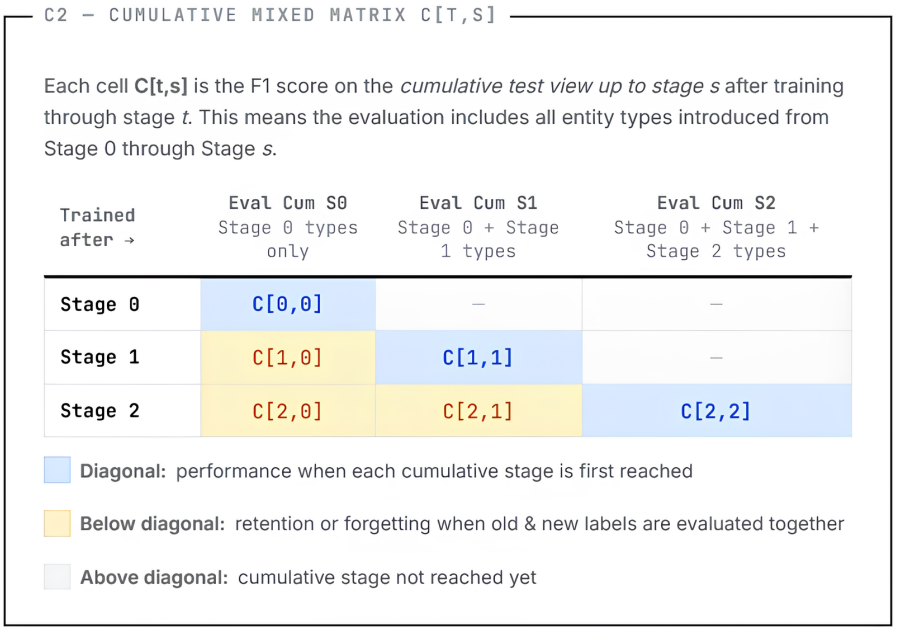

Figure 4.11: Cumulative mixed evaluation matrix $C[t, s]$. Each cell records F1 on the cumulative test view up to stage s after training through stage t, making this the primary view for retention and forgetting under mixed-label evaluation.

Continual-learning metrics. Let K denote the final stage index. The continual-learning summary metrics used in this thesis are derived from the two complementary views above. Plasticity and intransigence are computed from the diagonal of the stage-restricted matrix $R[t, s]$, because they measure how well each stage is learned when first introduced. By contrast, ACC, Forgetting, and BWT are computed from the cumulative matrix $C[t, s]$, because they are intended to reflect final retained utility and backward degradation under

realistic mixed-label evaluation. Figure 4.12 summarizes these metrics and their interpretation.

Figure 4.12: Continual-learning metrics used in ASC-PIE. Plasticity and Intransigence are derived from the stage-restricted matrix $R[t, s]$, while ACC, Forgetting, and BWT are derived from the cumulative mixed matrix $C[t, s]$.

The final average cumulative performance after all stages is defined as

$$\text{ACC} = \frac{1}{K+1} \sum_{s=0}^{K} C[K, s]. \tag{16}$$

Equation 16 summarizes final continual-learning performance by averaging the last row of the cumulative matrix.

$$\text{Plasticity} = \frac{1}{K+1} \sum_{s=0}^{K} R[s, s]. \tag{17}$$

Equation 17 measures how well the model learns each stage when it is first introduced.

$$\text{Forgetting} = \frac{1}{K}\sum_{s=0}^{K-1}\big(C[s,s] - C[K,s]\big). \tag{18}$$

Equation 18 measures the average drop in cumulative performance between the stage at which a stage was first learned and the final stage.

$$\text{BWT} = \frac{1}{K}\sum_{s=0}^{K-1}\big(C[K,s] - C[s,s]\big). \tag{19}$$

Equation 19 measures backward transfer under cumulative mixed evaluation. Values closer to zero indicate weaker backward degradation, while negative values indicate forgetting.

$$\text{Intransigence} = \frac{1}{K+1}\sum_{s=0}^{K}\big(1 - R[s,s]\big). \tag{20}$$

Equation 20 captures the difficulty of learning newly introduced stages when they first appear.

Together, $R[t,s]$ and $C[t,s]$ provide complementary views of continual-learning behavior. The restricted matrix is used to interpret isolated stage learning, whereas the cumulative matrix provides the primary basis for retention, forgetting, and final continual-learning comparison under mixed-label evaluation.

Robustness and efficiency. In addition to in-distribution accuracy, ASC-PIE evaluates robustness to realistic perturbations and reports practical efficiency indicators. Robustness experiments introduce controlled noise such as casing variation and character-level substitutions to measure sensitivity under operational text conditions. Efficiency is assessed using runtime-oriented indicators such as latency and throughput for inference, which are especially relevant for large instruction-following models. These dimensions complement extraction accuracy by characterizing whether a method remains reliable and deployable under realistic constraints.

4.6 Reproducibility and Ethical Considerations

ASC-PIE is designed to be reproducible and auditable, reflecting the operational sensitivity of PII-related extraction systems. We therefore standardize corpus construction, model interfaces, staged evaluation, and artifact export so that results can be reproduced and re-verified as the experimental study evolves.

Reproducible corpus construction. Corpus construction follows deterministic preprocessing and schema-mapping rules, and the resulting splits are fixed and disjoint across all experiments. ASC-PIE exports explicit label maps and preserves source identifiers as metadata, enabling reconstruction of the consolidation process and targeted analysis by source. It also generates synchronized dataset representations aligned with each model family, ensuring that comparisons are performed over consistent inputs and evaluation splits.

Standardized evaluation artifacts. All model families are evaluated through the same shared harness, which normalizes outputs to canonical (type, text) entities and reports a consistent set of metrics. Evaluation runs produce standardized artifacts, including summary files, per-type metric tables, and diagnostic outputs used directly in later comparative analyses. This reduces the risk of family-specific post-processing inconsistencies and supports transparent reporting of extraction quality, validity, and stage-wise retention.

Environment and configuration tracking. To support reruns and auditability, the experimental study preserves environment summaries, dependency exports, and run-specific configuration records across model families and adaptation settings. Where appropriate, the implementation code and supporting experiment scripts are also maintained in a version-controlled repository so that reported results can be rechecked at both the configuration and evidence levels.

Ethical safeguards and privacy-aware mitigation. Because the task concerns sensitive information, we follow established definitions of PII and treat privacy protection as

a primary motivation for extraction. To reduce the risk of exposing personal identifiers, ASC-PIE emphasizes the consolidation of public resources and augmentation with synthetic PII patterns rather than the collection of real personal data. Synthetic augmentation is constructed using non-operational values for structured identifiers and controlled generation procedures for name-like entities, with care taken to avoid producing real identifiers. In the continual-learning setting, this concern also affects mitigation design: replay can improve retention but requires storing raw historical examples, whereas distillation and SPRINT-PP are privacy-aware alternatives that avoid raw replay. SPRINT-PP further limits retained history to compact model-side summaries, including prototype centroids and previously learned classifier-head rows, rather than original text examples, improving compatibility with privacy-sensitive deployment constraints.

Dataset documentation. We document corpus composition, schema design choices, preprocessing decisions, and intended evaluation use to support responsible reuse and interpretation. This aligns with recommendations for dataset transparency, including explicit reporting of composition, processing decisions, and limitations. Later chapters provide further reporting on dataset balance, label coverage, and error modes observed during evaluation.

4.7 Summary

This chapter introduced ASC-PIE, the Augmented Synthetic Corpus for PII and Incremental Evaluation, as a unified corpus and protocol for studying PII extraction under evolving requirements. We defined the unified 19-type label schema, described how heterogeneous public resources and a thesis-specific synthetic component are normalized and consolidated, and specified fixed, leakage-controlled splits that remain consistent across all experiments. We then formalized task formulations for encoder-based token classification, encoder-decoder structured prediction, and decoder-only instruction-following extraction, together with the adaptation paradigms used in this thesis, including supervised fine-tuning, in-context learning, and staged incremental updating.

The chapter also established the class-incremental protocol used to study catastrophic forgetting under evolving PII requirements. In this setting, new entity groups are introduced across stages while previously learned entity tokens may remain in the text but be relabeled as `O`, creating the poisoned-gradient setting addressed later in the thesis. To study this problem under shared conditions, ASC-PIE supports baseline sequential fine-tuning, replay, distillation, and the proposed SPRINT-PP method within the same staged evaluation framework.

Finally, we specified a unified evaluation protocol that normalizes outputs across model families to canonical `(type, text)` entities, reports strict and normalized matching, measures validity for structured generation outputs, and computes continual-learning metrics from complementary stage-restricted and cumulative mixed evaluation views. Together, these components establish the methodological foundation for the comparative analyses reported in the results chapter.

Overall, ASC-PIE serves not only as a dataset construction effort, but as an integrated experimental framework that links corpus design, model adaptation, evaluation, and continual-learning analysis under a single methodology. This integration is important because the research questions in this thesis cannot be answered reliably through isolated model runs or dataset-specific settings. By enforcing common schemas, synchronized representations, shared scoring rules, and a consistent staged protocol, ASC-PIE provides the methodological basis needed to compare model families fairly, interpret forgetting behavior rigorously, and assess whether targeted privacy-aware mitigation strategies such as SPRINT-PP offer practical advantages under evolving PII extraction requirements.

5 Results and Analysis

5.1 Chapter Overview and Reporting Conventions

This chapter presents the empirical results of the ASC-PIE study and organizes them according to the four research questions introduced in Chapter 1. The analysis proceeds from the comparison of model families under supervised fine-tuning, to the contrast between fine-tuning and in-context prompting, to the staged continual-learning study, and finally to the role of corpus design and synthetic augmentation. The chapter concludes with a synthesis of the main findings and cross-cutting observations.

To maintain methodological consistency, all reported results are drawn from the standardized exported artifacts produced by the implementation pipelines defined under the ASC-PIE framework. Unless otherwise stated, cross-family supervised fine-tuning comparisons are reported on a common held-out evaluation subset of $n = 1000$ examples, while the in-context learning experiments use a fixed subset of $n = 500$ examples across all shot settings. The staged continual-learning analysis is reported separately using complementary stage-restricted and cumulative mixed evaluation views, together with the continual-learning metrics derived from them [12].

Across all experimental blocks, predictions are scored through the shared ASC-PIE evaluation harness. Results are reported primarily using *strict* entity-level precision, recall, and F1, following standard entity-centric evaluation practice in NER and extraction [3, 22, 24]. *Normalized* F1 is reported as a secondary view that tolerates minor formatting differences in entity text through case-folding, whitespace normalization, and punctuation trimming. For structured-generation models, *validity rate* is additionally reported to quantify the proportion of outputs that satisfy the required schema and label constraints, consistent with prior work emphasizing output verification in generation-based extraction [15, 57]. Where relevant, we also report *salvage* scores, which measure extraction performance under a more tolerant parser and are used only as diagnostic indicators rather than as the primary basis

for ranking models.

All experiments were executed on a Dell Precision 3660 workstation running Windows 11, equipped with a 13th Gen Intel i7-13700 CPU, 128 GB RAM, and an NVIDIA RTX A6000 GPU with 48 GB VRAM. The experimental software stack was implemented in Python using PyTorch and Hugging Face Transformers and Datasets, with supporting analysis through standard scientific Python tooling. Decoder-only fine-tuning additionally relied on PEFT and `bitsandbytes` to enable QLoRA-style adaptation under the available hardware budget, and batched generation was used where needed to obtain stable latency and throughput measurements. These environment details are reported here because some of the later comparisons involve not only extraction quality but also runtime, memory, and structured-output reliability. Full hardware and software details are reported in Appendix A.

In addition to extraction quality, the chapter uses a small set of auxiliary indicators to support interpretation. *Hallucinations* refer to predicted entities that are not supported by the gold annotations, while *leakage rate* summarizes missed gold entities under the unified evaluator. For prompting-based experiments, throughput and average latency are reported where available to characterize practical inference cost. For the continual-learning study, performance is summarized through the stage-restricted matrix $R[t, s]$, the cumulative mixed matrix $C[t, s]$, and the derived metrics including ACC, Plasticity, Forgetting, Backward Transfer, and Intransigence [12]. Together, these conventions ensure that comparisons across model families, prompting protocols, and forgetting-mitigation strategies are made on a common and transparent basis.

5.2 RQ1: Comparison of Model Families under Supervised Fine-Tuning

5.2.1 Main Extraction Results on the Unified Comparison Subset

Table 5.1 and Figure 5.1 summarize the main supervised fine-tuning comparison across model families on the unified held-out comparison subset of $n = 1000$ examples. The results show a clear separation between the strongest encoder and encoder–decoder systems, which achieve near-ceiling extraction quality, and the lower-performing decoder-only and weaker seq2seq configurations, whose main limitation is recall rather than precision.

RoBERTa-large achieves the strongest overall result with a strict F1 of 0.9909, followed very closely by FLAN-T5-base at 0.9872. Both models also maintain perfect validity (1.000) and extremely low leakage rates, indicating that they not only extract PII accurately but do so in a reliably consumable format under the shared evaluator. This places RoBERTa-large as the best overall supervised model in the thesis and justifies its later selection for the staged continual-learning study.

The remaining encoder models form a second tier. ModernBERT-large reaches a strict F1 of 0.8464, while BERT-large-cased reaches 0.8068. BERT-large-cased shows closely matched precision and recall, indicating broadly consistent but weaker extraction than RoBERTa-large. ModernBERT-large outperforms BERT-large-cased in strict F1 and achieves a much higher normalized F1 (0.9254), suggesting that many of its strict errors arise from surface-form mismatches reduced by normalization rather than from complete failure to identify the correct entity.

Among the structured-generation baselines, FLAN-T5-base substantially outperforms BART-base. BART-base drops to a strict F1 of 0.3475 with validity 0.561. Its precision remains relatively high (0.8585), but recall falls to 0.2179, indicating that the model is conservative when it produces valid outputs and fails mainly through missed entities and structure-related generation weakness rather than excessive hallucination.

The decoder-only fine-tuned models show a similar precision–recall imbalance. Llama3.1-8B reaches a strict F1 of 0.6896 with validity 0.814, whereas Qwen2.5-7B reaches 0.4003 with validity 0.543. In both cases, precision is high (0.8467 for Llama3.1-8B and 0.9451 for Qwen2.5-7B), but recall is much lower, especially for Qwen2.5-7B. This indicates that the decoder-only models are often able to produce correct entities when they commit to a prediction, but under strict JSON-based scoring they fail to recover a substantial fraction of the entity mentions present in the gold reference annotations. This pattern further suggests that these models are more sensitive to the combined demands of extraction and structural compliance than the strongest encoder and encoder–decoder baselines.

Taken together, these results answer RQ1 in a clear way. Under supervised fine-tuning and a unified evaluation protocol, encoder-based token classification and strong encoder–decoder structured generation both provide very high extraction quality, with RoBERTa-large and FLAN-T5-base clearly leading the comparison. Decoder-only instruction-following models remain less competitive in this setting, largely because they combine lower validity with substantially lower recall, which limits their end-to-end extraction effectiveness despite often high precision.

Table 5.1: Main supervised fine-tuning comparison on the unified ASC-PIE evaluation subset ($n = 1000$).

Model	**Family**	**Valid**	**Strict P**	**Strict R**	**Strict F1**	**Norm. F1**	**Leakage**
RoBERTa-large	Encoder	1.000	0.9888	0.9929	**0.9909**	**0.9910**	0.0071
FLAN-T5-base	Enc–Dec	1.000	0.9859	0.9885	0.9872	0.9873	0.0115
ModernBERT-large	Encoder	1.000	0.8420	0.8509	0.8464	0.9254	0.1491
BERT-large-cased	Encoder	1.000	0.8061	0.8075	0.8068	0.8093	0.1925
Llama3.1-8B	Decoder	0.814	0.8467	0.5817	0.6896	0.6899	0.4183
Qwen2.5-7B	Decoder	0.543	0.9451	0.2539	0.4003	0.4008	0.7461
BART-base	Enc–Dec	0.561	0.8585	0.2179	0.3475	0.3564	0.7821

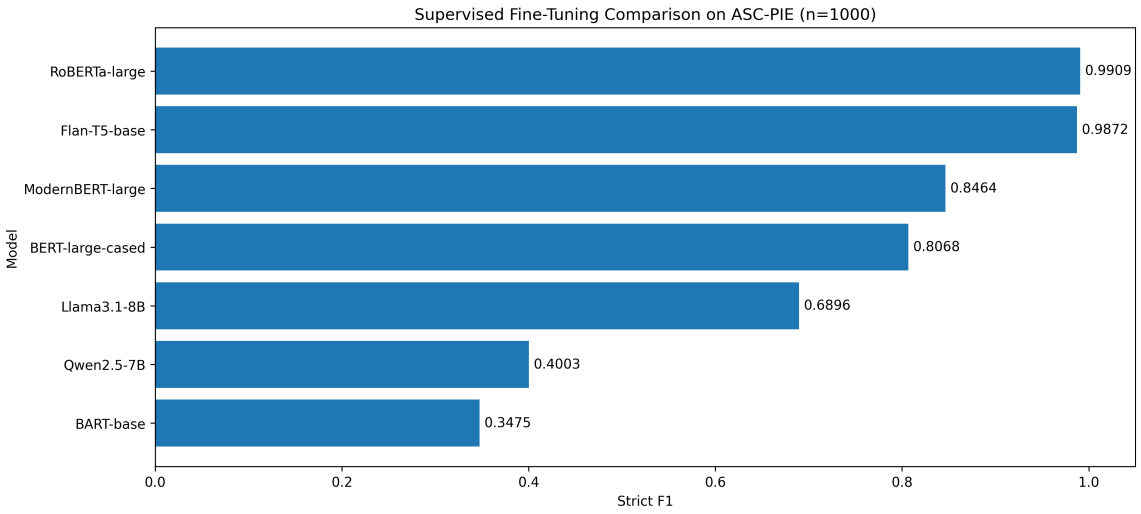

Figure 5.1: Supervised fine-tuning comparison across model families on the unified evaluation subset ($n = 1000$), ranked by strict F1.

Overall, the main supervised fine-tuning comparison shows a clear hierarchy across model families. RoBERTa-large achieves the strongest end-to-end result, while FLAN-T5-base is the closest competing generative baseline. The remaining encoders form a second tier, whereas decoder-only models and BART-base are limited mainly by recall and structural reliability. These results identify RoBERTa-large as the strongest overall supervised model under the unified ASC-PIE protocol.

5.2.2 Per-Type Performance and Error Patterns

The aggregate results in the previous subsection hide substantial variation across entity types. A first clear pattern is that the strongest models are not equally strong on all categories. RoBERTa-large and FLAN-T5-base remain near ceiling on many high-support types, including `NAME`, `EMAIL`, `DATE`, `CITY`, `STATE`, `SIN`, and `PHONENUM`. However, performance drops more noticeably on lower-support or structurally diverse categories such as `COUNTRY`, `NATIONALID`, `TAXNUM`, and `TIME`. Figure 5.2 shows this pattern clearly for the encoder and encoder–decoder models on the shared full-test basis. This indicates that the main remaining challenge under supervised fine-tuning is not common PII detection, but the long-tail and boundary-sensitive

cases that motivated the ASC-PIE design.

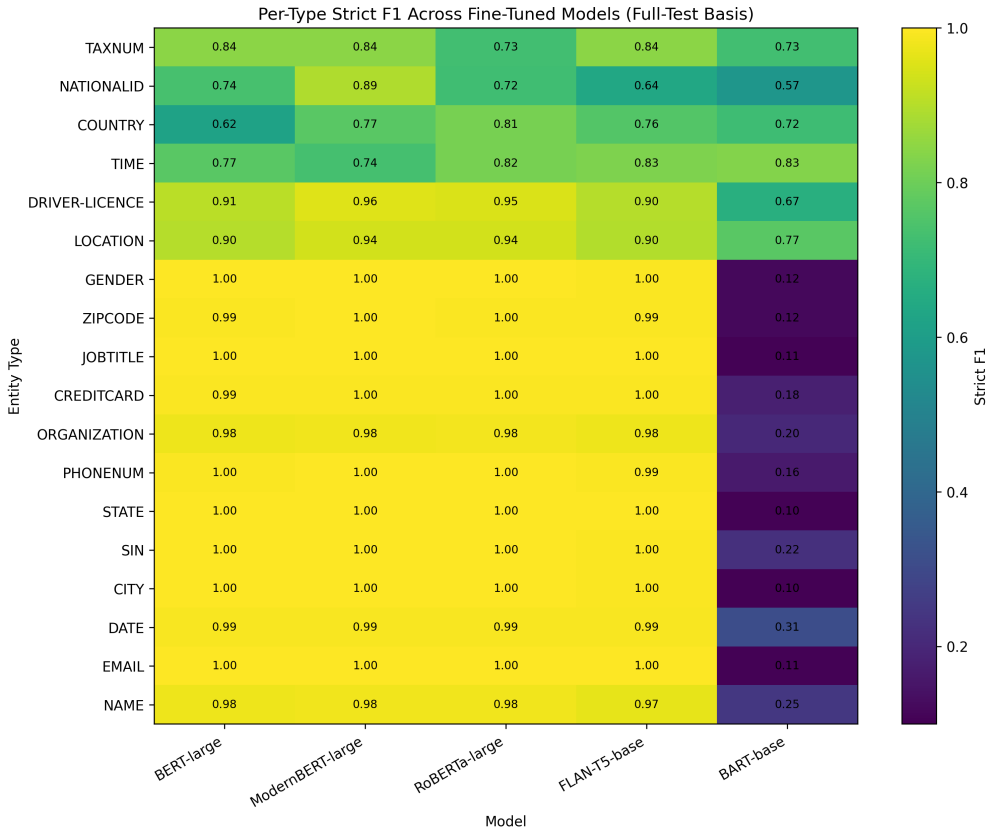

Figure 5.2: Per-type strict F1 across fine-tuned encoder and encoder–decoder models on the shared full-test basis. `PASSPORT` is omitted because it has zero support in the evaluation files.

The heatmap also clarifies the contrast between the best encoder and the best seq2seq model. RoBERTa-large is more consistent across the full label set, but FLAN-T5-base remains highly competitive and in some cases matches or slightly exceeds the encoder on selected categories, especially among structured and temporally formatted types. This suggests that constrained structured generation can be a viable alternative to token classification when the output interface is well controlled. At the same time, FLAN-T5-base is slightly less uniform than RoBERTa-large across the full inventory, which is consistent with the small overall ranking gap reported earlier.

BART-base exhibits a qualitatively different error profile. Its weakness is not confined to one or two rare categories, but extends across much of the label space. In Figure 5.2,

BART-base remains substantially below the stronger models on most common types and collapses especially severely on categories such as `CITY`, `STATE`, `JOBTITLE`, `ZIPCODE`, `EMAIL`, and `GENDER`. This pattern is consistent with the aggregate results: BART-base is not primarily a high-hallucination system, but rather a low-recall and low-validity system whose generation behavior fails to recover many gold entities reliably.

A more focused view is provided in Figure 5.3, which isolates selected rare and structured PII types. Three points stand out. First, `DRIVER-LICENCE` is comparatively well handled by the stronger models, especially ModernBERT-large and RoBERTa-large, suggesting that regular structured formats can be learned effectively. Second, `NATIONALID` remains more difficult and shows larger variation across models, with ModernBERT-large performing best in this focused set. Third, `COUNTRY` is consistently weaker than most common PII categories, indicating that contextual ambiguity and lower support remain challenging even for otherwise strong models. These differences show that overall F1 alone would hide meaningful variation in how models handle rare or semantically diffuse identifier types.

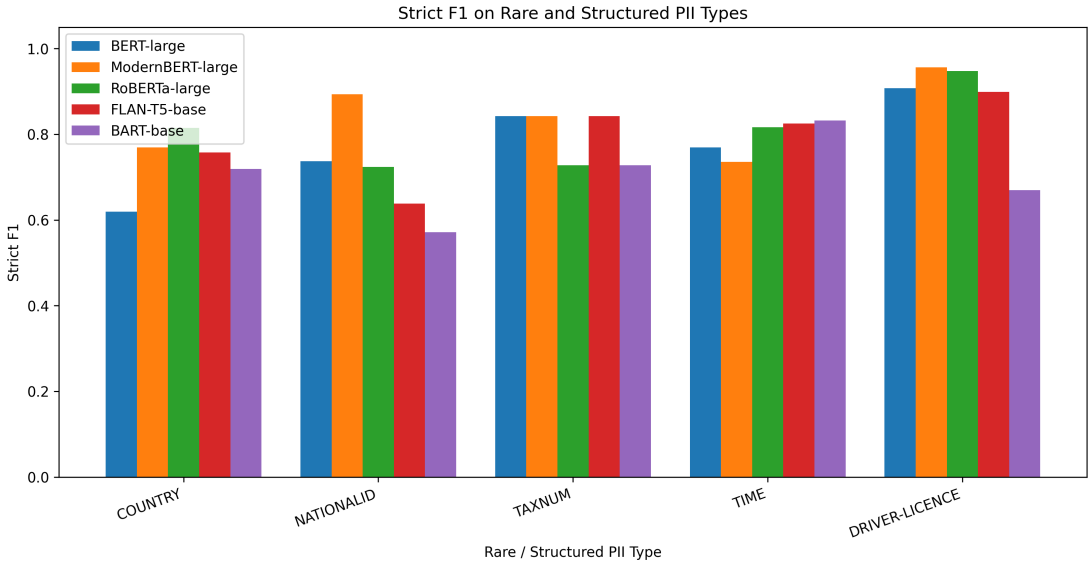

Figure 5.3: Strict F1 on selected rare and structured PII types under supervised fine-tuning.

The encoder diagnostics further indicate that the main encoder failure mode is usually boundary-related rather than arbitrary type substitution. The token-level confusion sum-

maries for BERT-large-cased, ModernBERT-large, and RoBERTa-large are dominated by transitions involving the non-entity class and inside tags, especially patterns such as `O` → `I-DATE`, `O` → `I-CREDITCARD`, `O` → `I-DRIVER-LICENCE`, and `B-NAME` → `I-NAME`. This indicates that encoder errors often arise through boundary fragmentation, missed span starts, or span-extension inconsistencies, rather than through confusion between unrelated semantic categories. In practical terms, the stronger encoders usually identify the correct entity category, but they sometimes fail to preserve the exact BIO boundary structure needed for perfect span reconstruction.

The decoder-only models show a different pattern and should be interpreted on their own comparison basis. As shown in Figure 5.4, Llama3.1-8B is consistently stronger than Qwen2.5-7B across most reported types under strict end-to-end scoring. The gap is particularly visible on `COUNTRY`, `ZIPCODE`, `JOBTITLE`, `STATE`, `SIN`, `DATE`, and `NAME`. Both decoder-only models perform relatively well on some structured categories such as `TIME`, `DRIVER-LICENCE`, and `TAXNUM`, but Qwen2.5-7B remains highly uneven across the broader label set. This matches the aggregate findings from the previous subsection: the decoder-only models are not uniformly poor at entity recognition, but their end-to-end extraction quality is limited by unstable recall and greater sensitivity to schema-compliant structured generation.

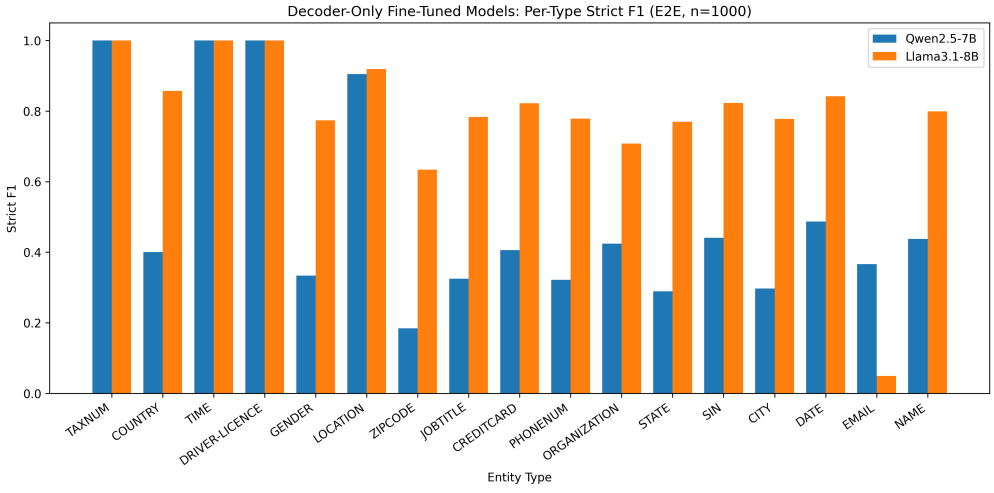

Figure 5.4: Per-type strict F1 for decoder-only fine-tuned models under end-to-end evaluation on the unified $n = 1000$ comparison subset.

Overall, the per-type analysis refines the answer to RQ1. RoBERTa-large and FLAN-T5-base are strong not only in overall F1 but also across most individual entity types, although difficult long-tail categories still separate them from perfect extraction. ModernBERT-large performs particularly well on some structured rare types despite trailing the top two models overall. BART-base fails broadly across the type inventory rather than only on rare categories. Finally, the decoder-only models show that fine-tuned instruction-following LLMs can succeed on selected structured types, but they remain substantially less consistent than the strongest encoder and encoder–decoder systems when evaluated under strict end-to-end extraction criteria.

5.2.3 Efficiency and Deployment Trade-Offs

Extraction quality alone is not sufficient to determine the practical value of a model family for deployment. The ASC-PIE results also reveal clear differences in training cost, memory demand, and inference speed, and these differences materially affect how the models should be interpreted in operational settings.

Within the encoder family, RoBERTa-large provides the strongest efficiency–accuracy balance. As shown in Figure 5.5, it achieves the best extraction results from RQ1 while also recording the lowest average epoch time among the three encoders (1471.6 s), slightly ahead of BERT-large-cased (1492.0 s), and well ahead of ModernBERT-large (1873.3 s). RoBERTa-large does use somewhat more peak memory than BERT-large-cased (8092 MB versus 7763 MB), but the gap is modest relative to the very large gain in extraction quality. By contrast, ModernBERT-large is both the slowest and the most memory-intensive encoder (9292 MB), which makes its weaker overall F1 and higher resource demand a less favorable trade-off under the current setup. The exported evaluation metrics also show that RoBERTa-large has the highest encoder-side evaluation throughput (339.1 samples/s), compared with 324.0 samples/s for BERT-large-cased and 267.6 samples/s for ModernBERT-large.

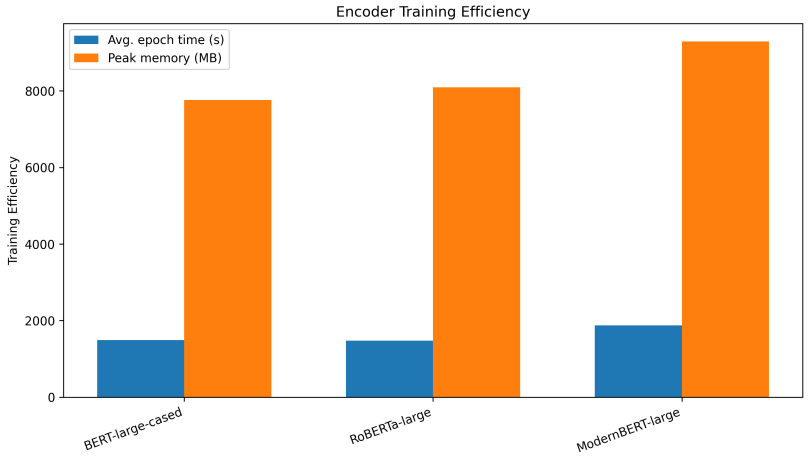

Figure 5.5: Encoder training efficiency in terms of average epoch time and memory usage.

The generative baselines show a different trade-off. As shown in Figure 5.6, the exported FLAN-T5 and BART notebooks indicate that BART-base is much faster than FLAN-T5-base in both training and generation. BART-base completes the three-epoch run in approximately 10,153 s, compared with 25,886 s for FLAN-T5-base, and its generation throughput is about 19.32 examples/s versus 6.70 examples/s. However, this speed advantage comes with much lower validity and much weaker extraction quality. FLAN-T5-base is therefore slower, but it achieves near-ceiling strict F1 together with perfect validity, making it the stronger deployment candidate when reliable structured outputs are required.

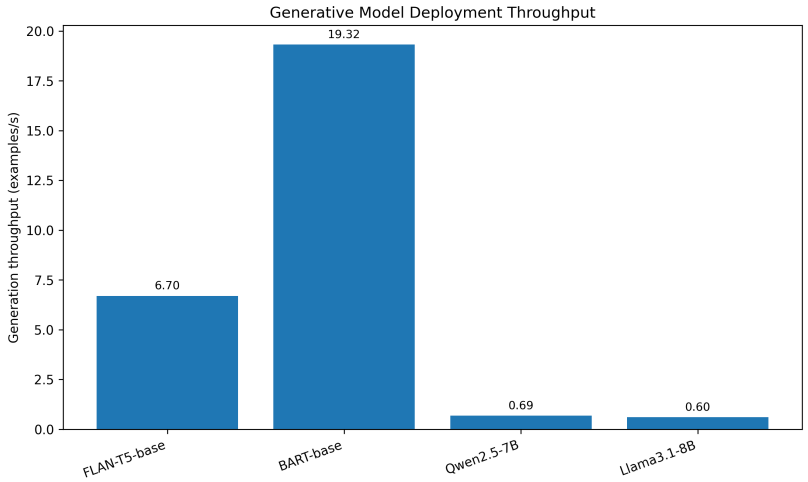

Figure 5.6: Generation throughput for the fine-tuned generative models.

The decoder-only models show a similarly clear deployment trade-off within their own family. As shown in Figure 5.7, Qwen2.5-7B is slightly faster than Llama3.1-8B, with average latency 1.445 s versus 1.646 s, p95 latency 1.49 s versus 1.70 s, and throughput 0.686 examples/s versus 0.603 examples/s. However, the speed advantage is small relative to the quality gap. In the main supervised comparison, Llama3.1-8B substantially outperforms Qwen2.5-7B in strict F1 and also provides a much higher validity rate. This means that, within the decoder-only family, the slight increase in latency for Llama3.1-8B is justified by materially better end-to-end extraction reliability.

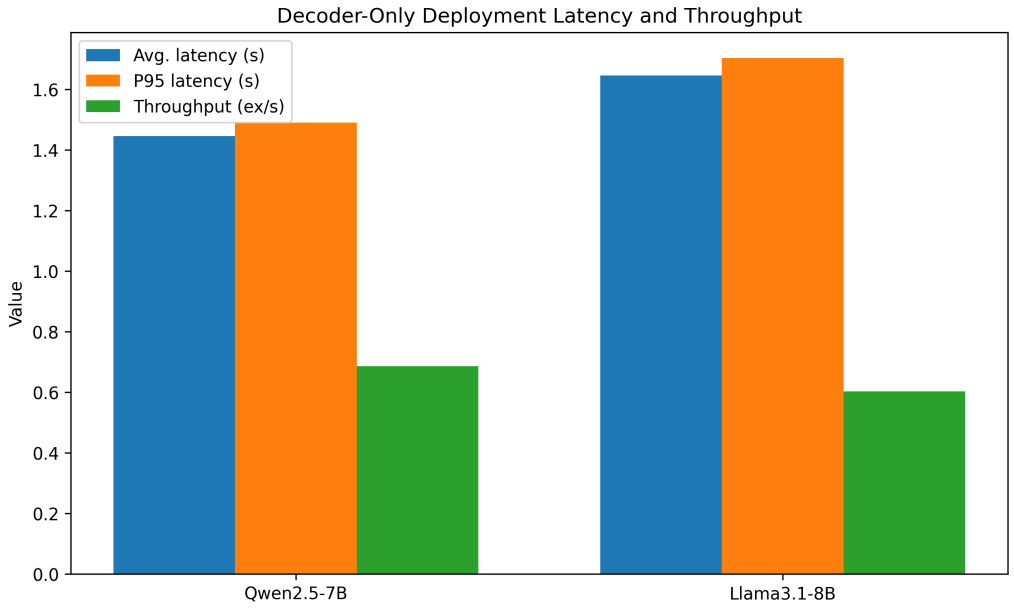

Figure 5.7: Average latency, p95 latency, and throughput for the fine-tuned decoder-only models.

Taken together, these results show that deployment conclusions depend strongly on the family being considered. For encoders, RoBERTa-large is not only the most accurate model but also empirically the most efficient choice under the tested setup. For encoder–decoder models, FLAN-T5-base achieves a much better accuracy–reliability trade-off than BART-base, even though it is slower. For decoder-only models, Qwen2.5-7B is slightly faster, but Llama3.1-8B offers a much stronger quality–validity profile. It should be noted, however, that the reported deployment-speed measurements are drawn from different exported pipelines,

for example full-test seq2seq generation logs versus batched $n = 1000$ decoder-only summaries. They therefore support family-level interpretation and practical trade-off analysis, but should not be read as a perfectly symmetric cross-family hardware benchmark.

Overall, the efficiency results reinforce the ranking observed in the accuracy analysis rather than reversing it. RoBERTa-large delivers the best encoder-side balance of quality and cost, FLAN-T5-base remains the strongest generative model despite being slower than BART-base, and Llama3.1-8B justifies its slightly higher latency through much better end-to-end extraction reliability than Qwen2.5-7B. In practical terms, speed alone is not the deciding factor in this study: the most useful deployment candidates are the models that combine strong extraction quality with stable structured outputs.

5.3 RQ2: Fine-Tuning versus In-Context Prompting

5.3.1 End-to-End Comparison of Fine-Tuning and ICL

RQ2 compares parameter-updating adaptation with parameter-free prompting under the same ASC-PIE extraction task. In practice, this comparison is instantiated within the decoder-only family, since in-context learning is evaluated on the instruction-following Llama3.1-8B and Qwen2.5-7B models, while their supervised counterparts are the QLoRA fine-tuned decoder-only baselines from RQ1. Because the fine-tuned and ICL results are reported on different standardized subsets ($n = 1000$ for fine-tuning and $n = 500$ for ICL), and because the exact ICL prompting protocols, shared decoding controls, and evaluation settings used in these experiments are summarized in Appendix A, Table A.2, the comparison should be interpreted as a controlled contrast between adaptation modes rather than as a perfectly symmetric score ranking. Table 5.2 therefore reports the fine-tuned models together with the best ICL configuration for each model and protocol, while preserving the evaluation subset size explicitly.

Table 5.2: Fine-tuning versus best ICL configurations for the decoder-only models. Fine-tuned runs are evaluated on $n = 1000$, whereas ICL runs are evaluated on $n = 500$.

Backbone	Mode	Protocol	Shots	N	Valid	Strict F1	Norm. F1	Latency (s)
Qwen2.5-7B								
	Fine-tuned	JSON	—	1000	0.543	0.4003	0.4008	—
	ICL (best)	JSON	3	500	0.906	0.7085	0.7392	26.7898
	ICL (best)	KV	3	500	**0.950**	**0.7782**	**0.8118**	1.7608
Llama3.1-8B								
	Fine-tuned	JSON	—	1000	0.814	0.6896	0.6899	—
	ICL (best)	JSON	5	500	0.856	0.7047	0.7238	3.0132
	ICL (best)	KV	3	500	**0.914**	**0.7253**	**0.7453**	1.5692

Several observations follow directly from Table 5.2. First, prompting is not a weak fallback baseline in this study. For both decoder-only models, the strongest ICL settings, especially under the KV protocol, reach substantially higher strict F1 than the corresponding fine-tuned decoder-only baselines on their respective evaluation subsets. This is particularly striking for Qwen2.5-7B, whose fine-tuned result remains weak, while its best KV ICL configuration reaches a strong strict F1 of 0.7782 together with a validity rate of 0.950.

Second, the protocol choice matters as much as the model choice. For both Llama and Qwen, the best KV configuration is stronger than the best JSON configuration, both in strict F1 and in practical latency. The difference is especially pronounced for Qwen2.5-7B: its best JSON ICL setting is much slower and still weaker than its best KV setting. This indicates that a lightweight key–value prompting interface is more effective than strict JSON prompting for parameter-free extraction under the current setup.

Third, the relative comparison between fine-tuning and ICL differs from the pattern observed in RQ1. Under supervised adaptation, decoder-only models lagged behind the strongest encoder and encoder–decoder systems. Under prompting, however, these same decoder-only backbones become much more competitive within their own family, especially when the prompt format is simple and the number of demonstrations is chosen carefully. In other words, the decoder-only models appear to benefit more from well-designed in-context conditioning than from the particular fine-tuning setup used in this thesis.

At the same time, the comparison should remain properly qualified. The fine-tuned and ICL results are not evaluated on identical subset sizes, and the ICL configurations also trade performance against prompt length and inference cost, which are analyzed more directly in the next two subsections. The main conclusion of this first RQ2 comparison is therefore not that ICL universally dominates fine-tuning, but that parameter-free prompting is a strong and practically relevant alternative for decoder-only PII extraction when the prompting protocol is chosen well.

Overall, the end-to-end comparison shows that ICL is a serious competing adaptation mode for the decoder-only models in this thesis. The best KV prompting configurations outperform the corresponding fine-tuned decoder-only baselines on their reported subsets, while also maintaining strong validity. This means that, for instruction-following extraction, the choice between fine-tuning and prompting cannot be reduced to accuracy alone and must be interpreted together with protocol design and inference cost.

5.3.2 Effect of Shot Count and Prompting Protocol

The ICL results show that performance is shaped by two interacting factors: the number of demonstrations and the output protocol. Figure 5.8 summarizes the strict F1 trends across shot counts for the JSON and KV prompting variants on both Llama3.1-8B and Qwen2.5-7B. Across all four configurations, the dominant pattern is a marked improvement from low-shot prompting to 3-shot prompting, followed by either saturation or slight degradation at 5 shots. This indicates that adding a small number of demonstrations is highly beneficial, but that increasing the prompt further does not guarantee continued gains.

For Llama3.1-8B, both protocols behave similarly at higher shot counts, but the KV setting is consistently slightly stronger than JSON. Under JSON prompting, strict F1 rises from 0.3490 at 0-shot and 0.3455 at 1-shot to 0.6997 at 3-shot and 0.7047 at 5-shot. Under KV prompting, the corresponding progression is 0.2874, 0.3558, 0.7253, and 0.7212. Thus,

the main gain comes from moving to 3-shot prompting, while the difference between 3-shot and 5-shot is marginal. In practical terms, Llama appears to benefit from moderate in-context guidance, but not from substantially longer prompts.

Qwen2.5-7B shows the same broad pattern but with larger protocol differences. Under JSON prompting, strict F1 moves from 0.4668 at 0-shot to 0.4388 at 1-shot, then jumps to 0.7085 at 3-shot before falling slightly to 0.6751 at 5-shot. Under KV prompting, the progression is 0.4466, 0.3746, 0.7782, and 0.7542. The highest overall ICL result in this study is therefore obtained by Qwen2.5-7B with KV prompting at 3 shots. This also reinforces an important qualitative point: the best ICL configuration is not produced by the most constrained or the longest prompt, but by a moderate number of demonstrations under the simpler output interface.

Figure 5.8: Strict F1 across shot counts for the JSON and KV ICL protocols.

The protocol effect becomes even clearer when validity is considered. Figure 5.9 shows that JSON prompting does not automatically dominate KV prompting in format compliance, despite its more explicitly structured output. For Llama3.1-8B, JSON validity remains relatively stable between 0.836 and 0.896, while KV validity ranges from 0.814 to 0.926. For

Qwen2.5-7B, JSON validity is high at all shot counts, peaking at 0.964 in the 0-shot case, but KV validity becomes even stronger once demonstrations are introduced, reaching 0.950 at 3-shot and 0.958 at 5-shot. This shows that the simpler KV protocol is not only more effective for extraction quality, but can also be more reliable in end-to-end formatting.

At the same time, high validity alone is not sufficient. Several low-performing settings still exhibit strong validity, which means that the models are often able to produce formally acceptable outputs even when the extracted entities are incomplete or inaccurate. The most important example is Qwen2.5-7B under JSON 0-shot prompting: validity is very high, yet strict F1 remains far below the best ICL settings. This confirms that end-to-end evaluation must consider both format compliance and extraction quality rather than assuming that one serves as a proxy for the other.

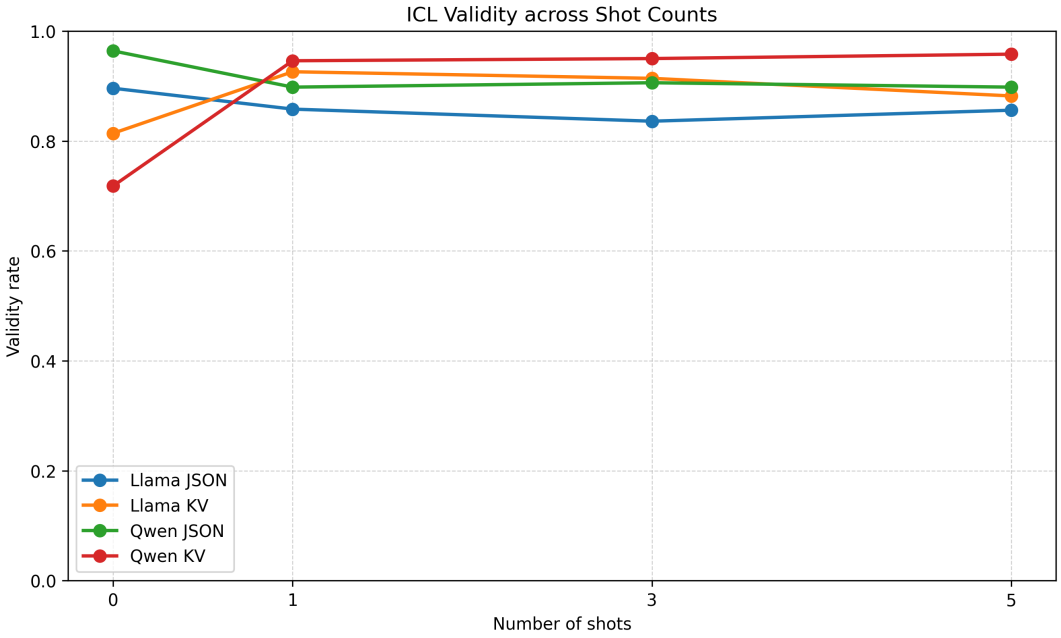

Figure 5.9: Validity rate across shot counts for the JSON and KV ICL protocols.

Taken together, these results show that shot count and prompting protocol jointly determine ICL effectiveness. The common trend across both models is that 3-shot prompting provides the most reliable improvement, while 5-shot prompting offers little additional benefit and can slightly reduce strict F1. In parallel, the KV protocol consistently provides a

better overall balance of extraction quality and validity than the JSON protocol, especially for Qwen2.5-7B. The implication is that effective ICL for PII extraction depends less on maximizing prompt size and more on choosing a compact demonstration regime together with an output format that reduces structural overhead.

Overall, the shot-sweep results show that ICL performance improves substantially up to 3 shots, but does not continue to improve monotonically beyond that point. Across both Llama3.1-8B and Qwen2.5-7B, the KV protocol is generally stronger than JSON in both strict F1 and practical reliability, with Qwen2.5-7B KV at 3 shots emerging as the strongest ICL configuration in this study.

5.3.3 Cost, Validity, and Practical Trade-Offs

The results above show that ICL effectiveness cannot be interpreted from extraction quality alone. In practice, each prompting configuration also incurs a different latency cost and a different level of structural reliability, so the most accurate setting is not necessarily the most operationally preferable. Figure 5.10 therefore summarizes the trade-off between average latency and strict F1 across all ICL configurations.

A first clear pattern is that the KV protocol dominates JSON in practical efficiency. For both models, KV prompting achieves comparable or better extraction quality at substantially lower latency. This difference is especially pronounced for Qwen2.5-7B. Under JSON prompting, latency increases sharply as the number of shots grows, reaching more than 50 seconds on average at 5-shot while strict F1 remains below the best KV configuration. By contrast, Qwen2.5-7B with KV prompting reaches the best overall ICL strict F1 in this study (0.7782 at 3-shot) with an average latency of only 1.76 s. This is a strong indication that strict JSON prompting introduces considerable structural overhead for Qwen without delivering a corresponding gain in end-to-end extraction quality.

Llama3.1-8B shows the same qualitative trend, although the gap between JSON and KV

is smaller than for Qwen. The strongest Llama KV setting, at 3 shots, reaches a strict F1 of 0.7253 with average latency 1.57 s, while the strongest JSON setting, at 5 shots, reaches 0.7047 with average latency 3.01 s. The KV protocol therefore again provides a better accuracy–latency balance. More broadly, the Llama points are concentrated in a much tighter latency band than the Qwen JSON points, which suggests that Llama is less sensitive to prompt-length growth in absolute runtime, even though KV remains the more attractive protocol.

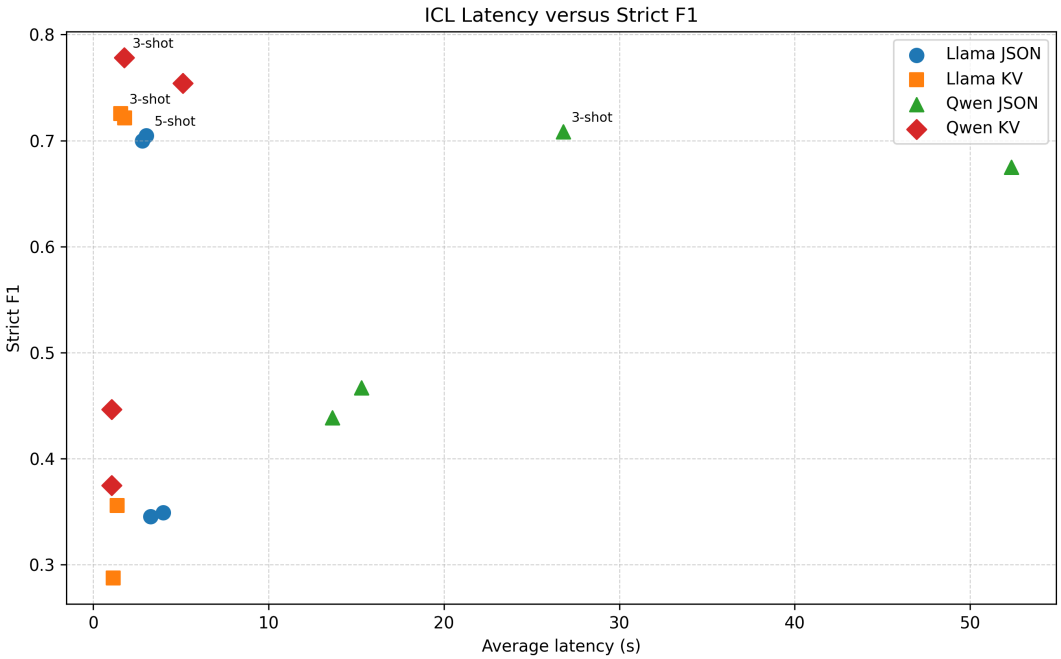

Figure 5.10: Average latency versus strict F1 across ICL configurations.

These results also reinforce that validity must be interpreted together with cost. A setting can produce highly valid outputs while still being inefficient or suboptimal in extraction quality. For example, several JSON configurations maintain strong validity, but this does not compensate for their higher latency and weaker strict F1 relative to the best KV settings. Conversely, the strongest KV configurations combine three desirable properties at once: high validity, strong extraction performance, and low latency. In that sense, the practical winner under ICL is not simply the most accurate prompt, but the one that lies closest to the best quality–cost frontier.

A second practical implication concerns scaling behavior with shots. The earlier subsection showed that 3-shot prompting is usually the best accuracy setting. The present latency analysis adds an important qualification: moving from 3 shots to 5 shots not only fails to improve strict F1 consistently, but also increases runtime. This means that longer prompts can be doubly disadvantageous, since they may reduce deployment efficiency without yielding better extraction. Under a deployment-oriented interpretation, 3-shot prompting is therefore a more defensible operating point than 5-shot prompting for both models.

Taken together, the cost and validity analysis sharpens the answer to RQ2. Prompting can be a strong alternative to fine-tuning for decoder-only extraction, but only when the prompting interface is chosen carefully. Simpler structured outputs and moderate demonstration counts yield the strongest balance of quality, reliability, and cost. More constrained or longer prompting schemes can remain usable, but they are not automatically better and may be operationally inferior despite acceptable validity.

Overall, the practical ICL trade-off is governed more by protocol design than by prompt length alone. KV prompting consistently offers a better balance of latency, validity, and strict F1 than JSON prompting, and 3-shot prompting emerges as the most attractive operating point for both models. In deployment terms, the best ICL setting is therefore not the longest or most constrained prompt, but the one that achieves strong extraction quality with minimal structural and runtime overhead.

5.4 RQ3: Class-Incremental Learning, Forgetting, and Mitigation

5.4.1 Stage-Wise Forgetting under Sequential Fine-Tuning

We begin RQ3 with the baseline sequential fine-tuning setting, which updates the selected `RoBERTa-large` backbone across the three class-incremental stages without any explicit forgetting-mitigation mechanism. This baseline exposes the poisoned-gradient setting directly: once a new stage is introduced, previously learned entity tokens remain in the text

but are relabeled as `O` in the current training supervision.

Table 5.3 reports the stage-restricted matrix for this baseline. After Stage 0, the model performs strongly on the restricted Stage 0 test view, reaching an F1 of 0.9630. After Stage 1, restricted Stage 0 performance collapses to 0.0000, while Stage 1 rises to 0.9820. The same pattern repeats at Stage 2: the model reaches 0.9944 on the current restricted test view, but both earlier stages fall to 0.0000. In the restricted view, sequential fine-tuning therefore behaves as near-complete stage replacement.

Table 5.3: Stage-restricted F1 matrix $R[t, s]$ for the baseline sequential fine-tuning strategy. Rows indicate the training stage completed, and columns indicate the restricted stage-specific evaluation view.

After training stage	**Eval Stage 0**	**Eval Stage 1**	**Eval Stage 2**
Stage 0	0.9630	0.0000	0.0000
Stage 1	0.0000	0.9820	0.0000
Stage 2	0.0000	0.0000	0.9944

The cumulative mixed view confirms the same conclusion under a more realistic evaluation setting. After Stage 1, cumulative Stage 0 performance already falls to 0.0000, and cumulative Stage 1 reaches only 0.3381. After Stage 2, the final cumulative row becomes [0.0000, 0.0000, 0.4371], showing that the baseline retains no usable performance on the earlier cumulative stages.

This behavior is also reflected in the derived CL metrics. The baseline reaches ACC = 0.1457, Forgetting = 0.6506, and BWT = −0.6506, indicating severe backward degradation by the end of training. At the same time, the restricted diagonal remains high across stages, showing that the baseline retains strong plasticity even as stability collapses.

An important nuance is that this catastrophic behavior is not visible from token-level accuracy alone. Even when earlier-stage F1 collapses, token accuracy can remain non-trivial because the staged data are dominated by non-entity tokens. For this reason, entity-level F1 is the critical measure in the continual-learning setting: it captures whether previously learned PII categories are still extractable, rather than whether the model continues to predict background tokens correctly.

Overall, the baseline sequential fine-tuning setting demonstrates catastrophic forgetting in its strongest form. In the stage-restricted view, each new stage is learned well while earlier stages collapse to zero. In the cumulative view, final retained performance is also very poor, confirming that naive sequential updating preserves plasticity but not usable prior extraction capability.

5.4.2 Replay, Distillation, and SPRINT-PP

Having established that naive sequential fine-tuning leads to near-complete loss of previously learned stages, we now compare the three mitigation strategies under the same staged protocol: replay, distillation, and the proposed SPRINT-PP method. Table 5.4 reports the main continual-learning summary metrics, while Figure 5.11 and Figure 5.12 provide compact stage-restricted and cumulative views of the three mitigation strategies. Figure 5.13 then summarizes the same comparison visually.

Table 5.4: Continual-learning summary metrics for the four update strategies.

Strategy	**ACC ↑**	**BWT ↑**	**Forgetting ↓**	**Intransigence ↓**	**Fgt_S0 ↓**	**Fgt_S1 ↓**
SPRINT-PP	**0.8350**	**-0.2075**	**0.2075**	**0.0156**	**0.2581**	**0.1569**
Distillation	0.8295	-0.2111	0.2111	0.0193	0.2645	0.1578
Replay	0.7651	-0.2726	0.2726	0.0278	0.3381	0.2070
Baseline	0.1457	-0.6506	0.6506	0.0202	0.9630	0.3381

Table 5.4 shows a clear ranking across the mitigation strategies. SPRINT-PP is the strongest overall method in this study, achieving the highest ACC (0.8350), the least forgetting (0.2075), and the best backward transfer (-0.2075). Distillation is a very close second on the same metrics, while replay remains clearly above the baseline but below both privacy-safe methods. The per-stage forgetting columns tell the same story: SPRINT-PP has the lowest forgetting on both Stage 0 and Stage 1, with distillation close behind and replay noticeably weaker. Taken together, the summary metrics indicate that SPRINT-PP provides the strongest overall balance of stability and plasticity in this study.

To make the stage-wise behavior easier to interpret, Figure 5.11 shows the three mitigation strategies as compact stage-restricted matrices. In this figure, the diagonal cells are shaded in light cyan-blue and indicate current-stage learning, while the cells below the diagonal are shaded in soft yellow and indicate earlier-stage recoverability in the restricted view. The baseline is omitted here because its collapse was already established in the previous subsection.

Replay

	0	1	2
0	0.9668	0.0000	0.0000
1	0.9481	0.9666	0.0000
2	0.8726	0.9879	0.9833

Distillation

	0	1	2
0	0.9668	0.0000	0.0000
1	0.9661	0.9793	0.0000
2	0.9635	0.9802	**0.9962**

SPRINT-PP

	0	1	2
0	0.9668	0.0000	0.0000
1	**0.9694**	**0.9907**	0.0000
2	**0.9694**	**0.9896**	0.9956

Figure 5.11: Stage-restricted F1 matrices for replay, distillation, and SPRINT-PP after each training stage. Diagonal cells represent current-stage learning, and the cells below the diagonal represent earlier-stage recoverability when evaluated in isolation.

The stage-restricted view shows that strong retention is not obtained by sacrificing current-stage learning. All three mitigation methods maintain near-ceiling restricted F1 on the final stage, and all three also preserve earlier stages substantially better than the baseline. At the same time, the ranking is still visible. SPRINT-PP is strongest on the earlier restricted stages after Stage 2, distillation is very close behind, and replay is clearly lower on Stage 0. This suggests that the main difference between the methods lies not in learning the new stage, but in how much of the earlier knowledge remains recoverable.

The cumulative mixed view is even more important, because it reflects retention under joint old-and-new label evaluation. Figure 5.12 uses the same color logic as the previous figure: the diagonal shows cumulative performance when each stage is first reached, while the cells below the diagonal show retained performance on earlier cumulative stages. This view therefore provides the main basis for assessing whether a method preserves usable prior knowledge once the label space has expanded.

Replay

	0	1	2
0	0.9668	–	–
1	0.8315	0.9521	–
2	0.6286	0.7450	0.9215

Distillation

	0	1	2
0	0.9668	–	–
1	0.8542	0.9688	–
2	0.7023	0.8111	0.9751

SPRINT-PP

	0	1	2
0	0.9668	–	–
1	**0.8600**	**0.9737**	–
2	**0.7086**	**0.8168**	**0.9797**

Figure 5.12: Cumulative mixed F1 matrices for replay, distillation, and SPRINT-PP after each training stage. The cells below the diagonal provide the primary retention view because old and new labels are evaluated jointly.

Figure 5.12 makes the final ranking especially clear. SPRINT-PP retains the strongest cumulative performance on both earlier stages and also remains strongest on the final stage. Distillation is consistently close behind, whereas replay is noticeably lower on all three cumulative values. The gap between SPRINT-PP and distillation is modest but consistent, while the gap between both privacy-safe methods and replay is substantially larger.

A particularly important comparison concerns the two privacy-safe methods. Distillation performs strongly and remains close to SPRINT-PP on both aggregate metrics and stage-wise retention. However, SPRINT-PP remains slightly better on every main summary metric reported in Table 5.4. It also provides an efficiency advantage, using 25.6% less training time than distillation, equivalently making training 34.4% faster. This strengthens the practical case for SPRINT-PP: it is not only the strongest privacy-safe method in accuracy and retention, but also the more efficient one.

Replay still provides a useful empirical reference, but it does not define the strongest upper bound in this study. It improves substantially over the baseline, yet remains weaker than both SPRINT-PP and distillation on final cumulative retention while also requiring stored raw historical examples.

These findings are summarized visually in Figure 5.13. The dashboard shows the same overall picture from a different angle: SPRINT-PP occupies the strongest position overall, distillation is the closest competitor, replay remains effective but lower, and the baseline shows by far the weakest retention profile.

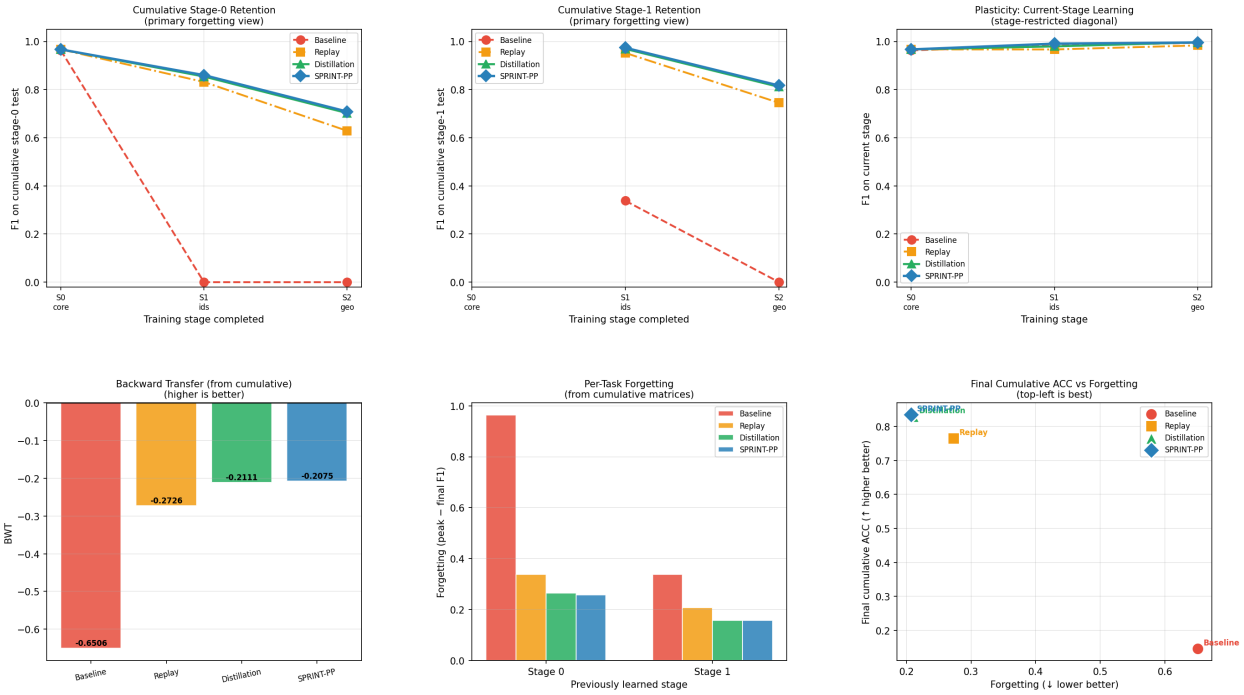

Figure 5.13: Continual-learning comparison across baseline sequential fine-tuning, replay, distillation, and SPRINT-PP.

Taken together, these results answer the mitigation part of RQ3 clearly. SPRINT-PP is the strongest method in the continual-learning evaluation, achieving the best balance of retention and plasticity without storing raw historical examples. Distillation is a close second and remains a strong privacy-safe baseline. Replay still improves substantially over naive sequential fine-tuning, but in this study it is weaker than both privacy-safe methods while also requiring raw-example storage.

Overall, SPRINT-PP is the strongest continual-learning strategy in this evaluation. It achieves the best cumulative accuracy, the least forgetting, and near-ceiling plasticity, while remaining privacy-safe. Distillation is a close second, whereas replay, although still effective, is weaker than both privacy-safe methods and requires storing raw historical examples.

5.4.3 Continual-Learning Metrics and Per-Class Retention

The aggregate continual-learning metrics in the previous subsection are useful for ranking the strategies, but they do not reveal *which* PII types are retained and which are lost. Table 5.5 therefore reports final per-class F1 for a small representative subset of entity types spanning the earlier stages and the final stage.

Table 5.5: Selected final per-class F1 scores after Stage 2 training. The subset highlights representative earlier-stage and final-stage categories.

Stage	Type	Baseline	Replay	Distillation	SPRINT-PP
S0	NAME	0.0000	0.9429	**0.9701**	0.9696
S0	EMAIL	0.0000	0.7832	0.9951	**0.9977**
S0	LOCATION	0.0000	0.6280	0.8325	**0.8397**
S1	CREDITCARD	0.0000	0.9921	0.9891	**0.9938**
S1	DRIVER-LICENCE	0.0000	0.8060	0.7500	**0.8889**
S1	NATIONALID	0.0000	**0.6667**	0.5000	0.4000
S2	COUNTRY	0.5000	0.2857	**0.7500**	0.4000

The first clear pattern is that the baseline still collapses on the earlier stages at the class level. All representative Stage 0 and Stage 1 types in Table 5.5 fall to 0.0000, confirming that the baseline does not preserve usable old-class knowledge by the end of training.

Among the mitigation methods, SPRINT-PP and distillation are broadly the strongest on the older classes. Both retain very high F1 on NAME and EMAIL, and both substantially outperform replay on LOCATION. SPRINT-PP is also strongest on CREDITCARD and DRIVER-LICENCE, while distillation is marginally strongest on NAME and clearly strongest on the selected Stage 2 type COUNTRY. Replay remains competitive, but its profile is less uniform and is clearly weaker on several Stage 0 categories.

At the same time, the class-level view shows that the overall ranking does not mean that one method dominates every type. Replay remains strongest on NATIONALID, for example, while distillation is strongest on COUNTRY. The value of this subsection is therefore not to overturn the aggregate ranking from the previous subsection, but to refine it: SPRINT-PP is still the strongest overall method, distillation is a close second, and replay remains more uneven across the retained label space.

Overall, the per-class results support the aggregate RQ3 findings while adding an important nuance. SPRINT-PP and distillation provide the broadest old-stage retention, whereas replay remains more uneven across classes despite still performing well on selected structured identifiers such as `NATIONALID`. This confirms that the advantage of SPRINT-PP is not tied to a single category, but to a stronger overall retention profile across the evolving label space.

5.5 RQ4: Effect of Corpus Design and Synthetic Augmentation

5.5.1 Overall Effect of Full ASC-PIE versus Public-Only Training

To answer RQ4, we perform a controlled corpus ablation on the strongest supervised model identified in RQ1, namely RoBERTa-large. The goal is to isolate the effect of ASC-PIE corpus design while holding the backbone and evaluation protocol fixed. Specifically, we compare a *public-only* training condition, constructed from the public-source component alone, against the full ASC-PIE training condition, which adds the thesis-specific consolidation and synthetic augmentation components. Both settings are evaluated under the same unified $n = 1000$ comparison subset used elsewhere in Chapter 6.

Table 5.6 and Figure 5.14 show that the difference between the two training conditions is very large. Under public-only training, RoBERTa-large reaches a strict F1 of 0.4152 and a normalized F1 of 0.4157. Under full ASC-PIE training, the same backbone reaches a strict F1 of 0.9920 and a normalized F1 of 0.9921. The gain is therefore not marginal, but transformational: the full ASC-PIE corpus improves strict F1 by approximately 0.577 absolute points while also eliminating the large recall deficit observed in the public-only condition.

The precision–recall breakdown helps clarify the source of this gap. The public-only model reaches a moderate strict precision of 0.6395, but its strict recall falls to 0.3073, indicating that the main weakness is not uncontrolled over-prediction, but failure to recover

a large fraction of the gold entities. By contrast, the full ASC-PIE model reaches both very high precision (0.9893) and very high recall (0.9947). This means that the benefit of the full corpus is not simply better formatting or cleaner output realization, but substantially better coverage of the target PII space.

The near identity between strict and normalized F1 in both settings provides an additional important interpretation. Because normalized F1 differs only trivially from strict F1, the observed gain cannot be attributed mainly to superficial formatting differences such as casing, whitespace, or punctuation. Instead, the corpus effect reflects a true improvement in extraction quality under exact matching. In other words, the unified schema design and synthetic augmentation do not merely make the outputs cleaner; they materially improve whether the correct entities are found at all.

A final point is that validity remains constant at 1.000 in both conditions, since this comparison is conducted within the encoder family. The ablation therefore isolates the effect of corpus design more cleanly than a comparison involving structured-generation validity constraints. Under a fixed encoder architecture, the difference between public-only and full ASC-PIE training is overwhelmingly a difference in extraction coverage and recall, not a difference in output format compliance.

Taken together, these results provide a direct answer to the first part of RQ4. The full ASC-PIE construction, including consolidation under a unified schema and the addition of the thesis-specific synthetic component, yields a dramatic improvement over public-only training even when the model architecture is held fixed. This establishes that corpus design in this thesis is not a minor preprocessing choice, but a central determinant of end-to-end extraction quality.

Table 5.6: Overall comparison between public-only and full ASC-PIE training using the same RoBERTa-large backbone on the unified $n = 1000$ comparison subset.

Training condition	**Valid**	**Strict P**	**Strict R**	**Strict F1**	**Norm. F1**
Public-only	1.000	0.6395	0.3073	0.4152	0.4157
Full ASC-PIE	1.000	0.9893	0.9947	0.9920	0.9921

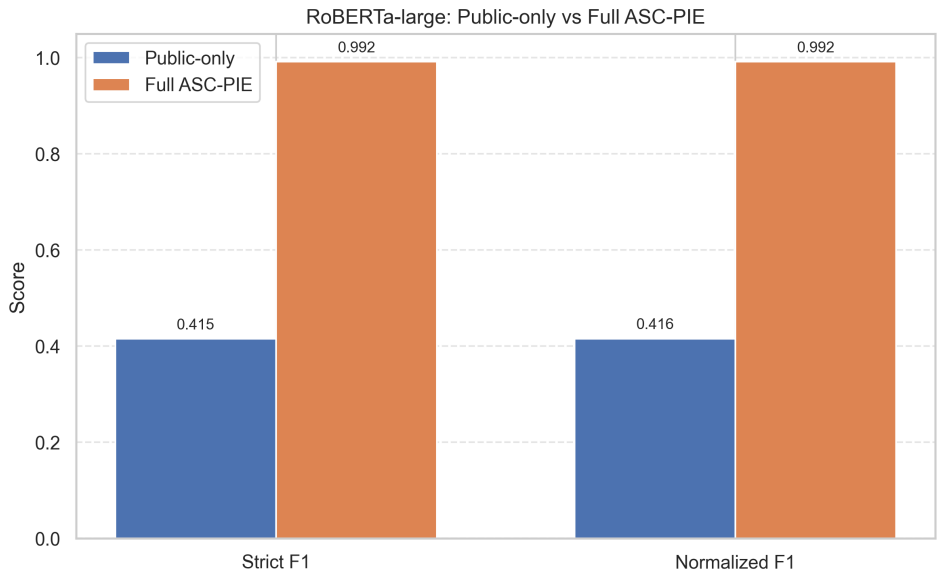

Figure 5.14: Overall comparison of RoBERTa-large trained on the public-only corpus versus the full ASC-PIE corpus.

Overall, the corpus ablation shows that the full ASC-PIE design contributes far more than a small incremental gain. With the backbone held fixed, public-only training remains recall-limited and achieves only moderate extraction quality, whereas full ASC-PIE training reaches near-ceiling exact-match performance. This shows that unified corpus construction and synthetic augmentation are central to the performance gains reported in this thesis.

5.5.2 Per-Type Gains for Rare and Structured PII

The overall gap reported in the previous subsection is not distributed uniformly across the label space. A more detailed view shows that the full ASC-PIE corpus improves performance on many PII categories simultaneously, but the size of the gain depends strongly on the type. Table 5.7 summarizes representative per-type strict F1 changes on the matched $n = 1000$ comparison subset.

A first important observation is that the gains are not confined only to obscure edge cases. Very large improvements appear on several practically important categories, including

JOBTITLE, EMAIL, SIN, CITY, NAME, and STATE. For example, JOBTITLE rises from 0.0000 under public-only training to 0.9982 under full ASC-PIE training, while EMAIL rises from 0.0786 to 0.9940, SIN from 0.2236 to 0.9980, and CITY from 0.2826 to 0.9995. These are not small corrections, but near-complete recoveries of categories that remain only weakly learned under the public-only condition.

A second observation is that the benefit is especially informative for rare or semantically difficult categories. COUNTRY, despite its low support on the comparison subset, improves from 0.8000 to 1.0000. LOCATION improves from 0.4477 to 0.9102, showing that the full ASC-PIE design also helps on boundary-sensitive span categories that are harder than regular alphanumeric identifiers. At the same time, some highly regular structured types such as CREDITCARD and DRIVER-LICENCE change only slightly, from 0.9786 to 0.9911 and from 0.9744 to 0.9756, respectively. This suggests that these regular formats are already relatively learnable from the public component alone and are not the main beneficiaries of the full corpus expansion.

A third observation is that the corpus effect is not monotonic across every low-support type. TIME and TAXNUM are slightly lower under full ASC-PIE than under public-only training on this subset. However, these categories have extremely small support in the $n = 1000$ comparison set, with only 5 TIME mentions and 3 TAXNUM mentions, so their differences should not be over-interpreted. In this case, the apparent decline is better understood as evaluation instability under very small sample counts rather than as evidence against the broader corpus-design effect.

Taken together, the per-type results sharpen the answer to RQ4. The contribution of the full ASC-PIE construction is not limited to a few synthetic edge cases, nor is it restricted to one narrow category of identifiers. Instead, the gain is broad but uneven: it is largest on categories that require richer lexical diversity, better contextual discrimination, or stronger coverage than the public-only condition provides, while already saturated regular formats benefit much less. This is exactly the pattern one would expect if systematic corpus con-

solidation and synthetic augmentation improve both label-space coverage and contextual generalization.

Table 5.7: Selected per-type strict F1 changes between public-only and full ASC-PIE training on the matched $n = 1000$ comparison subset.

Type	Public-only F1	Full ASC-PIE F1	Gain
JOBTITLE	0.0000	0.9982	+0.9982
EMAIL	0.0786	0.9940	+0.9154
SIN	0.2236	0.9980	+0.7744
CITY	0.2826	0.9995	+0.7169
NAME	0.2716	0.9824	+0.7108
STATE	0.2891	0.9986	+0.7095
LOCATION	0.4477	0.9102	+0.4625
PHONENUM	0.7340	0.9938	+0.2598
COUNTRY	0.8000	1.0000	+0.2000
CREDITCARD	0.9786	0.9911	+0.0125
DRIVER-LICENCE	0.9744	0.9756	+0.0012
TIME	1.0000	0.8333	-0.1667
TAXNUM	1.0000	0.7500	-0.2500

Overall, the per-type analysis shows that the benefit of full ASC-PIE training is broad but not uniform. The largest gains appear on categories that depend on richer lexical coverage, contextual diversity, or improved label-space consolidation, whereas highly regular structured identifiers were already close to saturation under public-only training. This indicates that the full corpus contributes most where public-only supervision remains coverage-limited rather than where the extraction pattern is already easy and regular.

5.5.3 Interpretation Relative to Earlier Model-Family Results

The corpus ablation in this section is conducted only on RoBERTa-large, so it should not be interpreted as a direct full-factorial comparison across all model families and adaptation modes. Nevertheless, it provides an important interpretive lens for the earlier findings in RQ1 and RQ2. In particular, it shows that a large portion of extraction performance depends

not only on the choice of architecture or adaptation strategy, but also on whether the corpus provides sufficient label-space coverage and type-specific support in the first place.

Figure 5.15 makes this point especially clearly by separating types that were seen during public-only training from types that were not. For types unseen in the public-only training condition, the public-only model achieves an average per-type F1 of 0.000, whereas the full ASC-PIE model reaches 1.000. Even among types that were seen in public-only training, the average per-type F1 still rises substantially, from 0.519 to 0.884. This means that the benefit of the full corpus is not limited to introducing entirely missing labels. It also improves learning on already-seen categories by providing stronger support, broader contextual variation, and better alignment under the unified schema.

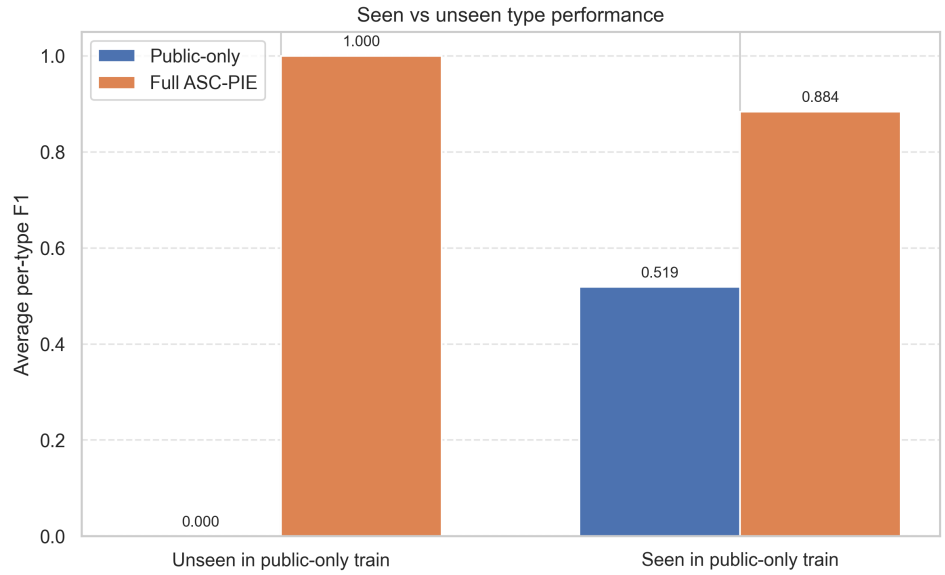

Figure 5.15: Average per-type strict F1 for types seen and unseen in the public-only training condition, comparing public-only and full ASC-PIE training.

This finding helps contextualize the earlier model-family comparison in RQ1. RoBERTa-large and FLAN-T5-base emerged as the strongest supervised models, but the present ablation shows that architecture alone does not explain high performance. A substantial part of the final result depends on whether the model is trained on a corpus that adequately covers both frequent and difficult PII categories. Put differently, the near-ceiling performance of

the best models in RQ1 should be understood as the result of a strong modeling choice operating on top of a substantially strengthened training corpus, not as the effect of architecture alone.

The same logic also informs the interpretation of RQ2. In-context prompting improved decoder-only performance substantially relative to the fine-tuned decoder-only baselines, especially under the KV protocol. However, the present RQ4 findings suggest that adaptation strategy and corpus design should not be treated as independent concerns. Prompting can compensate for some weaknesses of supervised adaptation in decoder-only models, but it does not remove the underlying importance of label coverage, schema consistency, and exposure to rare or difficult types during corpus construction. In this sense, corpus design remains foundational even when the adaptation mode changes.

Finally, the RQ4 results also help explain why certain categories remained difficult in the per-type analyses from RQ1. The strongest gains from full ASC-PIE appear on types that are either sparsely supported, lexically diverse, or contextually ambiguous under the public-only condition. This is consistent with the earlier observation that long-tail and boundary-sensitive categories are the main remaining challenge in PII extraction. The corpus construction strategy adopted in this thesis therefore operates as a direct mitigation of that challenge: it expands coverage where the public-only condition is weakest, rather than merely inflating already-easy regular formats.

Taken together, these findings refine the answer to RQ4. The main contribution of ASC-PIE is not only that it adds synthetic examples, but that it restructures the training signal itself through corpus consolidation, unified labeling, and coverage expansion. Earlier results in the thesis showed which models and adaptation modes perform best under this setting. The present subsection shows why that setting matters: without the full corpus design, even a strong encoder backbone remains severely limited on both unseen and under-supported types.

Overall, the RQ4 ablation suggests that corpus design is a foundational factor underlying the earlier model-family results rather than a minor preprocessing detail. The strongest architectures in this thesis perform well not only because of their modeling capacity, but because ASC-PIE supplies the label coverage, schema consistency, and difficult-type support needed to realize that capacity. This makes corpus construction and synthetic augmentation central to the thesis contribution, not merely supportive implementation choices.

5.6 Summary of Key Findings and Cross-Cutting Observations

This chapter answered the four research questions through a unified comparison of model families, adaptation modes, continual-learning behavior, and corpus design.

For RQ1, the results showed a clear hierarchy under supervised fine-tuning. RoBERTa-large achieved the strongest overall extraction performance, with FLAN-T5-base as the closest competing generative baseline. The remaining encoders formed a second tier, while decoder-only models and BART-base were limited mainly by recall and, in the generative settings, by lower structural reliability.

For RQ2, the results showed that in-context learning is a strong practical alternative to supervised fine-tuning within the decoder-only family. The best KV prompting configurations outperformed the corresponding fine-tuned decoder-only baselines on their reported subsets, while also maintaining strong validity. Across both Llama3.1-8B and Qwen2.5-7B, performance improved substantially up to 3-shot prompting, but did not continue to improve beyond that point, and KV prompting consistently offered a better quality–cost balance than JSON prompting.

For RQ3, the staged continual-learning study confirmed that catastrophic forgetting is severe under naïve sequential fine-tuning. Among the mitigation strategies, SPRINT-PP achieved the strongest overall continual-learning performance, with distillation as a close

second and replay clearly above the baseline but below both privacy-safe methods. SPRINT-PP achieved the best balance of retention and plasticity without relying on raw replay, while also providing a training-time advantage over distillation. These results show that privacy-safe continual adaptation is not only feasible, but can outperform replay under the present staged protocol.

For RQ4, the corpus ablation showed that ASC-PIE corpus design is a major contributor to performance rather than a minor preprocessing detail. Under the same RoBERTa-large backbone, public-only training remained strongly recall-limited, whereas full ASC-PIE training reached near-ceiling exact-match performance. The gains were especially large for under-supported, context-sensitive, and previously unseen types, showing that corpus consolidation and synthetic augmentation materially strengthened the learning signal.

Several cross-cutting findings also emerge. Output validity is essential for practical generative extraction, since strong entity recognition is useful only when outputs remain consistently parseable. Deployment trade-offs depend strongly on both model family and interface choice: RoBERTa-large provided the strongest encoder-side balance, FLAN-T5-base was the most reliable supervised generative model, and KV prompting was the most promising decoder-only ICL setting. Overall, the chapter shows that extraction quality in PII-aware NER is jointly determined by architecture, adaptation strategy, and corpus design rather than by any one factor alone.

Taken together, the results establish the main empirical claims of the thesis. Encoder-based token classification remains the strongest supervised baseline overall, prompting can be a competitive alternative for decoder-only extraction when carefully designed, privacy-safe forgetting mitigation is not only feasible but can be highly effective, and the ASC-PIE corpus itself is a major source of the gains achieved across the study. These findings provide the basis for the broader interpretation and final conclusions presented in the next chapter.

6 Conclusion

This thesis investigated PII-aware extraction under evolving task requirements through the ASC-PIE framework, a unified corpus and evaluation protocol spanning encoder-based token classification, encoder–decoder structured generation, and decoder-only instruction-following extraction. Across the supervised comparison, RoBERTa-large emerged as the strongest overall model, while FLAN-T5-base showed that structured generation can remain highly competitive when output constraints are well controlled. The prompting study further showed that in-context learning can be a strong alternative for decoder-only extraction, especially under compact KV prompting with moderate shot counts.

The staged continual-learning study confirmed that catastrophic forgetting is a serious challenge in class-incremental PII extraction. Among the evaluated mitigation strategies, SPRINT-PP achieved the strongest overall continual-learning performance, with distillation as a close second and replay above the baseline but below both privacy-safe methods. These results show that forgetting can be mitigated effectively without storing raw historical examples, and that targeted correction together with prototype-based and classifier-level preservation can support strong privacy-safe continual adaptation. The corpus ablation further showed that ASC-PIE itself is a major source of the reported gains: unified schema design, source consolidation, and synthetic augmentation materially improved extraction quality, especially for under-supported and difficult PII types.

Taken together, these findings show that robust PII-aware extraction depends on the interaction of the model family, adaptation strategy, and training-corpus design. The main contribution of this thesis is therefore not only a comparison of models, but a unified basis for studying accuracy, reliability, maintainability, and privacy-aware continual adaptation within one consistent experimental framework. Although the present study focuses on English PII extraction, the ASC-PIE framework is sufficiently general to support extension to other languages and related structured extraction tasks under evolving requirements.

6.1 Limitations

ASC-PIE focuses on English PII extraction, reflecting the availability and maturity of English-language source corpora needed to construct a unified benchmark under controlled schema normalization and staged evaluation. The findings should therefore not be assumed to transfer directly to multilingual or cross-lingual settings without further validation. At the same time, the overall methodology of this thesis, including schema-guided corpus consolidation, synthetic augmentation for under-supported PII types, and structured comparison across adaptation strategies, is not inherently language-specific and can be extended to other languages and related extraction settings.

The experimental comparisons are representative rather than exhaustive by design. The supervised study selects strong and practically relevant models from the encoder, encoder–decoder, and decoder-only families to establish a controlled basis for cross-family comparison under the ASC-PIE protocol. Likewise, the continual-learning study is conducted on the selected best-performing backbone so that forgetting mitigation can be analyzed in depth on a strong extraction setting rather than being diluted across many weaker configurations.

Some comparisons are intentionally standardized for control, but are not perfectly symmetric in every detail. In particular, the fine-tuning and ICL experiments use different fixed subset sizes to keep each study computationally tractable while preserving consistent within-block comparison. Similarly, deployment-speed measurements are drawn from the exported pipelines associated with each model family, which supports practical comparison but limits how strongly some quantitative differences should be generalized beyond the present setup.

SPRINT-PP substantially improves over naïve privacy-safe updating and demonstrates that forgetting can be reduced without storing raw historical examples. However, it does not eliminate the continual-learning problem entirely, and its current evaluation is still limited to the selected backbone and staged setting used in this thesis. The method should therefore be understood as a strong privacy-aware mitigation strategy whose results are promising

but still require broader validation across additional backbones, incremental protocols, and deployment conditions.

6.2 Future Work

An immediate extension of this work is to expand ASC-PIE beyond English and evaluate whether the same unified schema and staged protocol remain effective in multilingual or cross-lingual settings. Another natural direction is to broaden the architectural study by including additional strong open-weight models and alternative structured-output interfaces, especially for instruction-following extraction.

For continual learning, future work should test SPRINT-PP and other privacy-safe mitigation strategies beyond a single backbone and determine how robust the present ranking remains across different architectures and incremental settings. It would also be valuable to study more realistic continual-learning conditions that combine label growth with stronger domain drift, noisier supervision, or partially overlapping annotation policies, and to explore whether targeted correction can be strengthened further without approaching the storage demands of replay.

The corpus-design results also indicate that dataset construction deserves further study in its own right. Future work could isolate the relative contribution of schema normalization, source consolidation, and synthetic augmentation more systematically, and could investigate higher-fidelity synthetic generation strategies for difficult long-tail PII types. Together, these directions would help move PII-aware extraction toward more robust, portable, and maintainable deployment settings.

References

[1] E. McCallister, T. Grance, and K. A. Scarfone, "Guide to protecting the confidentiality of personally identifiable information (pii)," National Institute of Standards and Technology, NIST Special Publication 800-122, 2010. DOI: 10.6028/NIST.SP.800-122 [Online]. Available: https://csrc.nist.gov/pubs/sp/800/122/final

[2] *Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 (general data protection regulation)*, Official Journal of the European Union; Article 4(1) defines personal data, 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng

[3] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007. DOI: 10.1075/li.30.1.03nad

[4] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2016, pp. 260–270. [Online]. Available: https://aclanthology.org/N16-1030/

[5] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2016, pp. 1064–1074. DOI: 10.18653/v1/P16-1101 [Online]. Available: https://aclanthology.org/P16-1101/

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, arXiv:1706.03762, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423 [Online]. Available: https://aclanthology.org/N19-1423/

[8] L. Wang, R. Li, Y. Yan, Y. Yan, S. Wang, W. Wu, and W. Xu, "Instructionner: A multi-task instruction-based generative framework for few-shot ner," *arXiv:2203.03903*, 2022. DOI: 10.48550/arXiv.2203.03903 [Online]. Available: https://arxiv.org/abs/2203.03903

[9] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, et al., "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, 2022. [Online]. Available: https://arxiv.org/abs/2203.02155

[10] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24, Academic Press, 1989, pp. 109–165. DOI: 10.1016/S0079-7421(08)60536-8

[11] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017. DOI: 10.1073/pnas.1611835114 [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC5380101/

[12] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021. DOI: 10.1109/TPAMI.2021.3057446

[13] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks," *Neural Networks*, vol. 113, pp. 54–71, 2019. DOI: 10.1016/j.neunet.2019.01.012

[14] T. B. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[15] S. Wang, X. Sun, X. Li, R.-Z. Ouyang, F. Wu, T. Zhang, J. Li, and G. Wang, "Gptner: Named entity recognition via large language models," *arXiv:2304.10428*, pp. 4257–4275, 2023. DOI: 10.18653/v1/2025.findings-naacl.239 [Online]. Available: https://arxiv.org/abs/2304.10428

[16] J. Zhang, X. Liu, X. Lai, Y. Gao, S. Wang, Y. Hu, and Y. Lin, "2iner: Instructive and in-context learning on few-shot named entity recognition," in *Findings of EMNLP 2023*, Association for Computational Linguistics, 2023, pp. 3940–3951. DOI: 10.18653/v1/2023.findings-emnlp.259 [Online]. Available: https://aclanthology.org/2023.findings-emnlp.259/

[17] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "Icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5533–5542. DOI: 10.1109/CVPR.2017.587 [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/papers/Rebuffi_iCaRL_Incremental_Classifier_CVPR_2017_paper.pdf

[18] D. Zhang, W. Cong, J. Dong, Y. Yu, X. Chen, Y. Zhang, and Z. Fang, "Continual named entity recognition without catastrophic forgetting," *arXiv:2310.14541*, pp. 8186–8197, Dec. 2023. DOI: 10.18653/v1/2023.emnlp-main.509 [Online]. Available: https://arxiv.org/abs/2310.14541

[19] Ai4Privacy, *Open-pii-masking-500k-ai4privacy*, 2025. [Online]. Available: https://huggingface.co/datasets/ai4privacy/open-pii-masking-500k-ai4privacy

[20] N. Monaikul, G. Castellucci, S. Filice, and O. Rokhlenko, "Continual learning for named entity recognition," in *Proceedings of the AAAI Conference on Artificial In-*

telligence, 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17600

[21] Y. Chen, L. He, and M. Yang, "Skdner: Continual named entity recognition via span-based knowledge distillation with reinforcement learning," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore: Association for Computational Linguistics, Dec. 2023, pp. 6689–6700. DOI: 10.18653/v1/2023.emnlp-main.413 [Online]. Available: https://aclanthology.org/2023.emnlp-main.403/

[22] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, and X. Qiu, "A unified generative framework for various ner subtasks," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2021, pp. 5808–5822. [Online]. Available: https://aclanthology.org/2021.acl-long.451/

[23] C. Hokamp and Q. Liu, "Lexically constrained decoding for sequence generation using grid beam search," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2017, pp. 1535–1546. DOI: 10.18653/v1/P17-1141 [Online]. Available: https://aclanthology.org/P17-1141/

[24] E. F. T. K. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceedings of CoNLL-2003 at HLT-NAACL*, 2003, pp. 142–147. [Online]. Available: https://aclanthology.org/W03-0419/

[25] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv:2106.09685*, 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[26] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *arXiv:2305.14314*, 2023. [Online]. Available: https://arxiv.org/abs/2305.14314

[27] A. Stubbs, C. Kotfila, and Ö. Uzuner, "Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1," *Journal of Biomedical Informatics*, vol. 58, S11–S19, 2015. DOI: 10.1016/j.jbi.2015.06.007 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046415001173

[28] F. Dernoncourt, J. Y. Lee, Ö. Uzuner, and P. Szolovits, "De-identification of patient notes with recurrent neural networks," *Journal of the American Medical Informatics Association*, vol. 24, no. 3, pp. 596–606, 2017. DOI: 10.1093/jamia/ocw156 [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28040687/

[29] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012 [Online]. Available: https://aclanthology.org/D18-2012/

[30] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162 [Online]. Available: https://aclanthology.org/P16-1162/

[31] L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Third Workshop on Very Large Corpora*, 1995. [Online]. Available: https://aclanthology.org/W95-0107/

[32] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703 [Online]. Available: https://aclanthology.org/2020.acl-main.703/

[33] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: https://jmlr.org/papers/v21/20-074.html

[34] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://arxiv.org/abs/1711.02173

[35] D. Pruthi, B. Dhingra, and Z. C. Lipton, "Combating adversarial misspellings with robust word recognition," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019, pp. 5582–5591. DOI: 10.18653/v1/P19-1561 [Online]. Available: https://aclanthology.org/P19-1561/

[36] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019, pp. 3645–3650. DOI: 10.18653/v1/P19-1355 [Online]. Available: https://aclanthology.org/P19-1355/

[37] D. Nozza, P. Manchanda, E. Fersini, M. Palmonari, and E. Messina, "Learningtoadapt with word embeddings: Domain adaptation of named entity recognition systems," *Information Processing & Management*, vol. 58, no. 3, p. 102 537, 2021. DOI: 10.1016/j.

ipm.2021.102537 [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0306457321000455

[38] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[39] I. Neamatullah, M. M. Douglass, L.-W. H. Lehman, A. Reisner, M. Villarroel, W. Long, P. Szolovits, G. B. Moody, R. G. Mark, and G. D. Clifford, "Automated de-identification of free-text medical records," *BMC Medical Informatics and Decision Making*, vol. 8, p. 32, 2008. DOI: 10.1186/1472-6947-8-32

[40] Gretel.ai, *Gretel-pii-masking-en-v1*, 2024. [Online]. Available: https://huggingface.co/datasets/gretelai/gretel-pii-masking-en-v1

[41] Z. Li and D. Hoiem, "Learning without forgetting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 614–629. [Online]. Available: https://arxiv.org/abs/1606.09282

[42] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. S. Torr, and M. Ranzato, *On tiny episodic memories in continual learning*, arXiv:1902.10486, 2019. [Online]. Available: https://arxiv.org/abs/1902.10486

[43] F. Cermelli, M. Mancini, S. Rota Bulò, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9233–9242. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Cermelli_Modeling_the_Background_for_Incremental_Learning_in_Semantic_Segmentation_CVPR_2020_paper.html

[44] E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou, *What learning algorithm is in-context learning? investigations with linear models*, arXiv:2211.15661, 2022. [Online]. Available: https://arxiv.org/abs/2211.15661

[45] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 12 684–12 693. [Online]. Available: https://proceedings.mlr.press/v139/zhao21c.html

[46] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2022, pp. 8086–8098. DOI: 10.18653/v1/2022.acl-long.556 [Online]. Available: https://aclanthology.org/2022.acl-long.556/

[47] J. Chen, Y. Lu, H. Lin, J. Lou, W. Jia, D. Dai, H. Wu, B. Cao, X. Han, and L. Sun, "Learning in-context learning for named entity recognition," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2023, pp. 13 661–13 675. DOI: 10.18653/v1/2023.acl-long.764 [Online]. Available: https://aclanthology.org/2023.acl-long.764/

[48] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. Daumé III, and K. Crawford, "Datasheets for datasets," in *Proceedings of the 5th Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML)*, 2018. [Online]. Available: https://arxiv.org/abs/1803.09010

[49] BigCode, *Bigcode-pii-dataset*, 2023. [Online]. Available: https://huggingface.co/datasets/bigcode/bigcode-pii-dataset

[50] ZihanWangKi. "Conllpp: Corrected conll-2003 named entity recognition dataset (dataset card)." Corrected version of CoNLL-2003 NER test labels; dataset card cites Cross-

Weigh as the source of corrections., Accessed: Feb. 22, 2026. [Online]. Available: https://huggingface.co/datasets/ZihanWangKi/conllpp

[51] nbroad. "Pii-dd-mistral-generated (kaggle dataset)." Synthetic dataset containing PII; described as generated using mistralai/Mixtral-8x7B-Instruct-v0.1 with over 2300 samples., Accessed: Feb. 22, 2026. [Online]. Available: https://www.kaggle.com/datasets/nbroad/pii-dd-mistral-generated

[52] M. Savkin, T. Ionov, and V. Konovalov, "Spy: Enhancing privacy with synthetic pii detection dataset," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, 2025, pp. 236–246. DOI: 10.18653/v1/2025.naacl-srw.23 [Online]. Available: https://aclanthology.org/2025.naacl-srw.23/

[53] H. P. Luhn, "Computer for verifying numbers," *IBM Journal of Research and Development*, vol. 4, no. 4, pp. 282–287, 1960.

[54] Fake Name Generator, *Terms of service*, 2006. [Online]. Available: https://www.fakenamegenerator.com/terms-of-service.php

[55] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020, pp. 8342–8360. DOI: 10.18653/v1/2020.acl-main.740 [Online]. Available: https://aclanthology.org/2020.acl-main.740/

[56] Z. Wang, J. Shang, L. Liu, L. Lu, J. Liu, and J. Han, "Crossweigh: Training named entity tagger from imperfect annotations," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Com-

putational Linguistics, 2019, pp. 5154–5163. DOI: 10.18653/v1/D19-1519 [Online]. Available: https://aclanthology.org/D19-1519/

[57] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, and H. Wu, "Unified structure generation for universal information extraction," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2022, pp. 5755–5772. DOI: 10.18653/v1/2022.acl-long.395 [Online]. Available: https://aclanthology.org/2022.acl-long.395/

[58] J. Zheng, Z. Liang, H. Chen, and Q. Ma, "Distilling causal effect from miscellaneous other-class for continual named entity recognition," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 3602–3615. DOI: 10.18653/v1/2022.emnlp-main.236 [Online]. Available: https://aclanthology.org/2022.emnlp-main.236/

[59] Y. Xia, Q. Wang, Y. Lyu, Y. Zhu, W. Wu, S. Li, and D. Dai, "Learn and review: Enhancing continual named entity recognition via reviewing synthetic samples," in *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2291–2300. DOI: 10.18653/v1/2022.findings-acl.179 [Online]. Available: https://aclanthology.org/2022.findings-acl.179/

[60] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3?" In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, Dublin, Ireland and Online: Association for Computational Linguistics, May 2022, pp. 100–114. DOI: 10.18653/v1/2022.deelio-1.10 [Online]. Available: https://aclanthology.org/2022.deelio-1.10/

[61] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?" In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11 048–11 064. DOI: 10.18653/v1/2022.emnlp-main.759 [Online]. Available: https://aclanthology.org/2022.emnlp-main.759/

[62] J. Ács, Á. Kádár, and A. Kornai, "Subword pooling makes a difference," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Association for Computational Linguistics, 2021, pp. 2284–2295. DOI: 10.18653/v1/2021.eacl-main.194 [Online]. Available: https://aclanthology.org/2021.eacl-main.194/

[63] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009. DOI: 10.1109/TKDE.2008.239

[64] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27 (NeurIPS 2014)*, 2014. [Online]. Available: https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks

[65] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1409.0473

[66] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, 2019. arXiv: 1907.11692 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1907.11692

[67] B. Warner et al., "Modernbert: A modern bidirectional encoder for fast, memory efficient, state-of-the-art performance," *arXiv:2412.13663*, 2024, ACL 2025. [Online]. Available: https://aclanthology.org/2025.acl-long.127/

[68] H. W. Chung, L. Hou, S. Longpre, et al., "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022. DOI: 10.48550/arXiv.2210.11416 [Online]. Available: https://arxiv.org/abs/2210.11416

Appendices

A Experimental Environment and Configuration

This appendix records the hardware, software, and execution settings used to run the ASC-PIE experiments. These details are provided to support reproducibility and to contextualize the runtime, latency, memory, and training-behavior observations reported in Chapter 5. Unless otherwise stated, the supervised fine-tuning, in-context prompting, and staged continual-learning experiments described in the thesis were executed under this environment.

Hardware and software environment. The ASC-PIE experiments were conducted on a Windows-based workstation equipped with an NVIDIA RTX A6000 GPU and 128 GB of system memory. Model development and execution were implemented in Python, with PyTorch and Hugging Face Transformers providing the core training and inference stack. Hugging Face Datasets was used for loading and storing the aligned ASC-PIE representations, while the PEFT and `bitsandbytes` libraries supported parameter-efficient and low-bit training for decoder-only models. Table A.1 summarizes the main hardware and software components referenced throughout the thesis.

Table A.1: Hardware and software environment used for ASC-PIE experiments.

Component	**Value**
Operating system	Microsoft Windows 11 Home, Version 10.0.26200 (Build 26200)
Workstation	Dell Precision 3660 (x64-based PC)
CPU	13th Gen Intel(R) Core(TM) i7-13700
System memory	128 GB (130,761 MB reported by system)
GPU	NVIDIA RTX A6000 (49,140 MiB VRAM)
GPU driver	NVIDIA driver 573.44 (WDDM)
CUDA (driver)	CUDA 12.8 (reported by `nvidia-smi`)
Python	Python 3.13.2 (64-bit)
PyTorch	2.10.0+cu126 (CUDA runtime 12.6)
Transformers	5.2.0
Datasets	4.5.0
Evaluate	0.4.6
PEFT	0.18.1
BitsAndBytes	0.49.2
cuDNN	91002

Execution setup. The experiments were implemented primarily through Jupyter notebooks. Standard scientific Python tooling, including `numpy` and `pandas`, was used for data manipulation, analysis, and reporting. GPU execution was enabled through CUDA support in PyTorch with cuDNN acceleration where available. Encoder and encoder–decoder models were trained directly on the RTX A6000, while decoder-only models used parameter-efficient fine-tuning with PEFT and low-bit loading through `bitsandbytes`. For decoder-only inference and evaluation, generation was executed in batched mode to obtain stable throughput and latency measurements. These execution choices are relevant because the thesis reports not only extraction quality, but also efficiency- and validity-related comparisons across model families.

Training and inference schedules. Within each model family, the main schedule was held fixed across compared backbones so that observed differences could be attributed primarily to model behavior rather than to divergent optimization settings. For prompting-only experiments, where model parameters are not updated, the relevant controls are the prompt budget, generation budget, deterministic decoding configuration, and shot schedule. The principal training and inference settings used across the thesis are summarized in Table A.3, while the accompanying paragraphs record the most important family-specific details.

Encoder token classification. All encoder-only models were fine-tuned under the same schedule. Training used a five-epoch budget with maximum sequence length 128, per-device batch size 16, and gradient accumulation 2, yielding an effective batch size of 32. Optimization used learning rate 2×10^{-5}, cosine scheduling, warmup ratio 0.1, and weight decay 0.01. Mixed precision used `bf16`, TF32 was enabled where supported, and checkpoint selection was based on validation F1 with early stopping. Fine-tuning also used an inverse-frequency class-weighting heuristic with additive smoothing and normalization:

$$w_c = \frac{N}{n_c + 1} \cdot \frac{1}{\max_j \left(\frac{N}{n_j + 1} \right)}, \tag{21}$$

where n_c is the count of label c in the training split and $N = \sum_c n_c$. The resulting weighted token-level cross-entropy was applied only over supervised token positions, with `-100` used as the

ignore index for masked subwords and padded tokens.

Encoder–decoder structured generation. FLAN-T5-base and BART-base were trained under the same seq2seq schedule. Training used three epochs, learning rate 2×10^{-5}, weight decay 0.01, and per-device batch size 8 for both training and evaluation. Inputs and targets were truncated to 512 tokens. Evaluation was performed at the epoch level, and generation was enabled during validation and test.

Decoder-only QLoRA fine-tuning. Decoder-only adapter training used LoRA adapters under a QLoRA setting. In the implemented configuration, the base checkpoints were loaded in 4-bit quantized form while only the low-rank adapter parameters were optimized. For both `Qwen2.5-7B-Instruct` and `Meta-Llama-3.1-8B-Instruct`, the adapter configuration used LoRA rank $r = 16$, scaling $\alpha = 32$, dropout 0.05, no bias adaptation, and the target modules `q_proj`, `k_proj`, `v_proj`, `o_proj`, `up_proj`, `down_proj`, and `gate_proj`.

In-context learning schedules. ICL experiments did not update model weights and were therefore defined by prompt budgets, decoding controls, and the shot sweep. For both decoder-only models, the experiments evaluated $k \in \{0, 1, 3, 5\}$ shots on a fixed test subset of 500 examples under seed 42. Prompts were budgeted to 4096 input tokens and generation was capped at 1024 new tokens. Decoding was deterministic, with `do_sample=False` and `num_beams=1`.

Table A.2: ICL prompting protocols and shared evaluation controls for decoder-only in-context experiments.

Protocol	**Required output format**	**Primary validity requirement**	**Shared controls**
JSON ICL	Single JSON object of the form `{"entities":[{"type", "text"}, ...]}`	Valid JSON with an `entities` list; each entry must contain an allowed ASC-PIE type and non-empty text	Models: Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct; shots $k \in \{0, 1, 3, 5\}$; fixed subset $n = 500$; deterministic decoding
KV ICL	Semicolon-delimited list of the form `TYPE: value; TYPE: value; ...`	Each segment must contain a valid ASC-PIE type and a non-empty value; null case: `No PII found.`	Models: Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct; shots $k \in \{0, 1, 3, 5\}$; fixed subset $n = 500$; deterministic decoding

Staged continual-learning schedule. The staged catastrophic-forgetting study was implemented on `roberta-large` using the shared encoder tokenization and weighted token-classification pipeline. Each stage was trained for three epochs with learning rate 1×10^{-5}, per-device training batch size 16, per-device evaluation batch size 8, gradient accumulation 2, evaluation accumulation 4, weight decay 0.01, and 500 warmup steps. Maximum sequence length was 128, and checkpoint selection was again based on validation F1. This shared schedule was used for the baseline, replay, distillation, and SPRINT-PP comparisons, with strategy-specific controls added on top of the common setup. Replay used a fixed rehearsal buffer and oversampling of prior-stage examples. Distillation used a frozen previous-stage teacher with temperature $T = 1.0$ and KD mixing weight $\lambda_{\text{KD}} = 0.5$. SPRINT-PP used a frozen previous-stage teacher together with prototype centroids and stored classifier-head rows, triage threshold $\tau = 0.5$, minimum prototype support of 8 observations, prototype EMA factor $\alpha = 0.99$, base weights $\lambda_{\text{PP}} = 1.0$ and $\lambda_{\text{HRA}} = 0.5$, and dynamic scaling of the retention terms across later stages. During SPRINT-PP updates after Stage 0, the token embeddings and the first six encoder layers were frozen to reduce optimization drift and training cost.

Table A.3: Hyperparameters and training or inference schedules used across ASC-PIE experiments. "Batch" reports per-device batch size and gradient accumulation, with effective batch shown where applicable.

Block	**Training horizon**	**Batch**	**Key optimization**	**Sequence / decoding controls**
Encoder token labeling	5 epochs (48,765 steps)	16 × accum 2 ($B_{\text{eff}} = 32$)	LR 2×10^{-5}; cosine schedule; warmup ratio 0.1; weight decay 0.01; `bf16`; early stopping patience 2	Max length 128; epoch-level eval/save; best checkpoint by validation F1; length-grouped batching
Encoder–decoder extraction	3 epochs ($\approx$ 14.4k steps)	8 (train/eval)	LR 2×10^{-5}; weight decay 0.01; epoch-level eval/save	Max input 512; max target 512; `predict_with_generate=True`; checkpoint limit 2
Decoder-only adapters (QLoRA)	1 epoch (2000 steps)	8 × accum 2 ($B_{\text{eff}} = 16$)	LR 2×10^{-4}; LoRA $r = 16$, $\alpha = 32$, dropout 0.05; eval/save every 200 steps	Max input 512; max new tokens 256; strict JSON output
ICL prompting (JSON and KV)	No weight updates	Inference batch 8 (where supported)	Seed 42; shot sweep $k \in \{0, 1, 3, 5\}$; deterministic decoding	Max input 4096; max new tokens 1024; `do_sample=False`; `num_beams=1`; test subset $n = 500$
Staged catastrophic forgetting	3 epochs per stage	Train 16 × accum 2; eval 8; eval accum 4	LR 1×10^{-5}; weight decay 0.01; warmup 500 steps; best checkpoint by validation F1	Max length 128; epoch-level eval/save; shared base schedule across all four strategies; Replay: fixed buffer + oversampling; Distillation: $T = 1.0$, $\lambda_{\text{KD}} = 0.5$; SPRINT-PP: $\tau = 0.5$, $\lambda_{\text{PP}} = 1.0$, $\lambda_{\text{HRA}} = 0.5$, proto min = 8, EMA = 0.99, dynamic retention scaling, embeddings + first 6 layers frozen after Stage 0

B ASC-PIE Reference Specifications

This appendix records the main reference specifications used throughout ASC-PIE. It brings together the unified label schema, the encoder tag inventory derived from that schema, and the final prompt protocols used in the in-context learning experiments. Collecting these materials in a single appendix makes the reference interface of ASC-PIE easier to consult, since all of them describe formal structures that remain stable across the experiments reported in the thesis.

Unified schema overview. ASC-PIE uses a unified privacy-oriented schema containing 19 canonical entity types. These types are grouped into six broader semantic categories: personal entities, temporal entities, contact information, professional and work entities, government and financial identifiers, and geographic entities. The same canonical type inventory is used across corpus construction, prompt design, model supervision, and evaluation. This consistency is important because the framework compares multiple model families under a shared entity inventory rather than under family-specific label sets. Figure B.1 provides a visual summary of the unified ASC-PIE schema.

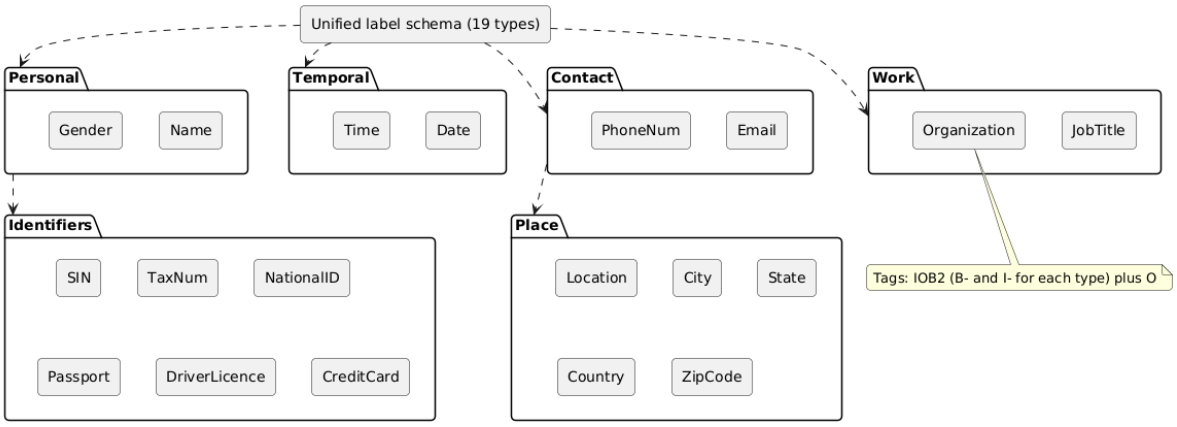

Figure B.1: Unified ASC-PIE label schema with 19 canonical entity types. Encoder-based models use an IOB2-derived tag set with `O`, `B-` tags for all types, and `I-` tags for 16 types.

Encoder tag inventory. For encoder-based models, the ASC-PIE schema is represented through an IOB2-derived tag inventory. This inventory includes the outside tag `O`, `B-` tags for all 19 types, and `I-` tags for 16 types. The types `GENDER`, `NATIONALID`, and `PASSPORT` do not include `I-` tags in the exported encoder label map because they are predominantly realized as single-token

entities in the constructed corpus. The resulting encoder inventory therefore contains 36 labels in total. Table B.1 lists the encoder tag inventory by semantic category and entity type.

Table B.1: ASC-PIE encoder tag inventory by entity type.

Semantic category	**Type**	**B-tag**	**I-tag**
Personal entities	NAME	B-NAME	I-NAME
	GENDER	B-GENDER	—
Temporal entities	DATE	B-DATE	I-DATE
	TIME	B-TIME	I-TIME
Contact information	EMAIL	B-EMAIL	I-EMAIL
	PHONENUM	B-PHONENUM	I-PHONENUM
Professional and work	JOBTITLE	B-JOBTITLE	I-JOBTITLE
	ORGANIZATION	B-ORGANIZATION	I-ORGANIZATION
Government and financial identifiers	SIN	B-SIN	I-SIN
	TAXNUM	B-TAXNUM	I-TAXNUM
	NATIONALID	B-NATIONALID	—
	PASSPORT	B-PASSPORT	—
	DRIVER-LICENCE	B-DRIVER-LICENCE	I-DRIVER-LICENCE
	CREDITCARD	B-CREDITCARD	I-CREDITCARD
Geographic entities	LOCATION	B-LOCATION	I-LOCATION
	CITY	B-CITY	I-CITY
	STATE	B-STATE	I-STATE
	COUNTRY	B-COUNTRY	I-COUNTRY
	ZIPCODE	B-ZIPCODE	I-ZIPCODE

Prompt protocol overview. ASC-PIE uses two main prompt protocols in the in-context learning experiments: a JSON protocol and a semicolon-delimited key-value (KV) protocol. Both protocols rely on the same 19-type ASC-PIE inventory, but they differ in how the expected output is expressed. Across all ICL runs, demonstrations are selected once with a fixed random seed and then reused consistently for the corresponding shot setting. The exact demonstration indices are preserved in the exported artifact files referenced in Chapter 4. Decoder-only ICL uses deterministic decoding with fixed prompt and generation budgets: the prompt budget is 4096 tokens, the maximum generation budget is 1024 new tokens, sampling is disabled, and generation uses a single beam.

JSON protocol. The JSON protocol requires the model to return a single JSON object with an `entities` field. Each extracted entity is represented by a `type` field and a `text` field. If no

PII is present, the expected output is {"entities": []}. The generic JSON prompt template used in the experiments is shown below.

```
System:
You are a PII extraction system.
Extract only the following ASC-PIE entity types:
NAME, GENDER, DATE, TIME, EMAIL, PHONENUM, SIN, TAXNUM,
NATIONALID, PASSPORT, DRIVER-LICENCE, CREDITCARD,
LOCATION, CITY, STATE, COUNTRY, ZIPCODE, JOBTITLE, ORGANIZATION.

Return JSON only in the following format:
{"entities": [{"type": "TYPE", "text": "MENTION"}, ...]}

Rules:
- Use only the allowed types listed above.
- Do not add explanations or extra text.
- If no PII is present, return:
{"entities": []}

User:
<INPUT_TEXT>
```

For shot-based prompting, demonstrations are inserted between the instruction block and the final query using the same JSON output structure. A compact demonstration example is shown below.

```
Example demonstration:
Input:
Sarah Johnson emailed sarah.j@example.com at 10:30 am.

Output:
{"entities": [
        {"type": "NAME", "text": "Sarah Johnson"},
        {"type": "EMAIL", "text": "sarah.j@example.com"},
        {"type": "TIME", "text": "10:30 am"}
        ]}
```

KV protocol. The KV protocol mirrors the encoder-decoder extraction format and requires the model to return a semicolon-delimited sequence of key-value pairs. If no PII is present, the expected output is the sentinel string `No PII found.` The generic KV prompt template used in

the experiments is shown below.

```
System:
You are a PII extraction system.
Extract only the following ASC-PIE entity types:
NAME, GENDER, DATE, TIME, EMAIL, PHONENUM, SIN, TAXNUM,
NATIONALID, PASSPORT, DRIVER-LICENCE, CREDITCARD,
LOCATION, CITY, STATE, COUNTRY, ZIPCODE, JOBTITLE, ORGANIZATION.

Return the output only in the following format:
TYPE: value; TYPE: value; TYPE: value

Rules:
- Use only the allowed types listed above.
- Separate multiple entities with semicolons.
- Do not add explanations or extra text.
- If no PII is present, return exactly:
No PII found.

User:
<INPUT_TEXT>
```

For shot-based prompting, demonstrations are inserted before the final query using the same KV structure. A compact demonstration example is shown below.

```
Example demonstration:
Input:
Sarah Johnson emailed sarah.j@example.com at 10:30 am.

Output:
NAME: Sarah Johnson; EMAIL: sarah.j@example.com; TIME: 10:30 am
```

C SPRINT-PP Supplementary Details

This appendix provides a compact conceptual supplement for the proposed SPRINT-PP method. Rather than repeating the experimental analysis from Chapter 5, it gathers the main design elements that help interpret the method itself: the acronym expansion, the staged token categories that motivate the correction mechanism, the relation of SPRINT-PP to prior mitigation strategies, and its specific connection to the MiB framework. Implementation schedules and training configurations are reported separately in Appendix A.

Acronym expansion and design intuition. The acronym **SPRINT-PP** stands for **Selective Prototype-Routed Incremental NER Triage with Privacy Preservation**. Each part of the name reflects a specific aspect of the final method. *Selective* indicates that correction is not applied uniformly, but only to suspected old-entity tokens that are currently labeled as `O`. *Prototype-Routed* indicates that the method retains compact prototype summaries of previously learned labels and uses them as part of the preservation mechanism. *Incremental NER* reflects the intended setting of staged class-incremental named-entity recognition under evolving label sets. *Triage* refers to the routing step that identifies suspected mislabeled old tokens before corrective supervision is applied. Finally, *Privacy Preservation* indicates that old knowledge is stabilized without storing raw historical examples. Table C.1 summarizes these components.

Table C.1: Meaning of the SPRINT-PP acronym.

Component	**Meaning**
Selective	Corrective supervision is applied only to suspected old-entity tokens that are currently labeled as `O`.
Prototype-Routed	Compact prototype centroids are retained and used as part of the preservation mechanism for earlier labels.
Incremental NER	The method is designed for staged class-incremental named-entity recognition under evolving label sets.
Triage	Suspected mislabeled old tokens are identified before selective correction is applied.
Privacy Preservation	Old-label knowledge is stabilized without storing raw historical examples.

Token categories under staged relabeling. SPRINT-PP is motivated by the fact that later-stage supervision mixes genuinely new entities with previously learned entities that remain present in the text but are now relabeled as background. In the staged setting used in this thesis, later-stage tokens can be interpreted through three categories. Type A tokens belong to current-stage entities and are correctly labeled by the current supervision. Type B tokens belong to previously learned entity types but are now labeled `O`; these are the tokens responsible for the poisoned-gradient problem targeted by SPRINT-PP. Type C tokens are true non-entity tokens and should remain labeled `O`. Table C.2 summarizes these interpretive categories.

Table C.2: Interpretive token categories under staged relabeling.

Category	Interpretation
Type A	Token belongs to a current-stage entity and is correctly labeled by the current-stage supervision.
Type B	Token belongs to an old-stage entity that remains in the text but is relabeled as `O` in the current-stage supervision. These tokens create the poisoned-gradient problem targeted by SPRINT-PP.
Type C	Token is a true non-entity token and should remain labeled as `O`.

The role of SPRINT-PP is to distinguish Type B tokens from genuine Type C background tokens without replaying raw historical examples. In the implemented method, this distinction is made primarily through frozen-teacher confidence over previously learned labels, after which routed tokens are corrected selectively and preserved through prototype anchoring and classifier-head stabilization.

Relation to prior mitigation ideas. SPRINT-PP shares some intuition with earlier mitigation strategies, but differs in the mechanisms used to preserve old knowledge under privacy-sensitive continual NER. Replay retains raw historical examples and therefore provides direct old-class supervision, but it does not explicitly isolate mislabeled old tokens as a separate problem. Distillation uses a frozen teacher and avoids raw replay, but it applies a broader preservation signal that does not focus specifically on suspected old tokens relabeled as `O`. SPRINT-PP combines frozen-teacher guidance with explicit token triage, prototype-based anchoring, and classifier-head stabilization, while remaining privacy-safe by design. Table C.3 summarizes these differences.

Table C.3: Compact comparison between SPRINT-PP and related mitigation ideas.

Property	Replay	Distillation	SPRINT-PP
Uses raw historical examples	Yes	No	No
Targets mislabeled old tokens explicitly	No	Indirectly	Yes
Uses frozen teacher guidance	No	Yes	Yes
Uses compact representation-level memory	No	No	Yes, via prototypes
Stabilizes previously learned classifier rows	No	No	Yes
Privacy-safe by design	No	Yes	Yes

Taken together, these components explain the intended role of SPRINT-PP in the thesis. It is designed as a privacy-safe continual-learning method that targets the mislabeled-token mechanism driving forgetting in staged NER while avoiding the storage of raw historical PII.

Relation to MiB. The closest methodological antecedent to SPRINT-PP is the MiB framework proposed by Cermelli et al. [43] for class-incremental semantic image segmentation. MiB identifies a functionally analogous challenge: pixels associated with previously learned classes are relabeled as background in newly introduced training stages. To mitigate this, MiB replaces hard background supervision with teacher-based soft correction. SPRINT-PP is inspired by this old-class-to-background intuition, but adapts it to privacy-oriented sequence labeling. In particular, SPRINT-PP operates over token sequences rather than pixels, routes only suspected old tokens rather than correcting all background positions uniformly, and augments teacher-guided correction with prototype memory and classifier-head stabilization without relying on raw replay. Table C.4 summarizes these differences.

Table C.4: Comparison of design dimensions between MiB and the proposed SPRINT-PP method.

Dimension	**MiB [43]**	**SPRINT-PP (This Work)**
Domain	Semantic image segmentation (pixel-level)	Transformer-based NER (token sequence labeling)
Granularity of correction	Pixels relabeled as background in later stages	Tokens relabeled as `O` in later stages
Old-class / background problem	Old-class pixels become background	Old entity tokens become `O` under staged relabeling
Teacher usage	Soft correction of background predictions	Selective teacher-guided correction of suspected old tokens
Memory representation	No explicit external memory beyond teacher guidance	Prototype centroids and stored classifier-head rows, without raw replay
Privacy setting	Not a privacy-oriented method	Designed explicitly for privacy-safe continual NER