ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Dipartimento di Interpretazione e Traduzione

Corso di Laurea Magistrale in

Specialized Translation – Translation and Technology

Automation of Corporate Language Services: Hybrid AI Approaches to Text Anonymization and Termbase Expansion

Tesi di Laurea Magistrale in Terminology

Relatore
Prof. Adriano Ferraresi
Correlatore
Prof. Alberto Barrón-Cedeño
Correlatrice
Sara Grizzo

Presentata da
Angela Forzatti

III – Marzo 2026
Anno Accademico 2025/2026

ABSTRACT

This thesis examines application scenarios of Artificial Intelligence (AI) and Automation to the workflows of a corporate Language Services (LS) department through two practical case studies. It focuses on hybrid AI approaches to Text Anonymization and Termbase (TB) Expansion, combining theoretical analysis with applied system design and evaluation. The first case study presents an LLM-based anonymization tool developed to ensure compliance with data protection regulations that integrates with the already existing platforms and workflows. The second case study proposes a semi-automated pipeline that performs subdomain identification, term extraction, concept consolidation, metadata augmentation and relation identification to enhance the internal TB. Both solutions combine deterministic processes, probabilistic models and human expertise, demonstrating how these components mitigate one another's limitations. Results show that AI-assisted workflows can reduce manual effort, improve scalability and support consistency, while expert human validation remains essential to ensure quality and compliance. The thesis situates these findings within the broader technological, professional and ethical transformation of the language industry, concluding that hybrid AI paradigms could be a way to enable the effective and responsible deployment of AI in corporate environments.

Keywords: Artificial Intelligence; Automation; Large Language Models; Hybrid AI; Text Anonymization; Terminology; Termbase Expansion; Translation Technology; Natural Language Processing; Language and Translation Industry; Human-Machine Collaboration; Human-in-the-Loop; Automatic Term Extraction.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS iv

LIST OF FIGURES vi

LIST OF TABLES vi

INTRODUCTION 1

CHAPTER 1: AUTOMATION AND ARTIFICIAL INTELLIGENCE 3

1.1. Chapter overview 3

1.2. Defining Automation and Artificial Intelligence 3

1.2.1. Understanding Automation: Concepts, Types and Evolution 3

1.2.2. Artificial Intelligence: from Machine Learning to Large Language Models 6

1.3. AI and Automation in the Translation and Language Industry 12

1.3.1. Automation across Language Industry Workflows 12

1.3.2. Diversification of Services 14

1.3.3. New Professional Skills and Profiles 15

1.4. Environmental, Social and Economic Dimensions of AI and Automation 17

1.4.1. Environmental Sustainability 17

1.4.2. Social and Ethical Implications 17

1.4.3. Economic, Societal and Cultural Implications 18

CHAPTER 2: TEXT ANONYMIZATION 21

2.1. Chapter Overview 21

2.2. Introduction 21

2.2.1. Context and Motivation 21

2.2.2. Core Concepts of Text Anonymization 22

2.2.3. Background on NLP Approaches to Anonymization 23

2.3. Preliminary Exploration of NLP Models 24

2.4. Methodology: Implementation of LLM-based approach 29

2.4.1. Application Overview and Workflow 29
2.4.2. Graphical User Interface 31
2.4.3. Security and Configuration 31
2.4.4. Parsing 32
2.4.5. PII detection via LLM 32
2.4.6. Anonymization 33
2.5. Results 35
2.6. Discussion 37
2.7. Additional Implementations and Use Cases 39
CHAPTER 3: TERMBASE EXPANSION 41
3.1. Chapter Overview 41
3.2. Introduction 41
3.2.1. Context and Motivation 41
3.2.2. Core Concepts of Terminology and Termbase Expansion 42
3.3. Methodology: Implementation of LLM-based pipeline 45
3.3.1. Data Acquisition 46
3.3.2. Phase 1: Subdomain identification 47
3.3.3. Phase 2: Extraction 48
3.3.4. Phase 3: Concept consolidation 50
3.3.5. Phase 4: Augmentation 52
3.3.6. Phase 5: Clustering and relation identification 54
3.4. Results 55
3.4.1. Quantitative overview 55
3.4.2. Qualitative analysis 56
3.5. Discussion 58
CONCLUSIONS 61
BIBLIOGRAPHY 63
APPENDIX 73

A. CHAPTER 2: TEXT ANONYMIZATION 73
B. CHAPTER 3: TERMBASE EXPANSION 77

LIST OF ABBREVIATIONS

Abbreviation	Full form
AI	Artificial Intelligence
ATE	Automatic Term Extraction
BiLSTM-CRF	Bidirectional Long Short-Term Memory with Conditional Random Fields
CAT	Computer-Assisted Translation
CMS	Content Management System
CRF	Conditional Random Fields
DL	Deep Learning
EITL	Expert-in-the-Loop
GenAI	Generative Artificial Intelligence
GUI	Graphical User Interface
HITL	Human-in-the-Loop
LLM	Large Language Model
LS	Language Services[1]
LSP	Language Service Provider
LSTM	Long Short-Term Memory
ML	Machine Learning
MT	Machine Translation
MTPE	Machine Translation Post-Editing
NER	Named Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
NMT	Neural Machine Translation
PII	Personally Identifiable Information
PLM	Pretrained Language Model
POS	Part-of-Speech
RAG	Retrieval-Augmented Generation
RegEx	Regular Expressions
RNN	Recurrent Neural Network
SDLXLIFF	SDL XML Localization Interchange File Format[2]
TB	Termbase

[1] Department at Munich Re.
[2] Bilingual file format used in Trados.

TM	Translation Memory
TMS	Translation / Terminology Management System
TMX	Translation Memory eXchange

LIST OF FIGURES

Figure 1.1. Example visualization of word embeddings in a vectorial space. Created with Word Embedding Demo (Xu et al., 2024). 9
Figure 2.2. CRF model pipeline. 27
Figure 2.3. BiLSTM-CRF model pipeline. 27
Figure 2.4. Overview of LLM-driven pipeline. 30
Figure 2.5. PII Anonymizer interface. 31
Figure 3.6. Core phases of Termbase Expansion pipeline. 46

LIST OF TABLES

Table 1.1. Summary of new services and tasks in the Language Industry. 15
Table 2.2. Gretel.ai dataset languages and number of documents. 25
Table 2.3. Gretel.ai PII types and occurrences in the dataset. 25
Table 2.4. Span vs BIO annotation example. 26
Table 2.5. CRF vs BiLSTM-CRF F1 scores. 28
Table 2.6. Tag definition structure. 34
Table 2.7. BiLSTM-CRF vs LLM PII detection 36
Table 2.8. LLM vs Hybrid anonymization F1 scores. 37
Table 3.9. Statistics of corpus text types. 47
Table 3.10. Metadata fields and values. 52
Table 3.11. Example of conceptual relation. 55
Table 3.12. Extraction pipeline in numbers. 56
Table 3.13. Candidate and validated relations by type. 56
Table 2.14. CRF per entity scores. 73
Table 2.15. BiLSTM-CRF per entity scores. 74
Table 2.16. Overview of libraries and packages. 75
Table 2.17. Full PII detection system message. 75
Table 2.18. Full PII detection prompt. 75
Table 3.19. Term classification prompt. 77
Table 3.20. Extraction prompt. 78
Table 3.21. Concept consolidation prompt 80
Table 3.22. Relation prompt. 80

INTRODUCTION

This thesis investigates how emerging technologies, in particular AI-assisted systems, can be concretely integrated into professional translation workflows by analyzing two projects developed during the internship at Munich Re's Language Services (LS), a global reinsurance company based in Germany[3]. The first project consists in the development of an anonymization tool to anonymize personal data in translation documents to ensure compliance with external and internal data protection requirements. The second project involves automating certain aspects of the terminology workflow to support the expansion and enrichment of the internal Termbase (TB), specifically via term extraction, metadata augmentation and relation identification.

These specific language-related case studies situate themselves in the middle of a broader discourse concerning the technological developments currently reshaping the language industry. From earlier innovations such as Computer-Assisted Translation (CAT) tools and Machine Translation (MT) (Faes & Massey, 2024), to more recent advances in Artificial Intelligence (AI) and Natural Language Processing (NLP), automation has profoundly impacted the very notion of translation, as well as workflows and professional roles (Briva-Iglesias & O'Brien, 2022; Declercq & Egdom, 2023).

Munich Re provides a particularly relevant environment to observe these dynamics. As one of the world's leading reinsurance companies, it operates across numerous markets and specialized domains, dealing with a diverse range of multilingual documentation: technical and financial reports, internal guidelines and press releases, contracts and other legal materials, IT documentation, and even personal documents. Within this setting, LS functions as an internal Language Service Provider (LSP) responsible for translation, revision, terminology management and language consultancy. Its work supports a wide range of business units and ensures that multilingual communication meets both regulatory and corporate standards. To manage this complexity, LS combines human expertise, essential for high-risk and confidential materials, with several technological tools, including CAT tools, MT systems and AI-assisted solutions. The aim of these projects was precisely to leverage these tools to support existing workflows and improve efficiency, consistency and data security.

Following this brief introduction, Chapter 1 will provide a theoretical and technical overview of the topics of automation and AI, including their applications in the industry and the socio-ethical and sustainability issues they raise; Chapters 2 and 3 will then present the two case studies developed during the internship: the tasks of LLM-assisted anonymization and TB expansion will first be theoretically framed and then illustrated through their concrete implementations.

[3] https://www.munichre.com/en.html

This thesis makes use of AI tools (in particular, Microsoft Copilot and ChatGPT) for the development of the code used in the two case studies, as well as the generation of preliminary sample and testing materials. All AI-generated content was reviewed and adapted to ensure that it operated correctly, achieved the intended outcomes and met quality and internal regulatory standards.

CHAPTER 1: AUTOMATION AND ARTIFICIAL INTELLIGENCE

1.1. Chapter overview

This chapter introduces the conceptual foundations of automation and Artificial Intelligence (AI) and situates them within the broader technological developments that are reshaping contemporary work practices. It outlines how automation has evolved from rule-based deterministic systems to data-driven probabilistic approaches enabled by Machine Learning (ML) and, more recently, by Large Language Models (LLMs). The chapter also connects these developments to the Translation and Language Industry, where automation has transformed workflows, roles and service offerings. These theoretical frameworks provide the basis for understanding the dynamics that underline the subsequent case studies of text anonymization and Termbase (TB) expansion.

The chapter is structured into three main sections: Section 1.2 defines automation and AI, tracing their evolution and examining key concepts such as cognitive automation, ML, Deep Learning (DL), and LLMs; Section 1.3 explores how these technologies manifest within the Translation and Language Industry, focusing on workflow automation, the diversification of language-related services and emerging roles and skills; finally, Section 1.4 expands the perspective to consider the wider environmental, ethical, economic and cultural implications of AI-driven automation.

1.2. Defining Automation and Artificial Intelligence

1.2.1. Understanding Automation: Concepts, Types and Evolution

1.2.1.1. Fundamental Types and the Shift in Perspectives

Early studies framed automation within a substitution paradigm, whereby "a zero-sum relationship between humans capabilities and automated systems [was assumed], suggesting that technological advancement necessarily reduces demand for human labor" (Alla, 2025, p. 1021) especially in routine tasks that follow "precise, well-understood procedures" and can be effectively codified for deterministic and rule-based systems (Autor, 2015, p. 11). To understand how automation has evolved beyond this substitution logic, it is necessary to clarify what automation entails in technical terms and which factors shape its design and development.

Parasuraman et al. (2000, p. 287) offer a definition of automation as a "device or system that accomplishes (partially or fully) a function that was previously, or conceivably could be, carried out (partially or fully) by a human operator". Automation can be applied to physical tasks and processes (mechanical automation), non-physical business tasks and processes (information and control automation) or a combination of both (Engel et al., 2022, page 341). Information and control automation can be further subdivided into four types, each corresponding to a stage of human information processing (Parasuraman et al., 2000, pp. 288–289): (i) information acquisition, (ii)

information analysis, (iii) decision and action selection, and (iv) action implementation. A single system can involve the automation of one or more dimensions across a continuum, from low (fully manual) to high (fully automatic), according to the task it is built to carry out and the needs and requirements of the human operators (Parasuraman et al., 2000).

Automation design, in particular what stages to automate and at which level, should account for criteria other than technical feasibility and economic motivation (Parasuraman et al., 2000). The primary criteria is the impact of automation on the human operator's performance: since the goal is to enhance rather than degrade human performance, design choices should (i) ease the mental workload of the human operator by being easy to use and helpful rather than increasing it with clumsy and complex implementations, (ii) not deprive the human operator of the situational awareness related to the system's operational status and work environment, (iii) not lead to complacency, which happens when a system is highly but not perfectly reliable but the operator develops an over-trust effect and fails to catch the system's errors or failures and (iv) not cause skill degradation in the human operator, who, especially in case of system failures, should possess the knowledge to compensate for it.

Other two secondary criteria have been identified by Parasuraman et al. (2000): on one hand, the reliability of the automation system should not be overstated simply because the system performs well in statistical testing, because operational scenarios could generate unexpected errors; on the other hand, it is important to account for the potential severity of a system failure, which means that a low-risk task could possibly be fully automated while it would be advisable to be cautious about fully automating a high-risk task.

Contemporary perspectives move away from the notion that automation is merely a replacement and propose it as a transformation of human activity: modern frameworks increasingly adopt an augmentation paradigm, whereby "well-designed automation can enhance human capabilities by handling routine cognitive tasks, thereby freeing workers to engage in higher-level analytical, creative, and interpersonal activities" (Alla, 2025, p. 1021; Autor, 2015; Moorkens, Way, et al., 2024). These tasks involve skills that do not have "explicit 'rules' or procedures" and thus cannot be easily broken down in codifiable steps (Autor, 2015). This phenomenon is defined in Autor (2015) as Polanyi's Paradox: there are skills and knowledge that are only tacitly understood by humans because they are "capabilities that the human species evolved, rather than developed" and are therefore difficult to formalize into explicit rules for computers (Autor, 2015, p. 12). These automation-resistant tasks can be grouped into: (i) abstract tasks, involving skills such as reasoning, communication abilities, intuition, judgment and expert mastery, and (ii) manual tasks involving skills such as sensorimotor abilities, physical flexibility, situational adaptability, visual and language recognition (Autor, 2015).

However, strategies have been researched and implemented to overcome Polanyi's paradox. Historically, engineers have used a strategy called environmental control, which is based on the idea that since machines are not able to adapt to the unpredictability of real-world scenarios, creating a simple and predictable environment allows them to operate reliably despite their lack of flexibility (Autor, 2015, p. 23). More recently, a different approach has emerged: instead of adapting the environment to the machine, ML has been used to adapt the machine to the environment. ML is a branch of AI concerned with designing systems capable of improving automatically through training: instead of relying on explicit rules, ML algorithms infer patterns through the observation of data, construct models based on them and use them to make predictions about new and unseen data (Engel et al., 2022; Russell & Norvig, 2022). This enables them to generalize beyond the data they have been trained on and therefore handle variability to a certain extent (Autor, 2015).

1.2.1.2. The New Frontier of Cognitive Automation

ML has been successfully applied to a subset of tasks involved in "knowledge and service work", i.e. cognitive tasks (Engel et al., 2022, p. 339). Cognition, in this context, is defined as "the process of developing knowledge and understanding" (Engel et al., 2022, p. 341). Scholars distinguish between human cognition and artificial cognition. Human cognition excels in what Kahneman (2015, in Engel et al. 2022, p. 341) refers to as system 1 thinking: "affective, fast, emotional, ad-hoc, stereotypic, subconscious thinking". By contrast, artificial cognition is designed to approximate "system 2 thinking, which is logical, […] conscious, and follows the probability theory paradigms". Cognitive automation builds on this distinction: it refers to the application of ML algorithms to approximate aspects of system 2 cognition, extending automation capabilities beyond deterministic, rule-based tasks into complex areas involving judgment, inference and probabilistic outcomes (Engel et al., 2022). Because cognitive automation interacts differently with tasks depending on the kind of skills and level they require, Engel et al. (2022) outline four broad implementation strategies, ranging from full automation for low creative and interpersonal tasks to augmentation as support for highly creative and interpersonal tasks.

ML-driven cognitive automation does not replace deterministic systems such Workflow Management (WfM), but orchestrates them in hybrid architectures that combine rule-based reliability with inference capabilities (Engel et al., 2022). Applied in Business Process Automation (BPA), ML-driven cognitive automation can significantly increase process efficiency (Engel et al., 2022). However, adoption in companies remains limited due to the "still comparably high price of cognitive automation tools, the required amounts of data, and the insecurity of organizations due to the unpredictability and probabilistic character of outcomes" (Lacity & Willcocks, 2018 in Engel et al., 2022, p. 346). This further highlights that the future of automation lies not in substitution, but in

hybrid intelligence, where human expertise and machine inference combined achieve better performance (Dellermann et al., 2019 in Engel et al., 2022).

1.2.2. Artificial Intelligence: from Machine Learning to Large Language Models

1.2.2.1. Artificial Intelligence: General Definition

AI is broadly defined as the capability of machines to "compute how to act effectively and safely in a wide variety of novel situations [and subfields], ranging from the general (learning, reasoning, perception, and so on) to the specific, such as playing chess, proving mathematical theorems, writing poetry, driving a car, or diagnosing diseases" (Russell & Norvig, 2022, p. 19).

Since its conception, AI research has explored two main conceptual approaches concerning how machines should operate (Russell & Norvig, 2022, p. 20):

- acting or thinking humanly: on one side, the focus was placed on simulating human thought and behavior by developing machines that act humanly, i.e. possess human capabilities (such as natural language understanding, knowledge representation, automated reasoning and even computer vision and robotics) and pass the Turing Test, or think humanly by modelling human cognitive processes;
- acting or thinking rationally: on the other side, the goal was to achieve rationality by developing systems that think rationally using logic and probability or act rationally by achieving the best outcome.

The standard modern approach to AI shifts the focus away from imitation and toward rationality: while human-like behavior is not excluded, contemporary systems prioritize "agents that make decisions under uncertainty to attain the best expected outcome" (Russell & Norvig, 2022, p. 22), also known as rational agents. As Russel and Norvig (2022, pp. 54–55) describe it, an agent is anything that perceives its environment (i.e., a percept sequence) through sensors and acts upon it through actuators according to its agent function (i.e., an abstract mathematical description) implemented by an agent program (i.e., a concrete implementation running within some physical system); this agent's performance is then measured according to what wants to be achieved. A rational agent therefore "should select an action that is expected to maximize its performance measure given the evidence provided by the percept sequence and whatever built-in knowledge the agent has" (Russell & Norvig, 2022, p. 58).

Finally, it is essential to clarify that the focus of this discussion is narrow AI, which refers to AI systems designed to perform specific domain tasks at or above the human level, and not Artificial General Intelligence (AGI), which refers to AI systems able to perform (or outperform) as humans in any intellectual domain; that is because AGI remains largely speculative rather than a technical reality

(Engel et al., 2022). Therefore, narrow AI, with its domain-specific capabilities, is the foundation upon which practical applications and business solutions are built at the moment.

1.2.2.2. Machine Learning

Early AI systems were built on Boolean logic and hand-crafted rules and knowledge (Russell & Norvig, 2022). However, two major problems arose: first, it was impossible to program in advance all possible paths and solutions; second, in complex domains a solution might not even be available or could involve skills not explicitly codifiable, as discussed in Section 1.2.1 (Autor, 2015). These limitations led researchers, from the 1980s onwards, to switch to a different approach that incorporated probability and ML (Russell & Norvig, 2022).

ML refers to a system that improves autonomously through experience (Jordan & Mitchell, 2015 in Engel et al., 2022). In practice, ML algorithms observe and analyze past examples to infer an underlying model, which is subsequently employed to make predictions on new data; these predictions are then evaluated and the feedback is used to further refine and optimize the model over time (Wilamowski & Irwin, 2018 in Engel et al., 2022; Russell & Norvig, 2022). The final goal of ML is to generalize from the inferred hypothesis to new and unseen data (Russell & Norvig, 2022).

ML systems are trained according to three main approaches (Russell & Norvig, 2022, p.671): in supervised learning, a system learns from an input-output pair by mapping the input to the correct output, also known as label; in unsupervised learning, “the agent learns patterns in the input without any explicit feedback” by, for example, clustering inputs with similar features; in reinforcement learning, the system receives positive or negative feedback and its goal is to optimize its actions to maximize the positive outcomes.

The modern evolution of ML has been made possible by the availability of large datasets and new training techniques. The explosion of so-called “big data”, i.e. huge datasets of texts, audios, images and videos, collected through the web, provided the raw material for training increasingly complex models (Hagos et al., 2024; Russell & Norvig, 2022). At the same time, transfer learning emerged as a crucial innovation: instead of training models from scratch, knowledge from one domain can be transferred to a new domain to enable the model to learn from it. For example, an LLM trained on everyday language already has the ability to understand language itself and this can be leveraged to further fine-tune it on specialized language, such as medical or financial, or for specific tasks, such as sentiment analysis or spam detection (Jurafsky & Martin, 2025; Russell & Norvig, 2022). This approach is particularly useful as it requires less data, computational effort and training time (Hagos et al., 2024).

1.2.2.3. Deep Learning for Natural Language Processing

While ML approaches improved performance in many tasks, when it came to Natural Language Processing (NLP) early systems "based on parsing and semantic analysis" still faced important limitations (Russell & Norvig, 2022, p. 907). NLP is a field concerned with understanding, processing and generating human language, in opposition to formal languages used so far to communicate with machines (Russell & Norvig, 2022). However, natural language exhibits a high degree of "complexity of […] phenomena in real text" and ML approaches did not manage to capture and generalize well its fluidity (Russell & Norvig, 2022, p. 907). To overcome these constraints, researchers turned to DL, which has become the dominant paradigm within ML and the primary driver of recent breakthroughs in NLP and related domains (Russell & Norvig, 2022).

DL refers to models organized into complex structures with multiple hidden layers and steps (Russell & Norvig, 2022). These structures are often referred to as neural networks, as they are meant to resemble the networks of neurons in the brain (Russell & Norvig, 2022). A key innovation in NLP, as explained by Russel and Norvig (2022), focused on producing effective representations of words and their relationships. Word embeddings replaced sparse word representations with dense vectors that place semantically similar words closer together in a large vectorial space, allowing for the identification of semantic relationships between words. Figure 1.1 illustrates this concept using a simplified three-dimensional projection of word vectors, where gender and age are example dimensions that highlight relationships between the words. To capture context beyond individual words, Recurrent Neural Networks (RRNs) – and later Long Short-Term Memory (LSTM) models – were designed to retain information over several processing steps; however, they still struggled to learn relationships between words separated by a large distance in a text due to gradual memory decay and bias toward more recent context (Russell & Norvig, 2022).

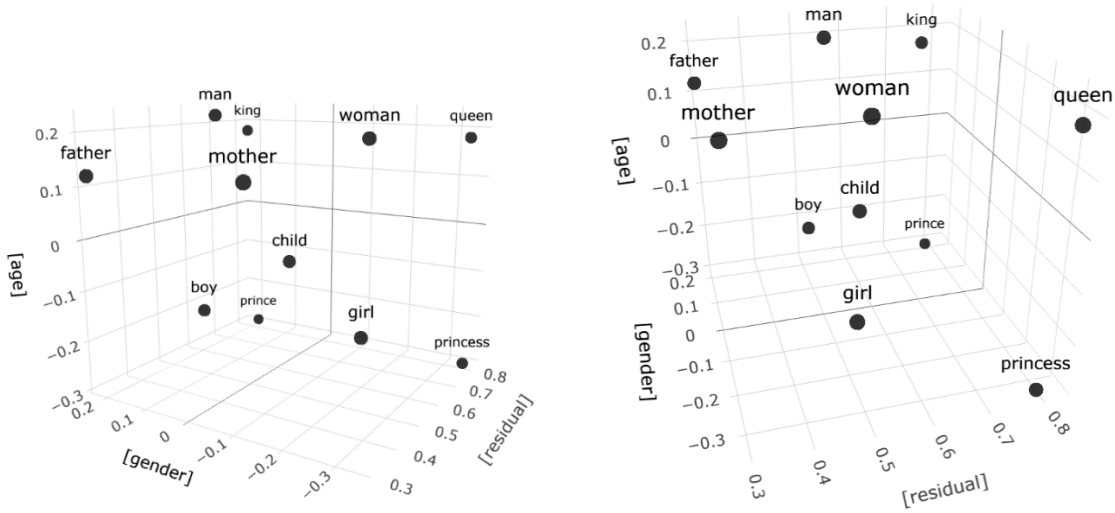

Figure 1.1. Example visualization of word embeddings in a vectorial space. Created with Word Embedding Demo (Xu et al., 2024).

These limitations were overcome thanks to the development of the Transformer architecture. In particular, the decisive breakthrough was the introduction of the self-attention mechanism, which enables parallel processing of an input text (Vaswani et al., 2023). Unlike earlier sequential models, Transformers can evaluate relationships between all tokens in a sequence simultaneously, allowing them to capture nuanced contextual information across entire documents rather than being constrained by short or decaying memory spans (Hagos et al., 2024). This innovation dramatically enhanced the quality of language representations, laying the groundwork for the development of LLMs (Jurafsky & Martin, 2025).

There are three types of Transformer architectures (Hagos et al., 2024; Jurafsky & Martin, 2025): (i) encoder-only architectures, such as the BERT-based models (Devlin et al., 2019), specialize in analyzing and classifying text by producing semantically and syntactically rich vector representations of words, which makes them particularly effective for tasks like sentiment analysis, Named Entity Recognition (NER) and information retrieval; (ii) decoder-only architectures, such as the GPT and LLaMA families (Radford & Narasimhan, 2018; Touvron et al., 2023), focus on predicting and generating new output tokens given an input sequence, making them the most suitable choice for tasks such as text generation; and (iii) encoder-decoder architectures combine both approaches: they encode an input sequence into a contextual representation and then decode it into a new sequence, but the output tokens are not required to align one-to-one with the input tokens, making it possible for the model to produce sequences of different lengths or even in different languages (for example, MT systems encode the meaning of the source sentence and then decode it into the target language).

The adoption of DL in NLP improved performance across a wide range of tasks; these can be grouped into broader categories (Hagos et al., 2024; Russell & Norvig, 2022):

- core language understanding and processing tasks focus on analyzing and comprehending the context and structure of a text; examples of such tasks include NER, sentiment analysis, Part-of-Speech (POS) tagging and information retrieval and extraction;
- sequence-to-sequence and generation tasks transform an input text into a desired output sequence; examples include MT, text summarization and Question Answering (QA);
- cross-domain applications extend NLP capabilities into other industries and systems; examples include chatbots, code generation, speech recognition.

1.2.2.4. Generative AI and Large Language Models

Generative AI (GenAI) refers to "a class of algorithms and models within AI and NLP that are designed to generate new, previously unseen data that are similar to existing examples by […] learn[ing] the underlying patterns and structures present in the training data" (Hagos et al., 2024, p. 5875). GenAI encompasses multiple modalities with different architectures and training mechanisms for different purposes, such as image and video generation, personalization and so on (Hagos et al., 2024). Within the field of NLP, GenAI has been implemented mainly in the form of LLMs with the aim of processing, understanding and most of all generating meaningful human language (Hagos et al., 2024). State-of-the-art LLMs are deep neural network architectures built predominantly upon the Transformer decoder-only architecture mentioned in Section 1.2.2.3 (Jurafsky & Martin, 2025). The core intuition that makes them so effective is that natural language can best be taught by processing millions of examples and learning to "predict the next word, again and again, based on context" (Jurafsky & Martin, 2025, p. 146). This process makes up the first training phase and is called self-supervised pretraining. The following training phases, i.e. instruction tuning and preference alignment, further refine the model to follow prompts more reliably and produce more meaningful and effective outputs (Jurafsky & Martin, 2025).

Functionally, LLMs operate as text generators: given an input prompt, they predict output tokens by assigning probabilities to possible following words and then generate coherent responses (Jurafsky & Martin, 2025, p. 147). This prediction mechanism has two defining properties: (i) it is autoregressive, meaning that it takes into account the previous words in the sequence as it generates the new ones – proceeding from left to right in most languages or right-to-left for languages such as Arabic – and (ii) it is conditional, meaning that a higher probability is assigned to tokens according to the prior context provided by the input or prompt (Jurafsky & Martin, 2025).

The quality of the generated output, therefore, depends on the statistical distribution learned during the training phase, as well as sampling strategies and prompt effectiveness. Temperature sampling controls the balance between determinism and creativity: a lower temperature favors high-probability tokens and produces more predictable text, while a higher temperature increases the chance for rare

tokens to appear resulting in a more diverse text (Jurafsky & Martin, 2025, p. 156). Additionally, prompt engineering, which consists of formulating an effective instruction prompt eventually also incorporating demonstrations in order to enable some in-context learning, helps guide the model towards more precise and coherent responses (Jurafsky & Martin, 2025).

Finally, LLMs can be considered a "general purpose technology" (Eloundou et al., 2023 in Rivas Ginel & Moorkens, 2025) equipped with a "broad understanding of the world and how language is used" (Hagos et al., 2024, p. 5886) and, therefore, do not possess specialized knowledge. As a result, it might be necessary to fine-tune an LLM for a particular domain or language that was absent, or only minimally represented, in the pre-training corpus (Jurafsky & Martin, 2025).

It is important to mention, however, that current LLMs (and, to varying degrees, other AI systems) deal with a series of challenges that stem from their data-driven and probabilistic nature:

- unpredictability: while LLMs can sometimes produce accurate results, there is no mechanism to enforce or guarantee that the generated tokens are correct or true (Jurafsky & Martin, 2025). Indeed, several cases of hallucinations (i.e., the generation of "nonfactual, untruthful information" (Bang et al., 2023, p. 7)), terminological inconsistencies and mistranslations have been observed (Moorkens & Guerberof Arenas, 2024);
- opacity of deep neural networks: these systems function as black boxes, meaning that users cannot determine why and how a specific output is generated and from what data (Alla, 2025; Engel et al., 2022);
- training data limitations: much of the data is scraped from the web, which raises copyright and ownership concerns (Jurafsky & Martin, 2025). At the same time, low-resource languages or niche domains are less digitally represented, leading to a lack of training data to build efficient models (Moorkens & Guerberof Arenas, 2024);
- data quality and bias: training corpora should be of high quality and undergo rigorous preprocessing steps – such as deduplication, tokenization, alignment and named entity tagging (van der Meer, 2024) – as well as safety filtering to remove toxic content and personal information (Jurafsky & Martin, 2025). Even after these processes, the perpetration from the training data to the generated output of gender, racial and social bias and stereotypes has been observed (Jurafsky & Martin, 2025);
- privacy concerns: despite the cleaning and filtering stages, training corpora can contain personal data that may resurface during text generation (Hagos et al., 2024; Jurafsky & Martin, 2025). In addition, users may inadvertently provide sensitive information while interacting with AI systems, which might then be used to further train the models (OpenAI, 2025).

1.2.2.5. Hybrid AI: Bridging Deterministic and Probabilistic Approaches

Hybrid AI is an innovative paradigm that integrates the strengths of deterministic approaches to overcome the limitations of purely data-driven probabilistic models (Shah, 2022). These models, as mentioned previously, can be highly accurate at identifying patterns in unstructured data, but often lack transparency and struggle to be consistent; deterministic systems, instead, rely on formal logic and therefore are not flexible nor adaptable, but provide the missing robustness of transparency and consistency (Shah, 2022).

In practical applications with LLMs, this can translate to solutions that involve (Rajgarhia et al., 2025): (i) a deterministic pre-processing step, such as data cleaning and structured pattern identification, (ii) a context-aware processing step that "leverages the LLM's semantic understanding to resolve ambiguities that are intractable for rules alone", and once again (iii) deterministic post-processing to validate, refine and standardize the LLM's output.

1.3. AI and Automation in the Translation and Language Industry

1.3.1. Automation across Language Industry Workflows

As briefly mentioned in the Introduction, the Translation and Language Industry at large has undergone fundamental changes driven by rapid technological advancements and automation, which have reshaped workflows, roles and the conception of language services (Doherty, 2016; van der Meer, 2024). The so-called technological turn has been marked by several defining innovations that have permeated not only translation itself, but also the surrounding processes.

The first major turn occurred in 1990s with the introduction of CAT tools, centered on Translation Memories (TMs) and Termbases (TBs) (Doherty, 2016; Massey et al., 2024). The second turn followed soon after and saw the rise of MT, from its early – and rudimentary – rule-based form to its statistical evolution fueled by the data stored in TMs (Granell & Chaume, 2023). The current and still ongoing major turn is marked by the integration of DL and ML technologies, mainly in the form of Neural MT (NMT) and LLMs (Massey et al., 2024). The ever-growing application of these systems in the language industry is enabling various forms of human-computer interaction (Briva-Iglesias & O'Brien, 2022) and language automation (Declercq & Egdom, 2023), allowing for "the large-scale application of AI in the language industry" (Massey et al., 2024, p. 78) and dramatically altering traditional workflows and roles (Pym & Torres-Simón, 2021).

In response to these technological advances, a shift from the traditional "human-in-the-loop" paradigm to the more specialized "expert-in-the-loop" paradigm has been reported: language professionals are required to possess full subject matter and language expertise to add value and complement the work of machines (Faes & Massey, 2024). In addition, new trainees are urged to

develop abstract skills such as flexibility, adaptability, creativity, problem-solving, interpersonal communication and critical thinking (Autor, 2015; Pym & Torres-Simón, 2021), as well as technical skills (Faes & Massey, 2024).

When it comes to linguistic production, contemporary translation relies heavily on translation technologies. CAT tools are the most widely employed, as they "increase productivity in translation and maintain consistency of their linguistic data across a growing number of languages and countries" (Doherty, 2016, p. 950); they rely on two key features: TMs, which store previously translated segment pairs that can be reused, and TBs, which are terminological databases containing specialized terminology (Kappus, 2024). In recent years, NMT has further pushed the boundaries of the industry: MT is defined as a tool designed to convert sentences from a source language (input) into corresponding sentences in a target language (output) (Granell & Chaume, 2023); this output is then checked and eventually edited by professional human translators. This workflow, called Machine Translation Post-Editing (MTPE), has already permeated most translation areas, so much so that "for the first time in history, 2020 was the year in which there were more post-editing than traditional translation projects on its platform" (Briva-Iglesias & O'Brien, 2022, p. 19). However, MTPE has also been associated with a "fitness for purpose" approach (Bowker 2020 in Moorkens & Guerberof Arenas, 2024), according to which translations are associated to purpose (what the content is for), value (how important the content is), shelf-life (how long the content will remain relevant and useful) and risk levels (how harmful potential errors could be) and therefore the amount of MT present in the workflow, if any, varies: on the lower spectrum of the above-mentioned criteria, content such as online reviews and tweets might be published as raw MT output; on the higher spectrum, content such as legal or medical documents requires human translation possibly aided by technology (Moorkens & Guerberof Arenas, 2024).

The area of WfM has also been touched by automation: Translation Management Systems (TMSs) "streamline and automate many of the manual processes involved in the translation and localization of content" (Kappus, 2024, p. 127); these tools provide project managers, translators and clients with a centralized system where job requests can be submitted, processed, sorted, carried out and delivered (Kappus, 2024). Finally, automation is starting to extend into other nearby fields, such as Automatic Speech Recognition (ASR) to "enable multilingual speech and apply it to text-based language technologies" (Moorkens & Guerberof Arenas, 2024, p. 71) or Audiovisual Translation (AVT) involving fully automatic dubbing, i.e. where AI performs the translation and transforms lip movements to match the target language audio (Granell & Chaume, 2023).

The theoretical foundations of automation discussed above can be directly observed in these technologies: CAT tools and TMS, for example, exemplify rule-based automation, where

deterministic processes streamline non-cognitive tasks (e.g., file preparation and segment concordance), while NMT systems exemplify cognitive automation, producing probabilistic output texts through ML. Together, these approaches can be complementary: for instance, a TMS may automatically send a file to an MT engine for translation and then display the raw output for the human post-editing step.

1.3.2. Diversification of Services

The shift toward AI and extensive digitalization is fundamentally restructuring the language industry. Diversification is not simply a trend, but a necessary strategy of adaptation in response to market pressures and changing working conditions, leading to the identification of tasks and sectors in which "the human factor will be essential and will add value" (Briva-Iglesias & O'Brien, 2022, p. 35). In this context, Language Service Providers (LSPs) are increasingly expanding their offerings beyond the core services of translation and interpreting into highly specialized, adjacent domains that emphasize technology, data, creativity and user interaction.

The domain of AI and data services has emerged as a central pillar of the modern language industry because the efficiency of current technologies, particularly LLMs and NMT, relies on massive amounts of high-quality data (van der Meer, 2024). LSPs now offer services across the entire lifecycle of linguistic data: (i) generation and collection, both manually and automatically by crawling the web (Moorkens & Guerberof Arenas, 2024); (ii) annotation and labelling, which involves assigning labels to instances such as documents, sentences or words to train AI systems through the supervised learning approach (Briva-Iglesias & O'Brien, 2022); and (iii) validation and curation, i.e. cleaning, formatting and preparing data for models (van der Meer, 2024). Moreover, human experts remain essential to refine model output by addressing and correcting errors or biases that they might inherit from the training data: for example, linguists and translators can help identify linguistic and cultural issues that may result in "machine translationese" (Vanmassenhove, Shterionov & Gwilliam 2021 in Briva-Iglesias & O'Brien, 2022). But beyond these services, the language industry is moving into fields previously considered prerogative of computer science, such as NLP, allowing linguists to contribute with their expertise to a host of new models and applications (Briva-Iglesias & O'Brien, 2022, p. 31).

Other services, instead, emphasize creativity and cultural knowledge, which are skills that cannot yet be fully substituted by machines. Such services include, for example, (i) transcreation, which requires in-depth understanding of the target culture, (ii) digital marketing and content creation, which rely on excellent communication skills, (iii) Search Engine Optimization (SEO) localization, which demands marketing knowledge and strategic insight and (iv) video game localization, which

combines “in-depth knowledge of the gaming industry, the genre of the game, its brand and voice, as well of the target culture […] to produce appropriate localization” (Briva-Iglesias & O’Brien, 2022).

Finally, diversification encompasses technologically driven solutions to manage the complex ecosystem of software, professionals and clients that surround the language industry. These include (i) specific technical solutions that connect vendors and professionals, such as Content Management Systems (CMS) or Product Information Management Systems (PIMS) (Kappus, 2024), (ii) services dealing with workflow planning and resource allocation (Faes & Massey, 2024), (iii) resource or corpora management tools for the translation process, such as Terminology Management Systems (TMS) and terminology extraction solutions (Angelone et al., 2024; Brandt, 2024) and (iv) pre-editing or source content optimization solutions, including formatting, spell checking, references and consistency checks (Brandt, 2024) as well as anonymization of sensitive information especially in MT-enhanced workflows (van der Meer, 2024). Table 1.1 offers a summary of the above-mentioned services and what they typically entail.

Service	**Typical tasks**
AI and Data Services (Moorkens & Guerberof Arenas, 2024; Briva-Iglesias & O’Brien, 2022; van der Meer, 2024)	• Data generation and collection • Data annotation and labelling for supervised ML • Data validation and curation • Error/bias correction • NLP models development or consultancy
Creative and Cultural Services (Briva-Iglesias & O’Brien, 2022)	• Transcreation • Digital marketing and content creation • SEO and videogame localization
Technical/Workflow Management Services (Kappus, 2024; Faes & Massey, 2024; Angelone et al., 2024; Brandt, 2024; van der Meer, 2024)	• CMS/PIMS implementation • Workflow and resource planning • Terminology management and extraction • Corpus/resource management • Source content optimization

Table 1.1. Summary of new services and tasks in the Language Industry.

1.3.3. New Professional Skills and Profiles

Building on the diversification of services outlined in Section 1.3.2, it is crucial to examine how language professionals themselves are adapting to these changes. Humans’ advantage over machines is especially marked in those abstract skills that are hard to automate (Autor, 2015), therefore, despite

advances in ML, core cognitive and creative skills remain central: creativity and originality are indispensable for jobs requiring transcreation or adaptation, such as marketing, videogame localization and audiovisual translation (Briva-Iglesias & O'Brien, 2022); critical thinking and problem-solving are essential, for example, to evaluate the output of an MT engine (Faes & Massey, 2024); adaptability and metaliteracy are needed to keep up with the "evolving nature of the digital landscape […] and technological advancements" (Granell & Chaume, 2023, p. 32); and communication and interpersonal skills are vital for interacting with clients and team members, as well as presenting strategies, methodologies and results (Autor, 2015; Pym & Torres-Simón, 2021).

Alongside these human-centric skills, the automation and AI-driven shift highlights the need for new technical and digital competences among language and translation professionals (Faes & Massey, 2024). Professional must be proficient in traditional translation technologies, such as CAT tools, TMSs and especially the process of MTPE (Kappus, 2024). They must also be comfortable with technology and IT tasks at large, including handling diverse file formats, web-based platforms and cloud collaborative environments (Briva-Iglesias & O'Brien, 2022; Kappus, 2024). Data skills are increasingly important, as professionals are tasked with gathering, processing and annotating high-quality linguistic datasets or corpora to use in a variety of subsequent AI-related tasks (Briva-Iglesias & O'Brien, 2022; Kappus, 2024). Some professionals focus on training and advisory roles, "not just in the use of technology, but also in understanding and communicating the underlying processes of technology [in order] to advise on their appropriate (and inappropriate) uses" (Moorkens & Guerberof Arenas, 2024, p. 89). In addition, programming and scripting skills, particularly in languages such as Python, allow professionals to access more technical roles within the industry (Faes & Massey, 2024; Kappus, 2024). Finally, the ability to leverage GenAI tools with the right prompts to generate effective outputs is becoming a necessary skill (Faes & Massey, 2024).

As the industry develops rapidly, and with it the required competences, there is no unified taxonomy of job positions and titles; instead, recent studies outline several combinations of the above-mentioned skills. For example, Faes and Massey (2024) distinguish two orientations, rather than fixed profiles: on one side the Expert Linguist and Translator, a profile requiring absolute mastery of all components of language combined with specialized domain expertise; and on the other side, the Linguistic Manager, who "understand the linguistics part, but [is] not [a] language expert per se" (Faes & Massey, 2024, p.30). The UPSKILLS project (Miličević Petrović et al., 2021), instead, proposes an overarching "modular and adaptable" professional profile, that of the Language Data and Project Specialist, articulated into four sub-profiles: on one side, the Language Data Analyst and Language Data Scientist with a focus on operative tasks and research; on the other, the Language Data Manager and Language Project Manager with a focus on planning and supervising tasks and

workflows. Briva-Iglesias and O'Brien (2022) report multiple branches emerging inside and across the industry and propose a hybrid profile, the Language Engineer, that combines "profound knowledge of language and culture" with "programming languages and AI" and is therefore able to "adapt to the new professional prospects arising from today's digitalization and automation by having a specific tech-symbiotic role" (Briva-Iglesias & O'Brien, 2022, p. 40).

1.4. Environmental, Social and Economic Dimensions of AI and Automation

1.4.1. Environmental Sustainability

The energy and resource demands of modern AI pose significant environmental concerns, necessitating a focus on sustainable deployment. Schwartz et al. (2000) propose a distinction between Red AI, which is purely focused "on performance without concern for cost or efficiency", and Green AI, which takes "into account the computational cost, encouraging a reduction in resources spent" (Moorkens & Guerberof Arenas, 2024, p. 85). In practice, the trajectory of AI development has leaned toward the Red AI model, whereby models have grown exponentially in size, reflecting a pursuit of performance gains over conscious development and usage (Moorkens & Guerberof Arenas, 2024). The main environmental concerns include (Moorkens, Castilho, et al., 2024; Moorkens & Guerberof Arenas, 2024): (i) the significant power and water resources to maintain data centers; (ii) the mining of rare earth minerals to build hardware; (iii) the inappropriate disposal of outdated hardware; (iv) the large emissions of carbon dioxide during training; and (v) the risk of exposure to toxic materials for workers in the manufacturing cycle of such technologies.

Some scholars have pointed out that energy requirements and emissions depend on multiple factors, such as time of day or employment of renewable energy (Shterinov & Vanmassenhove, 2022 and Dodge et al., 2022 in Moorkens & Guerberof Arenas, 2024), and others note that these technologies may offset energy consumption or emissions elsewhere (Moorkens & Guerberof Arenas, 2024). However, it is a difficult comparison to make, thus the broader question of "whether sustainable AI is possible in an economy targeted at perpetual growth" (Heilinger, Kempt & Nagel, 2023 in Moorkens & Guerberof Arenas, 2024, p. 85) remains open.

1.4.2. Social and Ethical Implications

Social and ethical concerns focus primarily on the direct impact of automation and AI on human labor, professional integrity, accountability, individual rights and discrimination. Many professionals report a certain level of anxiety related to AI and automation. Alla (2025) identifies two main forms: skill obsolescence anxiety, as professionals fear that "their existing competencies may become irrelevant as AI capabilities expand" (Alla, 2025, p. 1028), and adaptation stress, which makes professionals resist adapting to new technologies and workflows. Among translators, in particular, automation

anxiety relates to uncertainties tied to an unstable market, distrust in the quality and reliability of new technologies, fear of being replaced by AI, reduced rates and the possibility of the profession itself losing prestige (ELIS, 2025; Rivas Ginel & Moorkens, 2025; Vieira, 2020). Moreover, many report feelings of loss of professional autonomy and dehumanization as processes and workflows are incorporated into machines that reduce their control and decision-making capabilities, while still "bearing responsibility for guarding the quality of what these complex workflows produce", as is the case in the MTPE workflow for instance (Alla, 2025; Carmo & Koponen, 2024, p. 213). This last argument connects with the issue of the opacity of contemporary AI systems discussed in Section 1.2.2.4: the difficulty that users encounter in understanding AI output leads to a sense of mistrust in its capabilities, especially in the face of evident errors in high-stake translations (Rivas Ginel & Moorkens, 2025).

Again in Section 1.2.2.4, it was noted that AI models can reproduce the biases embedded in training data and generate "biased, unfair, and discriminatory" outputs (Hagos et al., 2024, p. 5886). Such biased responses are not merely theoretical concerns, but can have tangible, real-world consequences by perpetuating stereotypes and reinforcing inequalities. For example, Jurafsky and Martin (2025) recount a few gender-related issues with MT systems: (i) they tend to use masculine forms when translating references to an individual whose gender is not explicit; (ii) they frequently rely on cultural stereotypes when translating from gender-neutral languages assigning male gender to individuals performing traditionally male-dominated occupations and vice versa; and (iii) they produce less accurate translations for sentences containing non-stereotypical gender roles.

1.4.3. Economic, Societal and Cultural Implications

Economic and societal issues pertain to macro-level structural changes in the labor market, wealth distribution, regulatory failures and systemic cultural impacts. Automation has contributed to the so-called polarization of the job market: since the tasks less suitable for automation (i.e. abstract and manual tasks) are "generally found at opposite ends of the occupational skill spectrum—in professional, managerial, and technical occupations on the one hand, and in service and laborer occupations on the other", it has been observed that automation tends to eliminate "middle-wage, middle education jobs" favoring instead "high-education, high-wage jobs at one end and low-education, low-wage jobs at the other end" (Autor, 2015, p. 12). This phenomenon highlights the replacement vs augmentation theory introduced in Section 1.2.1.1: the rapid decrease in middle-skill jobs supports the replacement theory, while the simultaneous increase of high-skill and low-skill jobs aligns with the augmentation perspective. Acemoglu et al. (2022) investigate the impact of AI in the hiring behavior of "AI-exposed" companies and find that their rapid increase in the use of AI has led

to an increase in their demand for AI-skilled workers and at the same time a decrease in the demand for previously sought skills; moreover, a general decrease in non-AI and overall hiring has been observed. The bottom line is, therefore, that the job market is rapidly changing and adjusting to the introduction of AI in various ways.

In the language and translation industry, this trend is clearly visible: the industry continues to grow in spite of the integration of automation and AI, yet

> "data on remuneration indicate structural wage dispersion in professional translation services, with some signs that this dispersion may increase in certain market segments as automated workflows and translation technologies are adopted more by large language-service providers more than by smaller companies and individual freelancers" (Pym & Torres-Simón, 2021, p. 39).

The use of TMs and MT systems by large companies is often employed "to justify a discount on the per-word rate by arguing that translators can translate faster by post-editing" (Briva-Iglesias, 2020 in Briva-Iglesias & O'Brien, 2022, p. 22). This devaluation applied by large companies translates into a strong price pressure for freelancers and small companies, who cannot compete with such rates, thereby reinforcing structural wage dispersion (ELIS, 2025; Pym & Torres-Simón, 2021).

Regarding the impact of AI on culture and language, linguists and translators' doubts are worth acknowledging. Automated translation still struggles to faithfully render "the intricate interplay of cultural subtleties, contexts such as historical and sociological ones, and the unique voice of an author", potentially weakening or distorting cultural richness and instead promoting standardization (Declercq & Egdom, 2023, p. 56). This standardization is evident at both lexical and syntactic levels: LLM-generated content and NMT translations are less linguistically rich, favoring instead the reproduction of patterns present in the training data (Moorkens & Guerberof Arenas, 2024). For example, internet users and media outlets have spotted several expressions, phrases and even special characters that are so over-used by certain LLMs that immediately instill the doubt in the reader that the text has been written (or edited) by an LLM (Csutoras, 2025; Paschalidis, 2025). Moreover, even post-edited translations appear to be flatter than translations from scratch, because the output produced by NMT influences the translator and inhibits creativity (Moorkens & Guerberof Arenas, 2024). Finally, as mentioned in Section 1.2.2.4, the availability of data in widely spoken and digitalized languages raises concerns over the "marginalization of lesser-resourced languages" both concretely, as those languages do not get access to the benefits of new technologies because systems do not have enough data to be trained effectively and therefore perform poorly (Moorkens & Guerberof Arenas, 2024), and ideologically, by "perpetuat[ing] a hierarchical linguistic structure" (Declercq & Egdom, 2023, p. 57).

Ultimately, from privacy concerns and cultural risks to workplace inequalities and wage dispersion, professionals – and the wider public – increasingly call for regulatory solutions, as the

“current policy landscape and regulatory gaps reveal significant inconsistencies in how different jurisdictions approach AI regulation in workplace contexts, with most existing frameworks focusing on consumer protection rather than worker rights and workplace equity” (Alla, 2025, p. 1031).

CHAPTER 2: TEXT ANONYMIZATION

2.1. Chapter Overview

This chapter examines the first use case developed during the internship at Munich Re, focusing on the anonymization of textual data for the translation workflow. Building on Chapter 1's discussion on automation and AI-driven solutions in the language industry, it emphasizes their application toward data protection and regulatory compliance. It presents an LLM-based anonymization tool tailored to the specific needs of Language Services (LS) that integrates in the already existing workflows to ensure compliance with external regulatory frameworks and internal guidelines.

This chapter is structured as follows: in Section 2.2, the concept of anonymization is introduced, together with its regulatory background, key technical implementations and challenges; Section 2.3 presents a preliminary exploration of NLP models for the detection of personal information, outlining their methodology and limitations in this context; Section 2.4 describes the design and implementation of the LLM-based anonymization tool, detailing its architecture, workflow and core components; Section 2.5 reports and compares the results obtained with NLP models and the proposed LLM-based approach; Section 2.6 discusses these results, highlighting strengths, weaknesses and trade-offs; and finally, Section 2.7 presents additional implementations and use cases that extend the tool beyond the original workflow.

2.2. Introduction

2.2.1. Context and Motivation

As data generation and collection practices continue to grow, protecting sensitive information has become a major concern for all players involved, from individual citizens to companies and institutions (Asimopoulos et al., 2024). For this reason, regulatory bodies have introduced or updated their frameworks to deal with data protection. In 2016, the European Union provided a cohesive and comprehensive regulation on "the protection of natural persons with regard to the processing of personal data and on the free movement of such data" (GDPR, 2016) that aims at defining what personal data is and how it has to be handled and stored. They establish that personal data – also known in other contexts as Personally Identifiable Information (PII) – is

> "any information relating to an identified or identifiable natural person […] such as a name, an identification number, location data, an online identifier or […] one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person" (GDPR, 2016, p. 33).

PII should only be collected for specific purposes, processed in a safe and secure manner and stored for no longer than necessary (GDPR, 2016, art. 5). Other regulatory systems have their own requirements: for example, the Health Insurance Portability and Accountability Act (HIPPAA) in the

United States outlines 18 categories of direct identifiers that must be removed to de-identify health information (Kocaman et al., 2023).

In LS, a wide range of documents are processed for translation and revision, some of which contain PII. According to company guidelines, clients are responsible for submitting documents that have already been anonymized; however, this requirement is not always met in practice. As a result, the need for an internal workflow solution to detect and process those documents that do not comply with these criteria was identified. This was even more paramount for two reasons: first, LS employs Trados Studio/Enterprise, a CAT tool that stores all translation segments in Translation Memories, making it essential to prevent personal data from being inadvertently retained in these resources[4]. Second, Trados provides access to a DeepL engine to pre-translate documents; as DeepL is a third-party service, any transmission of PII to it, especially when it comes to confidential documents, should be avoided. Moreover, the solution needed to be able to (i) reliably detect many kinds of PII across multiple languages, (ii) integrate seamlessly with Trados Enterprise, (iii) be accessible and easy to use for colleagues without coding expertise and (iv) comply with internal security guidelines.

2.2.2. Core Concepts of Text Anonymization

Text anonymization, or data anonymization, is "the process by which personal data are irreversibly altered so that a data subject can no longer be identified directly or indirectly, either by the controller or in collaboration with any other party" (Pissarra et al., 2024, p. 1). The fundamental purpose of text anonymization is to ensure compliance with data protection laws and policies and, at the same time, share and utilize textual data for translation, research, analysis and other tasks (Mozes & Kleinberg, 2021).

The overall process typically involves two main steps: effectively identifying sensitive data and subsequently masking it (Asimopoulos et al., 2024). The initial step often applies Named Entity Recognition (NER) models to identify such data (more in Section 2.2.3), then various anonymization strategies are implemented based on the data itself and the privacy requirements. Key techniques include (Asimopoulos et al., 2024): (i) removal, i.e. substitution of personal data with generic placeholders (e.g. [ANON], [REDACTED], etc.); (ii) categorization, i.e. substitution of personal data with labels that reference directly the type of PII (e.g. [NAME], [DATE OF BIRTH], etc.); and (iii) pseudonymization, i.e. substitution of PII with randomly generated alternatives of the same type (e.g. "Max Mustermann" instead of "John Smith").

The development of effective anonymization solutions involves major challenges, notably achieving a good balance between preserving privacy and maintaining data integrity. Privacy

[4] Placeholder tags are stored in TMs only with their ID, but no other metadata is stored, therefore the PII is inaccessible.

protection and mitigation of the re-identification risk is paramount; on the matter, scholars claim that technical evaluation metrics, like precision and recall, are insufficient to evaluate an anonymization system as missing even a single important entity can invalidate the entire anonymization process and reveal a person's identity (Mozes & Kleinberg, 2021)[5]. Preserving the integrity of the text is necessary to ensure its usefulness for the secondary tasks it is being anonymized for (e.g., translation, model fine-tuning, linguistic analysis, etc.); highly aggressive anonymization methods may result in high utility loss (Mozes & Kleinberg, 2021, p. 4). In real world scenarios, automating anonymization is a complex task: more sophisticated anonymization solutions generally achieve higher performance, but they are "always prone to miss certain entities" and can be resource-intensive to implement (Pissarra et al., 2024, p. 2).

2.2.3. Background on NLP Approaches to Anonymization

Automated text anonymization first relies on models capable of identifying sensitive data in unstructured text. NER is the most commonly used approach for the identification and classification of specific words and phrases as entities (Pissarra et al., 2024). NER models can be trained to recognize different kinds of entities: from basic ones such as personal names, locations and organizations (Tjong Kim Sang, 2002) to Protected Health Information (PHI) (Kocaman et al., 2023) or financial data (Watson et al., 2024).

NER-based PII detection is typically formulated as a sequence-labelling task, where each token in a sentence is assigned a label indicating whether it belongs to a sensitive entity (Asimopoulos et al., 2024). Most sequence-labelling tasks operate on token-level annotations using schemes such as BIO, a schema best suited for handling multi-word expressions as it assigns "O" to all non-entity tokens in a sequence, "B" to the token at the beginning of an entity and "I" to the token(s) inside an entity (Ramshaw & Marcus, 1995).

Over time, various models have been developed to perform NER. Conditional Random Fields (CRFs) are an example of early statistical models (Lafferty et al., 2001): they "encode[] a conditional probability distribution" to learn associations between a set of manually engineered features that represent linguistic, orthographic, positional and contextual properties of tokens and their labels (Huang et al., 2015; Zhang et al., 2008). Subsequent deep learning models, such as Long Short-Term Memory (LSTM) networks, leverage the advances of word embeddings to capture contextual dependencies without extensive feature engineering (Asimopoulos et al., 2024). A specific implementation is the BiLSTM-CRF (Huang et al., 2015), which combines a bidirectional LSTM that

[5] Although this limitation is well-established in the literature, most research on NER and anonymization continues to report precision, recall and F1 for comparability. This work follows that convention while explicitly acknowledging their insufficiency and complementing them with qualitative analysis to provide a more comprehensive evaluation.

captures sequential context from both "past and future input features" and a CRF output layer that enforces a logical progression across BIO tags.

More recently, Transformer-based encoder architectures, such as BERT (Devlin et al., 2019), have set state-of-the-art for token-classification tasks thanks to their ability to process text in parallel and learn complex relationships between information regardless of distance (Asimopoulos et al., 2024). However, fine-tuning such models for domain-specific NER can require significant computational resources and high-quality annotated datasets (Mishra et al., 2025). Decoder architectures and generative LLMs enable prompt-based NER without task-specific training, but they struggle with consistency and can introduce hallucinations (Asimopoulos et al., 2024; Pissarra et al., 2024).

2.3. Preliminary Exploration of NLP Models

Given that LS at Munich Re did not have access to either a GPU or a custom dataset, a preliminary exploration of less resource-intensive NLP models was carried out to evaluate whether they could provide an efficient solution for our use case[6]. This exploration focused on two sequence-labelling models mentioned in Section 2.2.3: a CRF model and a BiLSTM-CRF model.

To train and evaluate these models, the Gretel.ai Synthetic PII Finance Multilingual dataset was selected (Watson et al., 2024). It contains 55,940 AI-generated documents annotated with 29 PII types across seven languages. This dataset made an interesting candidate because of its size, multilingual support and especially domain relevance and PII coverage, as other standard NER datasets would not have sufficed for our anonymization needs.

However, several limitations emerged. First, the dataset is highly imbalanced both across languages, as English documents account for half of the dataset (Table 2.2 details the number of documents per language) and across PII types, as values such as name and date occur tens of thousands of times, whereas others appear only a few hundred times (Table 2.3 summarizes the occurrences in the training set per PII type). These imbalances pose a significant challenge for models that rely on supervised learning, as they tend to "favor the majority class disproportionately" (Altalhan et al., 2025, p. 13687).

[6] These models were tested only on PII detection and basic span masking, as inference was performed on plain textual inputs, leaving complex document format parsing and full anonymization with tags to later stages.

Language	Number of documents
English	28,910
Spanish	4,609
Swedish	4,543
German	4,530
Italian	4,473
Dutch	4,449
French	4,426
Total	55,940

Table 2.2. Gretel.ai dataset languages and number of documents.

PII type	Occurrences	PII type	Occurrences
account_pin	1,266	iban	1,814
api_key	922	Ipv4	1,591
bank_routing_number	1,452	ipv6	1,191
bban	1,477	last_name	1,594
company	56,338	local_latlng	802
credit_card_number	1,224	name	89,642
credit_card_security_code	1,275	passport_number	1,426
customer_id	1,823	password	789
date	75,830	phone_number	8,277
date_of_birth	2,339	ssn	1,313
date_time	767	street_address	37,845
driver_license_number	1,269	swift_bic_code	1,917
email	12,914	time	15,735
employee_id	1,696	user_name	906
first_name	2,565		

Table 2.3. Gretel.ai PII types and occurrences in the dataset.

A second limitation concerns the annotation format: the Gretel.ai dataset provides span-based annotations, whereas both models require the dataset to be annotated using the BIO scheme (Table 2.4 illustrates the difference between span and BIO annotation).

“The Agreement is signed on January 2nd 2025 by John Smith.”	
"pii_spans": [{"type": "DATE", "start": 27, "end": 42}, {"type": "NAME", "start": 47, "end": 56}]	"BIO": [{"word": "The", "tag": "O"}, {"word": "Agreement", "tag": "O"}, {"word": "is", "tag": "O"}, {"word": "signed", "tag": "O"}, {"word": "on", "tag": "O"}, {"word": "January", "tag": "B-DATE"}, {"word": "2nd", "tag": "I-DATE"}, {"word": "2025", "tag": "I-DATE"}, {"word": "by”, "tag": "O"}, {"word": "John", "tag": "B-NAME"}, {"word": "Smith", "tag": "I-NAME"}]

Table 2.4. Span vs BIO annotation example.

Therefore, to adapt the dataset for both the CRF and BiLSTM-CRF models, some preprocessing steps were applied. First, several PII types were not relevant for our use case and were therefore discarded from the training data (account_pin, api_key, company, customer_id, date, date_time, ipv4, ipv6, local_latlng, password, time, user_name). Subsequently, `spaCy` (Honnibal & Montani, 2017) was used to tokenize the dataset and then the original span boundaries had to be aligned with the token boundaries produced during tokenization; this step introduced some discrepancies, as not all spans cleanly aligned and had to be dropped. Finally, sentences containing rare PII classes were oversampled via duplication, a common resampling technique to mitigate class imbalance (Altalhan et al., 2025).

After preprocessing, for the CRF linguistic and orthographic features were generated for each token and the model was trained to associate those features to the correct labels. The model was trained for four hours on standard CPU hardware and achieved a comprehensive F1 score of 0.68 on the test set. Performance varied widely across PII types: some entities reached relatively high scores such as 0.77, while many others had F1 scores of 0.00, likely due to insufficient representation or inadequate feature engineering (see Appendix A Table 2.14 for detailed per entity metrics). Figure 2.2. summarizes the full pipeline.

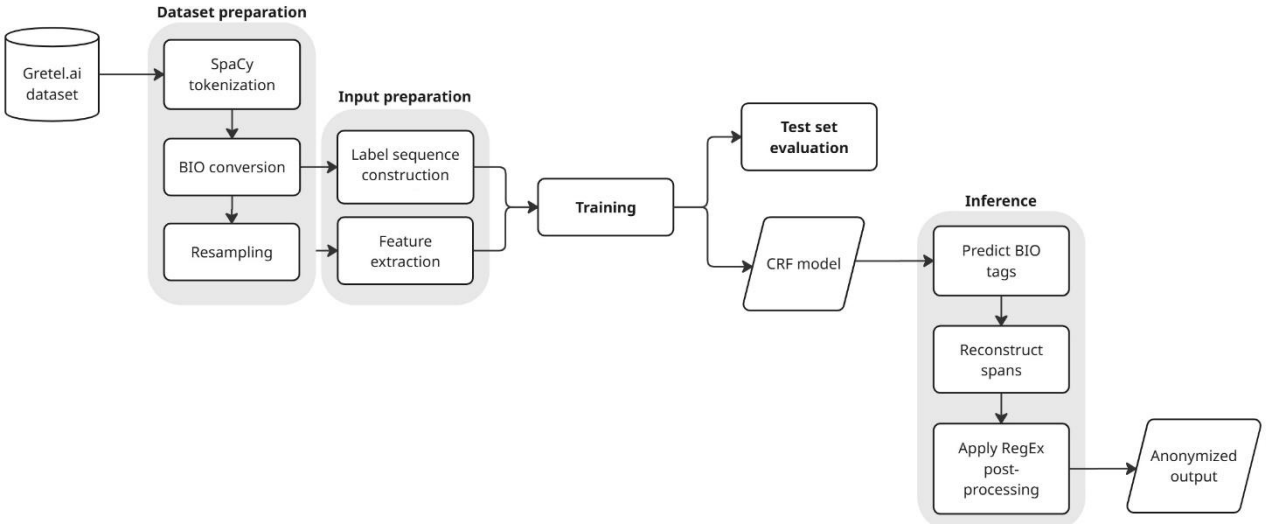

Figure 2.2. CRF model pipeline.

The BiLSTM-CRF instead required additional preprocessing steps beyond the shared pipeline: instead of manually engineered features, pre-trained word embeddings (FastText by Bojanowski et al., 2017) were used and dynamic padding and chunking were applied to handle sentences of different lengths. The model was trained for six hours on standard CPU hardware and achieved a comprehensive F1 score of 0.89 on the test set, with most PII types scoring above 0.75 and only one type scoring 0.00, again likely due to data scarcity (see Appendix A Table 2.15 for detailed per entity metrics). Figure 2.3 offers an overview of the pipeline.

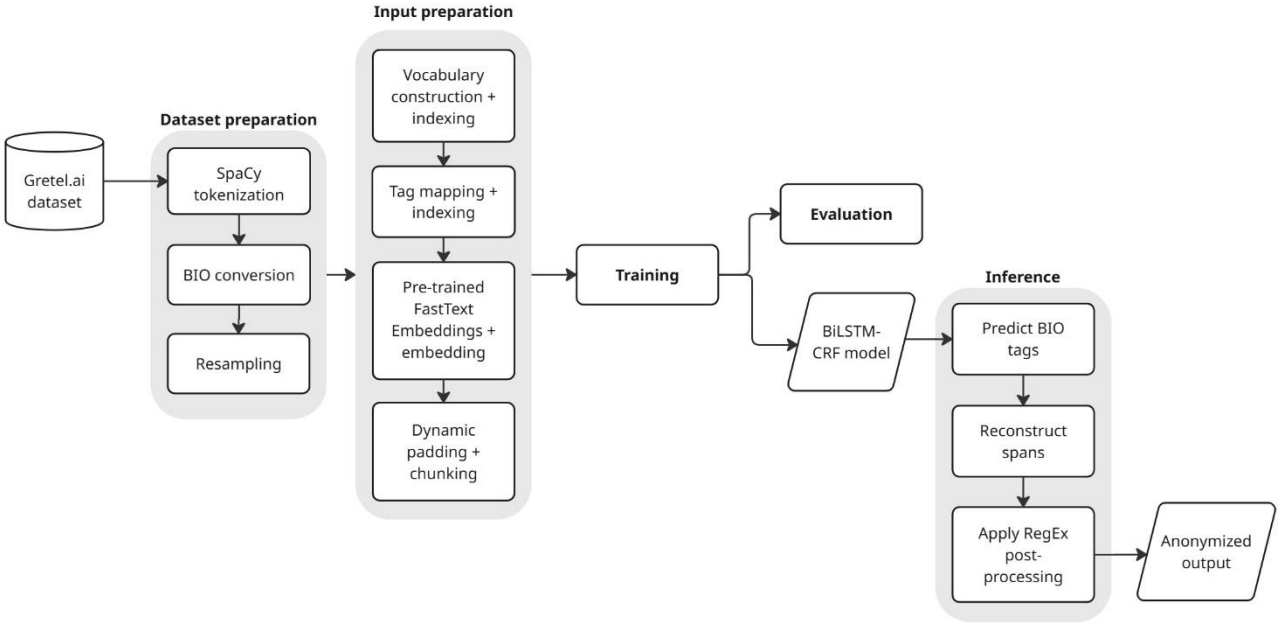

Figure 2.3. BiLSTM-CRF model pipeline.

Both models were tested on new and unseen synthetic sentences with common PII types for each supported language to see how their performance on simulated real-world data compares to the evaluation carried out on the test set. Since some PII types were either absent from the dataset or too sparsely represented to be learned effectively (employee_id, tax_id, vehicle_plate and id_card), they were added or supported via RegEx at this stage[7]. This approach, however, was inherently limited, as RegEx-based detection can only capture perfectly formatted spans that match the query exactly, lacking a certain flexibility and context-awareness (Dhar, 2025). As summarized in Table 2.5, the CRF achieved an average F1 score of 0.54[8], a decrease of 0.14 compared to the test set performance, with high scoring languages such as Swedish and Italian at 0.75 and low scoring such as Dutch at 0.25 and French at 0.29. BiLSTM-CRF achieved an average of 0.77[8], a decrease of 0.12 compared to the test set evaluation, with English as the highest scoring with a perfect 1.00 and French as the lowest with a 0.44.

Language	**CRF (F1)**	**BiLSTM-CRF (F1)**
English	0.67	**1.00**
Spanish	0.50	**0.75**
Swedish	0.75	**0.89**
German	0.60	**0.80**
Italian	0.75	**0.89**
Dutch	0.25	**0.67**
French	0.29	**0.44**
Average	0.54	**0.77**

Table 2.5. CRF vs BiLSTM-CRF F1 scores.

Qualitative analysis revealed that the CRF could only reliably detect name entities but failed to fully identify most other PII types; the other entities (vehicle plates, employee IDs and a German tax ID) were detected through RegEx rules. Moreover, no other variation of the SSNs was detected by either model and street addresses were often missed or only half identified by the BiLSTM-CRF. Both models rely heavily on the quality of the training data and the CRF, in particular, also on manually engineered features. As a result, they struggle to generalize to PII types that exhibit high structural variability (e.g., street addresses, identification numbers) or that are sparsely represented in the dataset, limiting their ability to learn consistent patterns across languages and formats.

[7] Only patterns belonging to the three most commonly translated languages in LS were implemented: German, British English and American English.

[8] Gold standard was annotated manually and F1 scores were computed calculating True Positives (correctly identified entities), False Negatives (missed entities) and False Positives (incorrectly identified entities).

Overall, these experiments with traditional models provided valuable insights but ultimately demonstrated that these approaches were insufficient for the requirements of our anonymization process. To make these models efficient enough for our use case, several improvements would have to be implemented, such as a custom dataset, more refined BIO alignment and resampling techniques or extensive RegEx support (Mishra et al., 2025). On the other hand, an encoder-only Transformer architecture such as BERT (Devlin et al., 2019), specialized in understanding language and the relationship between words, could have been a good alternative solution; however, fine-tuning a BERT model for the PII types and languages needed for our use case would have been too resource intensive (Mishra et al., 2025).

2.4. Methodology: Implementation of LLM-based approach

The above-mentioned findings and constraints, together with the availability of two self-hosted open-source LLMs[9] accessible via API, motivated the transition to an LLM-based approach, which promised strong generalization capabilities, multilingual support and the ability to handle diverse PII types and documents (Asimopoulos et al., 2024). To meet the specific requirements outlined in Section 2.2.1, a dedicated anonymization solution was designed and implemented: PII Anonymizer is a Python-based Graphical User Interface bundled as a Windows executable which leverages the self-hosted LLM to identify personal data in SDLXLIFF files and replace it with Trados-compatible tags. The following subsections give an overview of how the tool operates step by step and how it is integrated into the translation workflow; they also offer a deeper explanation of the main components and core stages of the pipeline, outlining its modular structure and explaining specific design decisions and technical solutions.

2.4.1. Application Overview and Workflow

To develop a solution that was easy to use and would integrate seamlessly with Trados Enterprise, PII Anonymizer was built in a way that would require very few clicks on the Project Managers' part and would process the input file exactly as it was downloaded from the platform and produce an output file that could be directly re-uploaded into the platform. Internally, this process is orchestrated by a controller module, which manages the execution of the pipeline's core stages in the correct sequence, ensures smooth interaction with the interface and retrieves and passes the necessary configuration information to the appropriate functions. Figure 2.4 schematically outlines the workflow steps.

[9] For confidentiality reasons, the exact names of the models and their parameters cannot be disclosed. Importantly, all experiments were carried out with the same model, chosen as it was the best performing one according to preliminary tests.

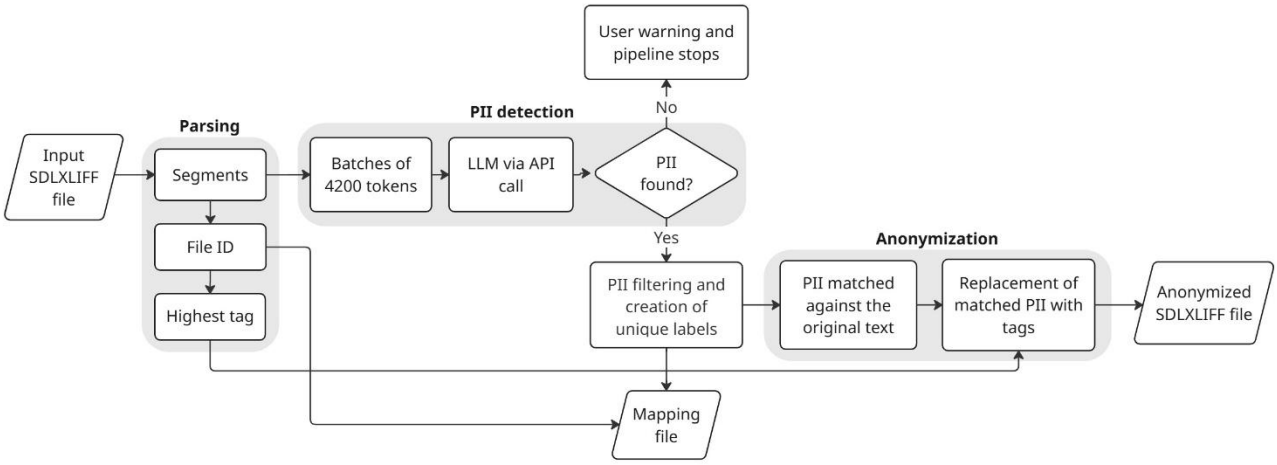

Figure 2.4. Overview of LLM-driven pipeline.

Once a translation project reaches the "PII Check" stage after the generation of the bilingual SDLXLIFF file on Trados Enterprise, the Project Managers download the file as is and place it in a dedicated *To Anonymize* folder on the internal SharePoint. After opening the PII Anonymizer tool and clicking on *Anonymize File(s)*, said folder opens automatically and Project Managers can select the file to be processed[10]. This file is first parsed, meaning that the SDLXLIFF structure is read and the source segments needed for PII detection, as well as other metadata for the subsequent steps, are extracted.

After parsing, the extracted segments are forwarded, together with the API credentials and the prompt, to the LLM for PII detection. To comply with internal rate limits, the segments are divided into batches of a fixed number of tokens and each batch is sent in turn with a mandatory delay between requests. During this phase, the interface displays progress messages such as "Sending batch x of y…" and "Waiting x seconds until next batch…" accompanied by an active countdown, giving the user clear indications of the progress and expected duration. The LLM then returns a structured list of detected PII grouped by label, which is post-processed into a consolidated PII map.

During anonymization, each detected PII is matched against the original text and replaced with Trados-compatible tags, ensuring that the SDLXLIFF structure remains intact. Once this step is completed, the anonymized file is saved in a dedicated *Anonymized* folder on the SharePoint and the corresponding PII mapping file in a *Mappings* folder. Finally, the tool opens the *Anonymized* folder automatically, so that the Project Managers can immediately replace the original file in Trados Enterprise with the anonymized version. After manually confirming that all PII have been anonymized and no formatting discrepancies are present, the "PII Check" step is marked as completed.

[10] The tool supports the selection of multiple files as it automatically creates a queue to process one file at a time.

2.4.2. Graphical User Interface

The PII Anonymizer interface was intentionally designed in a simple and straightforward manner (Figure 2.5): the *Anonymize File(s)* button opens the *To Anonymize* SharePoint directory where the original files are stored and initiates the anonymization process after file selection; the *Anonymized Files Archive* and *Mappings Archive* buttons open the corresponding *Anonymized* and *Mappings* directories, where the anonymized files and the mappings are stored; the *Exit* button simply closes the tool. At the bottom, the progress bar and status label provide continuous feedback to the user on the current step of the pipeline. In practice, the UI is coded in a layout module, where the main window, its style and all interactive elements are defined, and linked to the core processing modules via the controller.

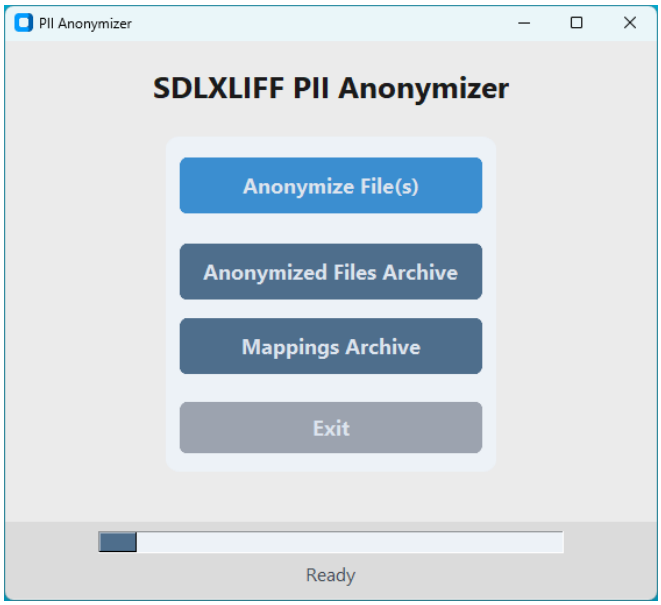

Figure 2.5. PII Anonymizer interface.

2.4.3. Security and Configuration

The implementation relies primarily on standard Python libraries complemented by some third-party packages (see Table 2.16 in Appendix A for the full overview). With the exception of the API calls, all processing occurs locally, ensuring that no document content is transmitted to external services and that the system adheres to Munich Re's confidentiality and security requirements.

The configuration component is handled in a single config folder which contains the `.env` files used to store API URLs and credentials for each user[11]. It also includes the `settings.yaml` file,

[11] To use the API endpoints, each team member has to have their own credentials. A system is implemented whereby each team member must add their individual `.env` file named after their own employee ID in the configuration folder. The tool recognizes the Window user (namely, the employee ID) and looks for the correct file. If it is not found, a default `.env` file with default credentials is also provided so that the tool can still function.

which specifies the available models, which one is currently selected as default and the full prompt used for PII detection. In addition, the settings file defines the paths to the SharePoint directories, so that they are automatically linked to the tool and can be accessed directly.

Because the application is bundled as a folder using `PyInstaller`, the configuration folder remains accessible and modifiable to all team members. Although storing credentials in editable `.env` files may appear to be a potential weakness, this design choice is necessary, as it allows each user to add or update their own API credentials. Moreover, the ability to access the settings file is also useful to easily modify the prompt or default model without rebuilding the executable. The application folder is stored on the internal SharePoint, where access is granted to specific employees.

2.4.4. Parsing

Parsing is the first step amongst the core activities after the input file is selected and uploaded. It is implemented in the `parser` module, where the `lxml` package is used to traverse the SDLXLIFF XML structure and extract the required elements: (i) it retrieves the unique file ID embedded in the file header, which is later stored in the mapping file to ensure correspondence between the two; (ii) it identifies all existing `<tag>` elements under `<tag-defs>` and determines the highest tag ID present, so new tags created in the anonymization stage can continue the numbering sequence without conflict; and (iii) it extracts only the textual content inside `<source>` elements to send to the LLM, as they perform best on natural language rather than structured markup (OpenAI et al., 2024) and to keep the token count as low as possible. In addition to these elements, the function returns the complete XML tree that will later be used to reconstruct the original file with the new tags.

2.4.5. PII detection via LLM

PII detection is the second main step in the pipeline and is implemented in `pii_detector` module. It receives the list of source segments extracted during parsing and prepares them in batches of 4200 tokens to respect the API limitation of 5000 tokens per minute. Since token counting is performed using `spaCy`'s multi-language tokenizer, as the self-hosted model's tokenizer is not publicly available, the threshold is intentionally conservative to account for estimation inaccuracies and prompt tokens. For each batch, two messages are created: a system message that assigns the role of an "expert at detecting Personally Identifiable Information (PII)" to the LLM (Schulhoff et al., 2025, p. 12) and a user message containing the prompt and the text segments (see Appendix A Table 2.17 and Table 2.18 for the full system message and prompt). The prompt opens with a directive, i.e. an explicit instruction to follow a task (Schulhoff et al., 2025), and a comprehensive list of the 18 specific

PII types the model is asked to detect[12], some with detailed constraints on what should and should not be included (e.g., excluding titles from names or institutional addresses from street addresses). It also contains further instructions to handle multilingual contexts, disambiguate number sequences and account for formatting exceptions (Schulhoff et al., 2025, p. 32). This represents an implementation of zero-shot prompting, designed to leverage the LLM's pre-existing knowledge without requiring task-specific examples (Sahoo et al., 2025). The choice of this kind of prompting strategy was made because the PII categories included are several and providing effective examples for each would have increased the token count significantly; moreover, some PII types are common enough that examples felt superfluous[13]. A minimal few-shot prompting snippet was provided only for the formatting exceptions (Sahoo et al., 2025). In addition, style instructions are provided to instruct the model to preserve the exact form, spelling and structure of the PII rather than normalizing it (Schulhoff et al., 2025, p. 12). Finally, the prompt provides detailed and strict output formatting guidelines (Schulhoff et al., 2025), specifying how the JSON output must be structured and instructing it to avoid including comments; however, the occasional comment still occurred and therefore a predefined JSON schema is included in the API payload to enforce a valid JSON output.

Each batch is sent with a mandatory 60-second delay to the appropriate chat completion endpoint. Batches are processed asynchronously using background threads to keep the GUI responsive: in practice, this mechanism allows long-running tasks to be executed without freezing the interface, so that while each batch is being processed, the application continues to inform the user through the progress bar and status label. A five-minute timeout is implemented to handle cases where the LLM does not respond.

Finally, the LLM response is parsed and postprocessed to remove a set of predefined entities[14]. Each detected PII is assigned a unique incremental tag (e.g., "NAME_1", "NAME_2"), so that during anonymization the same PII is masked with the same unique label. When all batches are processed, the function returns the full mapping for the anonymization step or, if no PII was found during this stage, it notifies the user and terminates the pipeline.

2.4.6. Anonymization

The last step in the pipeline is the actual anonymization of the original file. The `anonymizer` module contains several main functions. The first function identifies occurrences of each PII value in the text using a four-stage matching strategy designed to encompass all variations. This is a hybrid solution

[12] The PII types are taken from the optimized Gretel.ai dataset.

[13] Nonetheless, a version with examples was tested, but it did not improve the results much, causing instead more false positives as the LLM would insert the examples into the PII map.

[14] Munich Re's short and full name, Munich Re subsidiaries' names, common titles in English and German.

designed to complement and mitigate the limitations of the LLM that, despite instructions in the prompt, may not detect each variation:

1. RegEx are used to find exact matches in the text (since exact matches are high-precision, they should be preferred when available);
2. `RapidFuzz`'s `fuzz.token_sort_ratio` is used to match reordered PII or mild spelling variations by computing a similarity score between strings of minimum 2 words[15];
3. two-word subphrases from PII values are generated to search and match partial PII[15];
4. RegEx are used to find titles (such as "Dr." or "Ms.") followed by names to help match standalone names or surnames which are missed by partial matching due to the two-word window; however, the code adjusts the start of the match to exclude the title from the candidate before adding it to the final list of matches so that it does not get anonymized.

The matched PII spans are then gathered in a final list which is sorted by span length first. When anonymizing, if a span is matched, any overlapping matches are discarded to ensure only the longest and more complete match occupies the text.

Tag creation is then handled by a subsequent function. Trados supports several types of tags, such as inline, structure, break or ghost tags; among inline tags, placeholder tags are "stand-alone tags [that] indicate the presence of non-translatable information in a segment" (RWS, 2024), which is the most suitable tag type for our anonymization needs. In the SDLXLIFF format, tags must have both a definition, which specifies all attributes and elements and goes inside the `<tag-defs>` section in the header, and an element that references the `tag id` in the body – the actual placeholder. The function creates the tag definition as displayed in Table 2.6.

```
<tag id="…">
<ph    name="LABEL_1"    line-wrap="false"    word-end="false"    seg-
hint="include">PII</ph>
<props><value key="OriginalEmbeddedContent">PII</value></props></tag>
```

Table 2.6. Tag definition structure.

The `<tag>` element is the parent container and, inside it, both `<ph>` and `<props>` appear as child elements with their own opening and closing tags. The `id` is unique and sequential and, as mentioned in Section 2.4.4, it picks up after the highest already existing `tag id` found during parsing, so that there are no conflicts with the original tags. The `<ph>` element wraps the visible placeholder text (the

[15] This is applied only to names and street addresses, as they are the two most likely PII types to appear in different forms in the text and it would be too risky to apply it to numbers. Moreover, the two-word window is implemented to keep the number of false positives amongst regular words to a minimum.

PII) and specifies attributes that define its behavior: `name` stores the unique sequential label associated with the specific PII present in the full mapping, `line-wrap="false"` prevents the placeholder from breaking across lines, `word-end="false"` indicates the placeholder does not represent word boundaries and `seg-hint="include"` suggests that Trados should include the placeholder in the segment content. Similarly, `<props>` encloses the `<value>` metadata element which stores the PII as `OriginalEmbeddedContent`; this enables de-anonymization when generating the target file, as the original content is automatically reinserted.

The placeholder tag is created in the last function and it is simply composed of `<x id="…">`, whereby the id matches the one in the tag definition. This function traverses all XML elements inside the SDLXLIFF file and replaces the matched PII with the placeholder tag, rebuilding the original text and tags surrounding the PII. This step is quite complex and unfortunately not always perfectly successful, as this type of XML format presents a vast number of tags, often nested or contiguous, making it quite difficult to preserve all elements in their right placements[16].

2.5. Results

The LLM was briefly tested on the same simple texts used to test the preliminary models and it significantly outperformed both by achieving perfect scores for all languages. However, a more robust baseline was needed, therefore the BiLSTM-CRF (the best performing between the preliminary models) was further tested against the LLM on real documents of varying length and languages, some more complex than others[17]. Though for privacy purposes the full documents and contained PII cannot be disclosed, Table 2.7 displays the F1 scores per language (and therefore per document)[18]. It is important to note that data associated with companies and public entities was not counted as PII, as they are publicly available data and according to internal guidelines do not count as PII, and the same PII reappearing in different formats was counted accordingly.

[16] It can happen that a tag or a punctuation mark is placed in the wrong position, causing formatting discrepancies with the original text. Therefore, a quality check is performed once the anonymized SDLXLIFF file is re-uploaded on Trados Enterprise.

[17] Inference was run with the models to benchmark PII detection capabilities; no parsing or full anonymization were implemented at this stage.

[18] Gold standard was annotated manually and F1 scores were computed calculating True Positives, False Negatives and False Positives.

Language	BiLSTM-CRF (F1)	LLM (F1)
Italian	0.33	**0.73**
Spanish	**0.80**	**0.80**
Swedish	0.54	**1.00**
Dutch	0.30	**0.56**
English	0.20	**1.00**
German	0.07	**0.40**
Norwegian	0.29	**0.93**
Polish	0.32	**0.86**
Portuguese	0.02	**0.50**
Average	0.32	**0.75**

Table 2.7. BiLSTM-CRF vs LLM PII detection

The results show that the LLM generally outperforms the BiLSTM-CRF across all languages, with particularly large gains for English, Norwegian and Polish. Spanish is the only exception where both models achieve the same F1 score; this can be explained by the characteristics of the Spanish document used for evaluation, as it contained mostly well-structured and frequently occurring PII types which the BiLSTM-CRF already handled relatively well.

Subsequently, the same documents were fully anonymized with PII Anonymizer to test how much the final optimized prompt and the hybrid approach improved the matching of PII – and consequently the anonymization. Table 2.8 summarizes the F1 scores per language.

Language	LLM (F1)	Hybrid (F1)
Italian	**0.73**	**0.73**
Spanish	**0.80**	**0.80**
Swedish	**1.00**	**1.00**
Dutch	0.56	**0.79**
English	**1.00**	**1.00**
German	0.40	**0.67**
Norwegian	0.93	**1.00**
Polish	**0.86**	0.75
Portuguese	0.50	**1.00**
Average	0.75	**0.86**

Table 2.8. LLM vs Hybrid anonymization F1 scores.

The results show that the hybrid anonymization pipeline improves overall performance compared to LLM-detection alone, raising the average F1 score from 0.75 to 0.86. The largest improvements occur in languages where the LLM struggled with specific PII types and boundaries, indicating that the deterministic components, such as fuzzy and partial matching, successfully compensate for gaps in the LLM's output. In contrast, languages such as Italian, Spanish, Swedish and English show no change, as the LLM already achieved high or perfect scores. Polish is the only language where performance decreases, due to increased false positives.

2.6. Discussion

The results confirm that the self-hosted LLM substantially outperforms our preliminary sequence labelling models in multilingual PII detection applied to real texts, achieving an average F1 score of 0.75 – or 0.86 with the hybrid components – compared to 0.32 for BiLSTM-CRF[19]. This advantage is particularly evident in languages not supported by the Gretel.ai dataset, where the LLM demonstrates strong adaptability. Qualitative analysis of the detected PII shows that the LLM recognizes different kinds of entities similar to a Social Security Number, like *pesel* in Polish or *codice fiscale* in Italian, as well as passport or ID numbers from different countries. It also manages to detect fewer false positives, as the BiLSTM-CRF struggles to disambiguate general number sequences from meaningful dates, document numbers and so on.

Overall, LLMs offer clear advantages, such as the ability to generalize across multiple languages and identify unexpected PII patterns without relying on separate datasets, handcrafted RegEx rules,

[19] It is important to note that the models were intentionally lightweight and trained on limited data; therefore, these findings should be interpreted strictly within this context and not universally.

or extensive fine-tuning (Asimopoulos et al., 2024). Despite these strengths, several limitations remain, due to their probabilistic and non-deterministic nature. LLMs can struggle to distinguish between personal and organizational entities and may occasionally hallucinate, inventing PII that does not appear in the source text. They also tend to normalize complex entities, reducing detailed addresses to shorter forms or reporting only full names and not abbreviations, and sometimes fail to consistently follow instructions, such as ignoring prefixes or titles. Furthermore, detection quality is highly dependent on the integrity of the source text: performance deteriorates with complex documents due to segmentation issues and formatting quirks. The choice of using a self-hosted LLM introduces additional constraints, such as mandatory delays between batches, which create a direct trade-off between accuracy and processing speed.

To address these challenges, the custom pipeline developed in this project incorporates several mitigation strategies. It ensures SDLXLIFF compatibility through robust parsing and tag handling, and employs prompt engineering strategies to reduce mislabeling, capture missing variants and improve disambiguation. Most importantly, it adopts the hybrid intelligence approach mentioned in Chapter 1 that combines the flexibility of LLMs with deterministic logic: while the LLM excels at handling ambiguous cases and detecting patterns across languages, deterministic components are used to validate outputs, match differently formatted PII, filter entities and construct a comprehensive PII map. Indeed, the second test proves that the optimized approach improves the anonymization capabilities of the LLM: the prompt helps avoid some misidentification by better guiding the LLM towards disambiguation of some categories of PII (for example, fewer dates are recognized as dates of birth and big company names are mostly not detected); while the hybrid components manage to match fragmented or inconsistent PII that were not identified by the LLM as variants (for example, partial addresses or single names and surnames after a title are mostly matched and anonymized).

From the perspective outlined in Chapter 1, this use case represents a case of task-level substitution driven primarily by resource constraints rather than quality enhancement. Indeed, while a human operator would likely perform a more complete anonymization, carrying out such a process manually for large volumes of documents would be excessively time-consuming and operationally unfeasible. Automation is therefore adopted not to surpass human performance, but to provide a practical compromise between accuracy, processing time and available resources. In this context, substitution can be seen as a strategy for repetitive, high-volume tasks that would otherwise require a disproportionate mental workload from the human operator part.

At the same time, it is important to acknowledge that no systematic anonymization process had been implemented internally prior to this solution; therefore, while high-risk tasks such as these require careful consideration when automated, this case illustrates how the absence of automation

may itself constitute a higher risk than the controller deployment of such a solution. Finally, this system exemplifies a mixed automation strategy, as anonymization constitutes only one step within a broader workflow and remains subject to human oversight and judgment, both in the prior decision of whether a PII check is required in the first place and in the subsequent review of the anonymized output.

Nonetheless, there are several paths through which this solution could be further improved. One possible direction involves investing in the fine-tuning of a Transformer-based architecture or an LLM using a custom, domain-specific dataset. This approach, while resource-intensive, could yield substantial gains, as recent systematic surveys consistently rank these among the top-performing approaches, especially when tailored to specific domains (Asimopoulos et al., 2024). Such alternatives could be trained to handle unique linguistic and formatting patterns, thereby reducing reliance on prompt engineering and mitigating the variability inherent in general-purpose LLMs. Alternatively, improvements could be pursued by refining the current hybrid approach: this might involve integrating more advanced LLMs or enhancing the post-processing and matching logic to better compensate for the limitations of the LLM's outputs.

2.7. Additional Implementations and Use Cases

Beyond the main SDLXLIFF anonymization workflow, the underlying architecture developed in this project was reused to build two additional anonymization solutions for different use cases. These are not extensions of the original tool, but independent implementations that leverage the same modular design and LLM-based detection logic.

The first implementation targets the anonymization of legacy Translation Memories in TMX format. Although TMX is structurally simpler than SDLXLIFF, these files are often substantially larger, which amplifies the impact of the one-minute delay required between LLM batches. In some cases, a full PII detection pass could take up to 20 hours. To make this process operationally feasible, a resumable batching mechanism was introduced: the mapping file records the index of the last processed batch, allowing the system to resume detection from the interruption point rather than restarting from the beginning. Finally, anonymization is performed by replacing detected PII with a generic placeholder ("XXX"), which is sufficient for legacy TMs. As this process was only meant to be carried out once to clean the TMs before re-importing them, this solution was implemented as a script without a GUI.

The second implementation adapts the architecture to support anonymization of other document formats, namely DOCX, PDF, XLSX and PPTX. Each format required a dedicated parser capable of extracting text content while preserving structural integrity, but the subsequent stages of the pipeline

– PII detection, mapping construction, and anonymization – remain largely unchanged. The PII are replaced with category-specific placeholders (e.g. “[NAME_1]”, “[PHONE_NUMBER_1]”) derived from the PII map rather than with Trados-compatible tags. This solution is also implemented as a GUI in a similar style to the SDLXLIFF PII Anonymizer.

These additional implementations support two distinct anonymization needs within the organization: on one side, they make it possible to anonymize legacy Translation Memories before re-importing them into Trados Enterprise, aligning with the principle outlined in Section 2.2 that personal data should not be stored indefinitely within software applications; on the other side, they enable the anonymization of reports, presentations, spreadsheets and other document types that may contain sensitive information, thereby extending the applicability of the approach beyond translation workflows and making it accessible to other departments that do not work with SDLXLIFF files. Together, these solutions illustrate the flexibility and reusability of the underlying architecture: the same core logic can be adapted to heterogeneous file formats and operational contexts with only minimal adjustments.

CHAPTER 3: TERMBASE EXPANSION

3.1. Chapter Overview

This chapter explores the application of LLM-driven approaches to Termbase (TB) expansion and enrichment within a corporate language department. While Chapter 2 discussed a concrete, highly-automated LLM-based anonymization tool to comply with regulatory requirements about privacy and data protection, this chapter shifts the focus to an exploratory, semi-automated LLM-assisted pipeline that enables the expansion and enrichment of LS' TB, while emphasizing a human-in-the-loop approach to ensure terminological accuracy.

The chapter is organized as follows: Section 3.2 outlines the theoretical foundations of terminology management, including key notions from terminology theory, Automatic Term Extraction (ATE), and metadata augmentation; Section 3.3 introduces the methodological framework and describes the overall architecture of the proposed pipeline, including data collection and the five main processing phases (subdomain identification, term extraction, concept consolidation, metadata augmentation, and relation identification); Section 3.4 reports the quantitative and qualitative results of the pipeline; Section 3.5 discusses the findings, highlighting methodological choices and limitations of the LLM-assisted workflow, as well as suggestions for future work.

3.2. Introduction

3.2.1. Context and Motivation

Terminology is essential for "supporting the common understanding and exchange of specialized knowledge across languages and cultures" (Mohamed et al., 2025, p. 16). Within the translation industry, the precise and consistent use of terms is not merely a matter of readability, but also a matter of professional and social responsibility, as mistranslations in high-stakes fields can result in serious consequences (Kageura & Marshman, 2019).

At Munich Re, the TB functions as a centralized repository of structured linguistic data that not only enables consistent translations, but can also support other business units: from enabling uniform communication in press releases and internal documentation, to offering precise definitions for policies and contracts, and to supplying domain- and corporate-specific lexical resources for the improvement of the self-hosted LLMs. Munich Re's terminology platform, Quickterm (Kaleidoscope, 2025), currently contains more than 70 thousand entries across six languages (English, German, Italian, French, Spanish and Chinese) and over 50 domains and sub-domains.

Given the central role of the TB, its entries must be standardized, conceptually coherent and sufficiently informative to support consistent use. One area that, however, remains underrepresented and insufficiently standardized is that of investment terminology. Given that investments constitute a

core pillar of the new Munich Re's multi-year strategic plan for 2030 (Munich Re, 2025), the lack of structured and high-quality terminological resources in this area represents a critical gap. The present project addresses this gap by (i) identifying and grouping existing entries related to the investment subdomain (currently under the broader "Finance" domain), (ii) extracting additional investment-related terms to expand coverage, and (iii) enriching both existing and newly extracted entries with relevant metadata.

To support terminologists in identifying and maintaining domain-specific terminological resources, the industry has long explored automatic and semi-automatic solutions (Yuan et al., 2017). Recent advances in AI, particularly in Natural Language Processing (NLP) and LLMs, offer new opportunities to support TB management tasks, including term extraction and metadata enrichment, but also introduce new methodological challenges that require systematic investigation (Tran et al., 2023).

As discussed more extensively in Section 2.3, two main constraints informed the methodological approach adopted for this use case: (i) the insufficient results obtained with the lightweight preliminary models trained in-house and (ii) the infeasibility of fine-tuning a Transformer-based Pretrained Language Model (PLM) with the available resources. Combined with the availability of self-hosted open-source LLMs, these considerations ultimately motivated the adoption of an LLM-based solution, with its inherent strengths and limitations.

3.2.2. Core Concepts of Terminology and Termbase Expansion

3.2.2.1. Terminology

The International Organization for Standardization (ISO) defines terminology as the "designation of a defined concept in a special language by a linguistic expression" (ISO, 2019). While on the surface level terms are not different from common words, they are employed in specialized domains to "facilitate unambiguous communication" and "are consolidated in relation to others [in] a conceptual system" (Kageura & Marshman, 2019, p. 61). Two common criteria that are generally used to distinguish terms from the general vocabulary are termhood, i.e. "the degree to which a linguistic unit is a term of a domain", and unithood, i.e. "the degree to which a sequence of tokens is a linguistic unit" (Kageura & Umino, 1996 in Judea et al., 2014, p. 291). That is because terms are typical of a specialized domain and can consist of either a single element (simple term) or multiple words (complex terms) (Conrado et al., 2013). Although most terms are nouns or nominal phrases, other word classes, such as verbs and adjectives, may be included if relevant for the domain (Conrado et al., 2013).

A defining principle of terminology that distinguishes it from general lexicography is concept orientation: in a terminological database, each entry "describes or corresponds to one concept" and "should contain all the terms that denote the concept that it describes[, i.e.] synonyms, spelling variants, abbreviations and any other variation or 'way' that people refer to that concept" (Warburton, 2024, p. 187). In contrast, lexicography is word-oriented: dictionary entries are organized around a single lexical form and list its various meanings, even when they correspond to different underlying concepts (Warburton, 2024). Moreover, because TBs are typically multilingual, the entries should also report the equivalent terms with their respective variants in other languages (Warburton, 2024).

3.2.2.2. Automatic Term Extraction

Term Extraction (TE) is "the task of identifying and ranking domain-specific words or multi-word expressions that represent key concepts within a corpus" (Chun et al., 2025, p. 1). It is a fundamental step for building terminological resources and can support several other tasks such as information retrieval, Machine Translation (MT) and document indexing (Kucza et al., 2018; Tran et al., 2023).

As mentioned in Section 3.2.1, researchers have tried automating this process to reduce the manual effort that this task requires. Automatic Term Extraction (ATE) refers to the use of computational methods to extract lists of candidate terms that are subsequently validated by a human expert (Tran et al., 2023). These systems have gone through continuous refinements over the years (Tran et al., 2023): from early rule-based approaches that mainly employed linguistic patterns to detect terms but were heavily language-dependent, to statistical approaches that introduced criteria such as termhood, unithood and C-value (Frantzi et al., 2000) but remained sensitive to data quality and quantity, and to hybrid methods that combined the two. However, these systems still faced major issues (Conrado et al., 2013): on one side, noise and silence, i.e. the extraction of strings that are not real terms or the failure to extract real terms; and on the other side, an inflated number of candidate terms that has to be validated by human terminologists, costing time and effort.

The shift toward ML, PLMs and LLMs has opened new paths, as their generalization abilities could create new opportunities to automate terminology management in a more efficient and scalable way than previous methods (Mohamed et al., 2025). In particular, PLMs can be fine-tuned on domain-specific corpora and are therefore considered state-of-the-art in in-domain scenarios, as they typically outperform other models when performing tasks such as ATE on texts belonging to the domain on which they were fine-tuned (Tran et al., 2023). LLMs, by contrast, are well suited to cross-domain and/or low-resource scenarios: their broad knowledge base enables them to adapt and generalize across texts from diverse fields and to perform effectively even when little or no domain-specific data is available. However, recent studies highlight various limitations, such as boundary errors that result in a high number of partial matches, over-prediction of general terms and under-prediction of rare

terms and hallucinations of candidate terms (Breton et al., 2025; Chun et al., 2025; Touvron et al., 2023).

Regardless of the underlying architecture, ATE systems typically follow a high-level "two-step procedure: (1) extracting a list of candidate terms, and (2) determining which candidate terms are correct" (Tran et al., 2023, p. 1). In the first step, the primary goal is to identify all potential lexical units within a corpus that might be domain-specific terms, prioritizing recall over precision (Warburton, 2024). The second step can be subsequent to the first, such as in statistical models, where linguistic or statistical measures were then computed on the extracted units to determine whether they classified as terms. Alternatively, modern neural models and LLM-based approaches collapse the two steps into a single one: instead of first generating a list of candidate units and then evaluating them, they identify and classify term spans directly within the sentence during the same pass (Touvron et al., 2023; Tran et al., 2023). Despite advances in automation, however, ATE is not yet reliable enough to fully replace professional expertise; literature emphasizes the need for "a human-in-the-loop approach […] to recover terms that were discarded or remove unwanted terms" (Di Nunzio et al. 2023 in Wissik, 2025, p. 129).

3.2.2.3. Automatic Metadata Augmentation

While ATE focuses on identifying candidate terms, metadata augmentation aims to enrich term entries with additional conceptual, contextual and linguistic information. In this work, automatic metadata augmentation refers to different automated solutions that enrich the terminological entries in the TB with supplementary information, such as definitions, context sentences, conceptual relations and grammatical attributes (Wissik, 2025).

Automatic definition generation aims to "describe the meanings of unfamiliar words or phrases" either within a specific context or for all possible contexts (Ide et al., 2026, p. 2). Recent studies have explored the use of LLMs for this task – either fine-tuned or through prompt engineering – with promising but still imperfect results that require manual intervention, as models are prone to hallucinate meanings and generate inconsistent or overly similar definitions across senses (Ide et al., 2026; Di Nunzio et al. 2024 in Mohamed et al., 2025). Beyond definitions, LLMs have also been applied to support the "automatic discovery of […] concept relations allowing for the scalable construction of domain-specific knowledge graphs"; this includes identifying hierarchical relations (e.g., hypernyms and hyponyms) and non-hierarchical relations (e.g., synonyms and antonyms) (Mohamed et al., 2025, p. 17).

Tasks such as context extraction, which involves retrieving authentic sentences from the same corpus from which the term is extracted to show how the term is used in practice, or grammatical metadata assignment, such as Part-of-Speech (POS) tags or term type, can also be automated using

different approaches, from deterministic rules to statistical NLP models and even LLMs (Warburton, 2024).

3.3. Methodology: Implementation of LLM-based pipeline

The following sections describe in detail the pipeline designed to semi-automatically enhance the quality and completeness of the TB for the investment subdomain. The development was guided by several primary objectives: (i) ensuring terminological precision, consistency and normalization across entries; (ii) complying with internal regulations and guidelines; (iii) guaranteeing traceability of sources and reproducibility; (iv) enabling scalability across domains; and (v) maintaining human oversight for all ambiguous or important decisions. Evaluation is conducted qualitatively by a human reviewer at multiple stages of the pipeline and is based on domain relevance, terminological correctness, adequacy of the term form and usefulness of the entry in the corporate environment. Importantly, this use case operates monolingually: only English entries from the TB were retained and only English documents were chosen for the extraction corpus[20].

From a high level perspective (Figure 3.6), the pipeline follows a modular, multi-phase architecture: (i) the existing TB is narrowed down to the relevant investment-related entries that serve as a starting point, (ii) new candidate terms and their relative context sentence are extracted from a corpus of internal documentation and then validated by a professional terminologist[21], (iii) these new terms are consolidated to group equivalents together under a single entry and then merged with the original TB, (iv) additional metadata features (i.e. definition and term type[22]) are added to both the existing entries when they were missing and to the new entries, (v) finally, vertical and horizontal relationships are established among the enriched set of entries to support a concept-oriented structure.

[20] Given the exploratory nature of the project, limiting the setup to a single language allowed the pipeline to be developed and evaluated under controlled conditions before extending it to multilingual scenarios.

[21] In this pipeline, context extraction is performed during the term extraction stage and not during the augmentation stage.

[22] Referred to as “type of designation” in Munich Re’s TB.

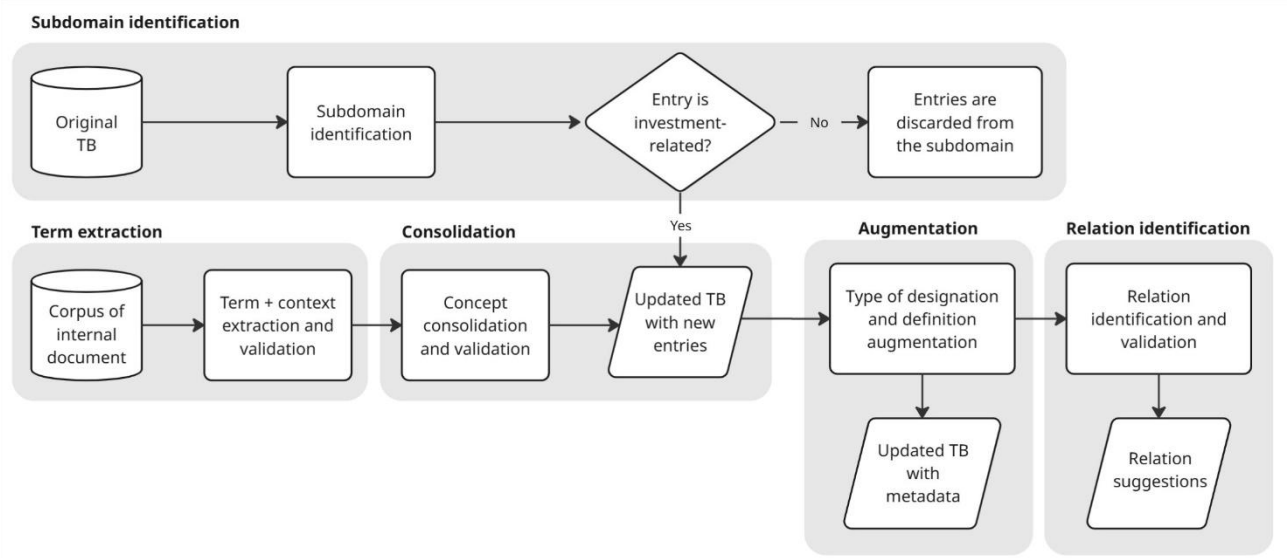

Figure 3.6. Core phases of Termbase Expansion pipeline.

The design of the pipeline was shaped by several practical constraints, including strict privacy requirements and limited computational resources (see Section 3.5). In addition, this use case was conceived as a pilot project to experiment with the resources available within LS and to assess the capabilities of the self-hosted LLMs. For this reason, the pipeline is composed of stand-alone Python modules that offer interactive options directly through the Visual Studio Code console, without a dedicated Graphical User Interface. The lightweight implementation leaves room for future developments, as ongoing initiatives may open new directions for the automation of terminology management.

3.3.1. Data Acquisition

Ten documents with a combined total of 60,250 words were collected to make up the corpus on which to perform term extraction. Since the TB is specific to Munich Re, mostly internal documentation was considered or external documentation used internally as guidelines. The Group Investment Management (GIM), the Financial and Regulatory Reporting (FRR) and Integrated Risk Management (IRM) Departments have all published internal documents on rules, guidelines and standards pertaining to the investment domain. Moreover, MEAG – Munich Re's wealth management company – provides additional documents, such as regulatory information. Finally, the Competence Centre for the German Investment Fund Industry (BVI)[23] publishes positions, i.e. "statements and viewpoints", to "promote sensible regulation of the fund business" that are consulted by Munich Re on a regular basis. Table 3.9 summarizes the number of documents and words per text type.

[23] https://www.bvi.de/en/positions/

Text type	Documents	Words
Positions[24]	1	2,821
Guideline	4	42,283
Policy	4	10,621
Code of Conduct	1	4,795
Total	**10**	**60,250**

Table 3.9. Statistics of corpus text types.

3.3.2. Phase 1: Subdomain identification

Since the TB did not contain a subdomain focused on investment, the first stage of the pipeline consisted in narrowing down the subset of entries under the general domain of Finance (1,827 terms) to identify only those entries relevant to the specific subject area of interest. The subdomain identification phase consisted of several steps.

First, a list of investment-specific keywords taken from two authoritative online glossaries[25] was compiled to serve as a reliable set of anchor terms representing the subdomain. This list was matched against all entries in the Finance domain: each entry was checked to determine whether its term contained at least one investment-related keyword. The output was split into two separate sheets: matched entries (1,130), i.e. terms that did contain at least one keyword representing the initial candidate subset of investment-related terms, and unmatched entries (697), i.e. terms that did not contain any of the keywords and therefore required further review.

For the unmatched entries, the relevant information (term and, when available, definition) was extracted and formatted into a structured list; this subset represented entries whose domain affiliation was unclear, either because the linguistic unit was too long and complex or because it simply did not appear among the keywords and therefore required semantic classification rather than simple string matching. The list was processed by an LLM using a contrastive prompt designed to distinguish investment-specific concepts from general financial terminology: the prompt defined inclusion and exclusion criteria and explicitly instructed the LLM to also provide the excluded terms, which improved the model's output[26] (see Table 3.19 in Appendix B for the full prompt). This step produced

[24] Most are in German and therefore not suitable for this English pipeline.

[25] From the European Central Bank (https://www.ecb.europa.eu/services/glossary/html/glossa.en.html) and J.P. Morgan Asset Management (https://am.jpmorgan.com/us/en/asset-management/adv/resources/glossary-of-investment-terms/).

[26] When only asked to provide the included terms, it would frequently list all terms as either included or excluded or hallucinate new investment-related terms. When asked to provide both lists, it would rarely hallucinate new investment-related terms and achieve a better balance of included vs excluded terms (214 and 483 respectively).

an expanded set of investment-related terms that was turned into a new keyword list so that the previously unmatched entries could be extracted from the dataset.

This produced a refined investment-specific subset (314) that was merged with the initial subset of matched entries to produce the final dataset (1,444). Taken together, these steps showed that most Finance-domain entries were indeed investment-related. These terms and their definition, when present, were then gathered to make up the final glossary, useful for deduplication during the extraction step. The final output was manually inspected to ensure that no unrelated terms were included.

This hybrid approach combining keyword matching and LLM-based semantic classification was adopted to maximize the results: keyword matching offered a quick classification method based on authoritative glossaries and effectively captured well-established investment terms; however, many entries in the original TB were complex multi-word expressions or less straightforward concepts that did not appear in these glossaries and therefore needed to be assessed further to determine whether they belonged to the investment subdomain.

3.3.3. Phase 2: Extraction

Once the TB was narrowed down to the specific subdomain and a glossary was created, the following phase consisted in extracting terms from the corpus of internal documentation. The self-hosted LLMs' own semantic understanding of unstructured texts was leveraged to perform ATE, complemented by careful prompt engineering and deterministic pre- and post-processing.

Because the workflow also required retrieving the exact context sentences in which the terms appeared, the initial prompting strategy instructed the LLM to identify both terms and their context sentences. However, a second approach was later tested to evaluate whether prompting the LLM to return just the terms without their context sentences would improve recall. A terminologist evaluated the output and found that the terms-only approach did indeed achieve higher recall (161 terms on the test document), but also introduced more noise, including generic items (e.g., "quality"), malformed surface forms (e.g., "investment proces") and truncated multi-word expressions (e.g., "risk limit" and "trigger manual" as separate terms); by contrast, the term-context method yielded fewer items (121 terms on the test document) but demonstrated higher precision and better term forms. Although the terms-only setup provided more candidate terms, its higher noise level increased the burden on human validators. Given that this experiment was conducted in an environment with an existing termbase, precision and reduction of manual filtering were prioritized over maximum recall; for this reason, despite its lower coverage, the term-context approach was ultimately selected for the workflow.

Once the extraction strategy was selected, the pipeline proceeded through a sequence of steps. The process begins with parsing of the corpus documents (PDF, DOCX, PPTX, XLSX, and SDLXLIFF are all supported) and the extracted text is segmented into sentences. Noise (such as boilerplate, bullet points, fragments, and special characters) is removed and a subset of higher-quality segments (without headers, footers, table of content) is further isolated to serve as a pool to repair LLM's generated context sentences, ensuring that the sentences are structurally well-formed and conceptually informative. In addition, the code collects all acronyms of length 2 to 6 characters that appear in parenthesis in the segments and builds an inventory that is later used to ensure consistent normalization, irrespective of how the LLM returns them.

The core extraction step relies on the LLM: to respect token limits, sentences are grouped into batches of roughly 4200 tokens[27] and each batch is sent to the model together with the extraction prompt and a JSON schema enforcing a valid output structure[28]. The prompt (see Table 3.20 in Appendix B) follows an instruction and few-shot design principle: it defines what a term is and how to identify one, provides positive and negative instructions to guide the identification of terms belonging to the investment domain and avoid general financial terms, and specifies formatting and casing rules. A final example summarizes the output behavior expected of the LLM. The response gets parsed and unique terms are collected across batches to avoid duplicates. This step produced 385 candidate terms across the ten documents.

A substantial deterministic post-processing stage follows, which is essential for mitigating the variability and limitations of the LLM's output. Each extracted term is canonicalized by (i) splitting acronyms that appear in parenthesis after the long form, so that they count as individual terms, (ii) uppercasing acronyms that appear in the inventory, in case the LLM returned them lowercased, and (iii) singularizing the head noun, if the LLM returned it plural. This produces a consistent display form of each term[29]. Although the LLM is prompted to return a context sentence together with each term, its output is not always structurally reliable; for this reason, a deterministic repair mechanism is applied only when the extracted sentence does not meet predefined quality criteria: if the sentence does not contain the term (singular/plural variant), lacks final punctuation or contains header-like elements, bullet points or fewer than six words beyond the term itself, the script replaces it with the first valid sentence found in the subset of segments that satisfies these criteria. If no such sentence is found, the term is not dropped, rather the placeholder "NO_VALID_CONTEXT" is inserted so that

[27] Though the LLM supports up to 5000 tokens per minute, the number is intentionally conservative to account for estimation inaccuracies when tokenizing and for prompt tokens.

[28] The script retries failed calls, so that no batches' content gets lost if the LLM does not respond or hits the timeout limit of 5 minutes.

[29] This is the only form of canonicalization applied to the terms, since other variants would be accepted in the TB as different designations relating to the same concept/entry (e.g. hyphenated variants, abbreviations, reordered variants).

the search can be performed manually by the terminologist. This repair step ensures that every term receives the best possible context sentence available. Frequencies are then computed combining singular and plural variants of each term to return a comprehensive count to support better judgment calls during the human validation step. Finally, the reference glossary from the subdomain identification phase is used to remove terms already present in the TB and the resulting dataset is collected in a file. To preserve traceability in subsequent phases and in the TB itself, an HTML-formatted link to the original source document is placed at the beginning of the file.

Following extraction, a human validation step is carried out interactively in the console so that the candidate terms can be assessed. Each term is displayed with its context sentence and frequency count and the terminologist can (i) keep the entry as is, (ii) discard it completely, (iii) edit the term and/or the context sentence inline, (ii) save it for a further check after the validation process is completed or (iv) skip the entry so that it gets moved to the end of the queue until it is classified. At the end of the process, the terms are all saved to different files (`yes`, `no` and `check`) and a log of the validation process is also produced. Once the terminologist has checked the terms saved for review, these are added to the file containing all validated terms, which will be used for the subsequent phase.

The entire extraction process was performed on each of the ten documents one by one, so that it was easier to track terms and their source file and less cognitively heavy to perform validation on the candidates. Moreover, at the end of each validation step, the new terms were added to the reference glossary, which is used to filter duplicate terms during a subsequent extraction phase on a new document.

3.3.4. Phase 3: Concept consolidation

In the consolidation phase terms that refer to the same concept were grouped together. This serves two main purposes: it supports the creation of the concept-oriented TB described in Section 3.2.2.1 by creating entries that contain term variants of a given concept, and it facilitates the subsequent definition augmentation phase, since definitions are added at the entry level.

After the extraction phase produced a file with all the validated terms and their context sentences, this file is sent to the LLM[30] with a prompt (see Table 3.21 in Appendix B) that instructs the model to determine which of the terms refer to the same underlying concept, narrowing the focus down to true conceptual equivalence between different forms of the same concept and specifying not to over-group merely related terms; in addition, the prompt allows for groups with two or more terms when clear equivalence exists, or with a single term when no equivalent is found. The prompt also provides

[30] Without the frequencies that would only add tokens but would not provide useful context for this step.

an example of the expected output: a structured JSON dictionary in which each concept group contains either a single term, a pair or multiple equivalent terms.

Importantly, due to the LLM's token limit, the terms are sent in batches and hence a script merges the output of each step at the very end, so that overlapping groups can be resolved. Merging occurs only when there is an exact match between the entire term, not based on shared words, to prevent accidental merges of unrelated terms. An additional filtering step ensures only terms present in the original file survive, preventing LLM-introduced variants from entering the final dataset.

Because conceptual equivalence is inherently semantic, a human validation step is introduced. However, not all groupings are presented to the terminologist, but only potentially risky ones that are more likely to be false positives: these include groups of three or more terms (e.g. "debt fund", "fund of fund" and "private debt fund") and two-term groups with no clear acronym and low lexical overlap, i.e. where the surface form of the terms differ substantially (e.g. "early warning system" and "trigger system" or "derivative financial instrument" and "derivative instrument"). Single-term concepts and clear acronym-full form pairs are auto-accepted, as to lessen the cognitive workload (e.g., "CAPM" and "Capital Asset Pricing Model"). The terminologist can then interactively (i) accept the grouping as is, (ii) edit the grouping in the console, (iii) remove one or more terms completely, (iv) split individual terms from the group so that they become single terms or (v) degroup the entire group into single terms.

Once this step has been completed, the resulting list of validated concepts is treated as final and is turned into a table that mirrors the structure of the original TB: 3ach row represents a single term occurrence, while the columns correspond to the metadata fields defined in the TB schema. At this stage, some fields are populated deterministically, either by assigning fixed values or through explicit lookup in the entries file. Table 3.10 provides an overview of the metadata fields and their values[31].

[31] Although the TB supports additional fields, these were deliberately excluded from the automation pipeline because they are intended to be populated manually and directly on the platform after import by the terminologists.

Field	Value
Concept	Each term is assigned a concept identifier, with all terms belonging to the same concept sharing the same concept ID
LineNr	Each term is assigned a sequential number independent of concept groupings
Subject area	Fixed value: “Business and Legal”
Field of Business and Legal	Fixed value: “Finance”
Subfield of Business and Legal	Fixed value: “Investment”
Superordinate concept	Left blank for augmentation
Subordinate concept	Left blank for augmentation
Associated concept	Left blank for augmentation
Opposite concept	Left blank for augmentation
Definition	Left blank for augmentation
Source of definition	Left blank for augmentation
Term	Term (retrieved from the entries file)
Source of term	Link to the source file (retrieved from the entries file)
Context	Context sentence (retrieved from the entries file)
Source of context	Link to the source file (retrieved from the entries file)
Type of designation	Left blank for augmentation

Table 3.10. Metadata fields and values.

The table is exported to Excel, as this format best supports manual review and is compatible with the platform for later import. At the end of this phase, the automation pipeline produces a complete and validated set of new concepts.

3.3.5. Phase 4: Augmentation

The augmentation phase enriches the TB with two missing metadata: the definition (and its source) and the type of designation. Because many pre-existing entries also lacked these metadata, augmentation was applied to the entire TB. Therefore, as a preliminary step, the new and existing entries were merged into a single Excel table.

The first metadata field addressed was “Type of designation”, which in our TB describes the formal shape a term can take, not grammatical or syntactic categories. In particular, the two values considered

were “acronym” and “abbreviation”. Each term was examined individually from the Excel table[32] and a set of pattern-based rules was applied: acronyms were identified as character sequences without spaces and consisting predominantly of uppercase letters (optionally containing common separators such as hyphens); while abbreviations were identified primarily based on the presence of a trailing full stop and short overall length[33]. Full forms, which do not match either pattern, were left untagged as they are considered the default case. The updated table was then written back to Excel. This step was implemented through explicit rules because token limitations would have made an LLM-driven approach unnecessarily time-consuming and because the task is inherently well-suited to rule-based methods that do not require semantic reasoning.

The second augmentation step concerned definitions. Because privacy restrictions prevented automated lookups in external web sources (more in Section 3.5), a set of authoritative online glossaries, from institutional bodies and established companies in the field, was manually collected and stored offline[34]. Since definitions are placed at the entry level, meaning that they refer to all term variants of a concept, the script processed the Excel file grouping rows by concept ID and selecting one full-form term per concept as the target row to store the definition. Full forms are preferred as they offer more reliable matches with the glossaries compared to acronyms and abbreviations. If the selected row already contains a definition, the concept is skipped so that existing content is retained. For concepts without definitions, the script searches the external glossaries[35] using light normalization so that minor punctuation or casing differences do not prevent a match. Finally, the chosen full-form row is populated with both the retrieved definition and an HTML-formatted link to the source reference.

Augmentation of grammatical or morphological categories was not necessary for this use case. The vast majority of entries are nouns or noun phrases[36], therefore POS-tagging would not provide meaningful terminological value to us. Moreover, further linguistic annotation such as gender classification, is not applicable to the English language. However, both metadata types could be incorporated for domains where additional grammatical distinctions are relevant or languages with grammatical gender.

[32] Pre-existing categorizations were deleted as inconsistent.

[33] These rules were intentionally restrictive to avoid over-classification.

[34] Local copies of external glossaries were created exclusively for offline processing due to privacy restrictions; sources are credited both in the local copies and in the TB.

[35] A priority order is defined so that glossaries from public institutions are consulted first; if no institutional definition is found, the script falls back to private companies’ glossaries.

[36] The few exceptions will be subject to manual review to determine whether they should remain in the TB at all.

3.3.6. Phase 5: Clustering and relation identification

The final major phase consisted in identifying relations among concepts, an important component of a concept-oriented TB, as mentioned in Section 3.2.2.3, particularly in a platform such as Quickterm that supports knowledge-graph visualization. Since this step is most effective when the entire TB is considered at once, but token limitations of the self-hosted LLMs prevented the upload of the full set, it was necessary to restrict the number of concepts to be processed simultaneously.

Therefore, a preliminary step was designed to roughly group the entries into semantic groups. Due to restrictions in accessing external resources, embedding-based methods were excluded and an approach leveraging TF-IDF and k-means was adopted (Hidayat et al., 2025; Marappan & Vignesh, 2024). Each entry – a string made up of the term, the definition and the context sentence (if available) – was encoded as a TF-IDF vector. TF-IDF is a statistical measure that assigns weighs to words according to how important they are for a given entry relative to the rest of the corpus; these weights then form a numerical vector that helps capture similarities in vocabulary and contextual elements across entries (Marappan & Vignesh, 2024). The resulting vectors were clustered using k-means, an algorithm that forms a predefined number of clusters (k = 30, in this case) by placing each entry into the group it is most similar to (Hidayat et al., 2025). A fundamental limitation of this approach is its reliance on syntactic similarity and lexical patterns rather than semantic meaning; consequently, entries sharing contextual relationships may be assigned to different clusters (Hidayat et al., 2025).

Once this preliminary clustering step was completed, the second step implemented the actual LLM-based identification of conceptual relations among entries belonging to the same cluster, ensuring that the token load remained computationally manageable. Each cluster of entries was sent to the model one by one with a prompt instructing it to perform the task in two distinct stages (see Table 3.22 in Appendix B): first, the model determined whether any conceptual relation existed between two entries, marking the pair as related when appropriate; second, for those pairs, the model assigned one of the predefined relationship types: superordinate (e.g., “securities” as the hypernym of “transferable securities”), subordinate (e.g., “credit default swap” as an hyponym of “credit derivative”), associative (e.g., “asset” as the object of an “asset allocation” process) and opposite (e.g., “bid price” and “ask price”). An additional category – equivalent – was included to catch those terms that, due to omissions or oversight, should have already be included in the same entry in the original TB, such as synonyms, spelling variants, acronyms (e.g., “credit risk spread” as the more explicit formulation of “credit spread”). For hierarchical relationships, the model was also required to specify the direction between the entries. Moreover, for every pair it marked as related, the LLM had to provide a brief justification for its choice to support the terminologist during validation. Table 3.11 provides an example of a subordinate entry, with direction and explanation.

CONCEPT 1: corporate bond (ID: 39043) CONCEPT 2: bond (ID: 60153) TYPE: subordinate DIRECTION: bond→ corporate bond EXPLANATION: A corporate bond is a type of bond, making bond a more general concept than corporate bond.

Table 3.11. Example of conceptual relation.

Following the LLM's response, entry pairs marked as related were presented to the human expert to be validated on the console immediately after generation; this step was performed cluster by cluster to avoid the cognitive overload that would result from merging all clusters together. The validation interface mirrored the one from previous validation tasks, giving the terminologist the options to (i) accept the suggested relationship, (ii) edit any field, (iii) discard the suggestion, (iv) skip a pair or (v) flag it for later review. The results were written to three separate files (`yes`, `no`, `check`). The relations were not automatically added to the TB, rather they were kept as suggestions for the terminologist to eventually integrate into the platform via cross-referencing.

3.4. Results

Given the exploratory and expert-driven nature of the task, results are reported using a mixed approach: quantitative results summarize the output at each stage of the pipeline and the degree of divergence between LLM proposal and expert validation, while qualitative analysis offers a report on recurring patterns observed during LLM-driven processes as well as validation decisions.

3.4.1. Quantitative overview

The original TB contained 1,827 terms labelled under the general Finance domain. After the subdomain identification step, which aimed to isolate investment-specific terms, 1,444 terms were retained (79% of the original dataset). The extraction step produced 385 candidate terms, of which 9 were discarded during deduplication and 177 were accepted (45.9% of the extracted candidates). With the addition of the 177 validated terms, the Investment TB reached a total of 1,621 terms, representing a 12.3% increase. During concept consolidation, 23 cases of equivalent term variants were identified among the validated terms and grouped under the same concept identifier, resulting in 154 concept-level entries. Table 3.12 summarizes term counts at each step.

Step	Counts
Original TB (Finance domain)	1,827 terms
After subdomain identification (Investment subdomain)	1,444 terms
Raw extraction candidates	385 terms
Validated terms	177 terms
Final Investment TB size	**1,621 terms**

Table 3.12. Extraction pipeline in numbers.

Metadata augmentation was applied to the merged TB, affecting both newly extracted and already existing entries. In total, 51 terms were marked as acronyms, while no abbreviations were found. Definition augmentation resulted in the addition of 188 definitions sourced from the external glossaries.

During the relation identification phase, the 31 processed clusters[37] showed moderate variation: the median size was 30.0 entries (Q1: 21.5, Q3: 46, IQR = 24.5), with sizes ranging from 5 to 112 entries, while the mean (39.03) exceeded the median due to the presence of a small number of very large clusters. Across all clusters, the LLM-driven relation identification phase proposed 278 candidate conceptual relations. Following human validation, 151 relations were confirmed, corresponding to a validation rate of 54.3 %. Candidate and validated relations are broken down by relation type in Table 3.13.

Relation type	Candidate relations	Validated relations	Validation share
Superordinate	20	16	80.0%
Subordinate	63	48	76.2%
Associative	174	76	43.7%
Opposed	6	6	100.0%
Equivalent	15	5	33.3%
Total	**278**	**151**	**54.3%**

Table 3.13. Candidate and validated relations by type.

3.4.2. Qualitative analysis

During term extraction, the LLM displays a consistent ability to identify domain-relevant multi-word concepts, alongside a smaller but recurrent set of items that fall outside the intended scope. Across several documents, the model regularly proposes concrete investment-related entries (e.g. “M&A

[37] Of the 30 originally formed clusters, one was too numerous for the LLM and therefore had to be manually split into three equal subclusters, while another could not be processed due to repeated parsing failures of the LLM’s response.

investment", "Capital Asset Pricing Model", "net insurance revenue", "tranching"), most of which were accepted during validation. At the same time, the output also includes generic or governance-oriented vocabulary (e.g., "portfolio management agreement", "business case", "commodity", "climate change mitigation"), which frequently occur in investment documents but do not strictly belong to the domain and were therefore mostly discarded during validation.

The casing and formatting of the terms as well as the quality of the context sentences cannot be assessed directly at extraction as post-processing is immediately applied and the resulting candidate list already reflects their processed form. While these rules successfully standardize the majority of entries according to the TB's standards, occasional discrepancies remain, such as lowercased acronyms and full forms (e.g., "capital asset pricing model" lowercased when it is a proper noun and should be uppercased), singularized terms that are however used only or mainly in their plural form (e.g., "Mergers and Acquisitions" being singularized to "merger and acquisition" when the correct form is plural), or incomplete or suboptimal context sentences. When the placeholder NO_VALID_CONTEXT appears, it can be replaced during validation with a sentence drawn directly from the source document; this typically occurs when the extracted term has been singularized while the original sentence contains an irregular plural form, preventing substring matching, or when the sentence appears in a table and was excluded during preprocessing (e.g., "contingent liability" having NO_VALID_CONTEXT because it appears only in the plural form in the text). More marginal cases also occur in which the LLM does not faithfully reproduce the term found in the source text, resulting in zero frequency counts and missing contexts (e.g., from the sentence "Investments in owner-occupied property (not for general investment purposes)", the model proposes the term "owner-occupied property investment").

LLM-driven concept consolidation performs quite well, identifying most full form–short form pairs (e.g., "Asset-Liability Management" and "ALM" or "Generally Accepted Accounting Principle" and "GAAP") but also some equivalent terms (e.g., "liquidity crisis planning" and "liquidity crisis plan").

Finally, during the relationships extraction phase, the LLM performs well when identifying clear hierarchical or equivalence relations grounded in established terminology: straightforward subordinate or superordinate relations are frequently proposed correctly (e.g., "corporate bond" being a subtype of "bond", "preference share" a subtype of share" or "credit default swap" as a type of "credit derivative"); similarly, true or near-synonyms and antonyms are often detected accurately (e.g., "initial capital" and "start-up capital", "risk-return profile" and "risk/return profile" or "divestment" as the opposite of "investment"). On the other hand, the model also shows a marked tendency to propose associative relationships that are conceptually and terminologically weak, as they

mostly refer to situational connections (e.g., “outside capital” and “corporate bond” were deemed associative because “outside capital can be raised through corporate bond” or “M&A investment” and “joint venture” were also deemed associative because “both are strategic investment decisions”, but both cases are overly generic connections). In some cases, the LLM correctly identifies that two concepts are related, but assigns an inappropriate relationship type (e.g., “asset” and “asset erosion” are initially proposed as subordinate but then edited to associative reflecting that “asset erosion” is a process involving an “asset” and not a subclass).

3.5. Discussion

This exploratory study demonstrates that LLMs can support multiple stages of terminology management when embedded in a supervised hybrid pipeline. Across term and context extraction, concept consolidation, metadata augmentation and relation identification, LLMs effectively acted as candidate generators, provided that their outputs were constrained through structured prompting and deterministic pre- and post-processing procedures. These architectural choices proved essential for ensuring terminological consistency and mitigating the variability typical of generative AI models. Another central design principle of the pipeline was the explicit integration of human experts: rather than aiming for full automation, the system was intentionally developed as a human- or expert-in-the-loop workflow in which LLMs propose candidates while terminologists retain control over validation. In general, quantitative results show that this configuration yields relatively high precision across stages, while the qualitative analysis confirms that expert validation remains necessary to resolve ambiguities, correct over-generalizations and enforcing domain boundaries.

More specifically, a deliberate choice was made to prioritize precision over recall during term (and context) extraction, motivated by the existence of an initial TB and the high effort of manual validation. While this choice resulted in a lower number of extracted candidate terms compared to an extraction strategy focused on recall, it substantially reduced noise and therefore validation effort. This design decision also mitigated a systematic tendency of the LLM to propose overly general or domain-adjacent candidates; though when cases still appeared human validation played a critical role in identifying and discarding them, ensuring that the retained entries were aligned with the investment-specific scope of the extraction phase. Overall, the results indicate that this precision-oriented approach is appropriate in corporate terminology settings, where accuracy and consistency are more important than coverage; by contrast, domains that lack an initial terminological resource would likely benefit from a high-recall strategy followed by more extensive validation.

Automated metadata augmentation further illustrated the complementary strengths of rule-based and LLM-based approaches: deterministic type of designation classification and glossary-based

definition retrieval provided quick and transparent results, making them particularly suitable for a corporate setting. This is particularly true for definition augmentation, as the experimental nature of LLM-generated definitions and the lack of reliable source attribution currently limits their applicability in contexts where traceability is required.

The results further indicate that LLM-assisted relation identification is feasible but particularly sensitive: while the models were able to propose a substantial number of plausible conceptual relationships, validation and subsequent analysis revealed a systematic tendency toward loose associative relations. This pattern suggests that, without strong contextual information and more explicit prompting, LLMs may favor broad semantic relations. Human validation therefore played a decisive role in filtering, editing or discarding such candidates.

Several limitations of this study must be acknowledged. First, the reliance on self-hosted LLMs with strict token limits directly influenced the architectural design of the pipeline: batching strategies and clustering were adopted to remain within these limits, but they may have limited the semantic context available to the model during term extraction and relation identification.

Second, restrictions about downloading and using external models and services limited two critical steps of the pipeline: on one hand, definition augmentation would have greatly benefitted by third-party APIs to source definitions from authoritative sources (such as the Merriam-Webster Dictionary API[38] or the Wikipedia API[39]); on the other hand, embedding-based methods using representation models (such as Sentence Transformers from Hugging Face[40]) would have enabled the creation of more semantically coherent clusters, thereby affecting the quality of the subsequent identification step.

Third, the pipeline was applied to a single domain and a single language. This means that no claims can be made regarding the generalizability of the results to other subject fields and languages. The overall pipeline design is in principle domain-agnostic, as domain-specific instructions are relegated to prompts and not to the deterministic components; however, to support other languages, a few components, such as singularization, token counting and the prompts themselves, may need adjustments.

Future research should explore several directions to address the limitations identified in this study. A key extension would be the integration of embedding-based clustering methods, which could improve semantic coherence and subsequently relation identification. Another important direction is multilingual extension: applying the pipeline to corpora in different languages would allow for

[38] https://dictionaryapi.com/
[39] https://en.wikipedia.org/api/rest_v1/
[40] https://huggingface.co/sentence-transformers

systematic evaluation of its adaptability; such experiments would also enable comparative analysis across languages and the enhancement of the TB with equivalents in other languages. Further studies could also examine strategies to mitigate the observed LLM biases toward generic terms and lose associative relations, for example through stricter or few-shot prompts or multiple validation mechanisms.

CONCLUSIONS

This thesis explored how automation and AI, in particular LLMs, can be integrated into a corporate Language Service (LS) department by adopting hybrid approaches that integrate deterministic processes, probabilistic models and human expertise. This approach was applied to two case studies, representative of two workflows central to LS: the first project concerned the development of a text anonymization tool designed to identify and mask Personally Identifiable Information (PII) in multilingual SDLXLIFF documents while ensuring compliance with internal security requirements; the second project involved the proposal of a semi-automated workflow for expanding and enriching the internal Termbase (TB), covering tasks from term and context extraction, concept consolidation, metadata augmentation and relation identification.

Working within Munich Re's organizational, regulatory and infrastructural constrains, this work has shown that hybrid architectures mitigate the limitations of purely rule-based or purely data-driven systems. Rule-based components offer transparency, controllability and compliance with strict privacy requirements, but they struggle with linguistic variability and domain-awareness. Conversely, probabilistic models, particularly LLMs, have good contextual and semantic awareness as well as generalization abilities, but introduce unpredictability and therefore require safeguards. Both case studies implement a similar hybrid architecture: (i) a deterministic preprocessing step to standardize inputs, (ii) a probabilistic LLM-driven core to leverage semantic knowledge and context-awareness, (iii) a deterministic postprocessing step to normalize outputs and enforce constraints, and (iv) a final human validation process. By integrating all these components, the proposed solutions achieve both flexibility and reliability and enable the integration of such AI tools into existing workflows and platforms.

Although the two case studies differ in scope and implementation, they further illustrate how operational tools and pipelines can be designed by applying the theoretical framework of augmentation and automation criteria. The principle of augmentation informed the decision to use AI as a support mechanism rather than a substitute for human judgment; this is especially evident in the TB case study, where expert validation is explicitly integrated into most steps of the pipeline, but it also applies to the anonymization tool, as human oversight remains essential to catch undetected entities. Automation criteria further guided design choices: interfaces were kept simple and user-friendly, situational awareness was preserved as much as possible through user messages, and full automation was avoided due to the complexity of the tasks at hand and the high-stake environment.

The text anonymization use case illustrates that the hybrid AI approach can produce a flexible and adaptable tool while meeting internal security requirements. In particular, the LLM demonstrated strong multilingual and cross-domain adaptability, as well as the ability to recognize diverse

Personally Identifiable Information (PII) patterns, while the deterministic components further improved performance by compensating for typical LLM weaknesses, such as fragmented or inconsistently formatted PII. In the TB expansion pipeline, the LLM proved effective as candidate generator across multiple stages, from term and context extraction to concept consolidation and relation identification, provided that its outputs were constrained through structured prompting and deterministic components. Furthermore, expert validation remained essential to enforce domain adherence and filter overly generic relation associations. Together, these findings indicate that hybrid architectures can reduce manual workload, increase consistency and improve scalability of processes and resources, while also highlighting the boundaries of current LLM capabilities and the importance of a human-in-the-loop workflow.

These results are shaped by several limitations. Strict privacy requirements restricted the use of external resources, such as public models or APIs. Limited computational infrastructure, most notably token and rate limits on the self-hosted LLMs, influenced several architectural decisions including prompt design, batching of a limited number of tokens and the use of semantic clustering for relation extraction; these limitations reduced the amount of context the model could process at once and introduced waiting times during execution. Such constraints limit the generalizability of the results to this specific setting and point to suggestions for future developments, some of which are currently still unfeasible in this environment, while others – such as improved deterministic matching of PII or multilingual extension of the TB pipeline – may be more easily implemented in time.

Beyond technical considerations, the thesis also reflects on the broader implications of AI adoption in the language industry. Automation requires language experts to develop interdisciplinary skill sets that combine linguistic knowledge with technological proficiency and evaluation skills. Both case studies, and the internship more broadly, highlight the importance of bringing these competencies together to design tools and workflows that are grounded in linguistic theory while leveraging emerging technologies.

In conclusion, this thesis argues that such hybrid AI architectures can deliver gains in efficiency and scalability in specific settings, while preserving reliability and accuracy; however, they do not entirely replace human expertise. Instead, they shift the focus of the role of language professionals toward the design and oversight of such systems. This collaborative model offers a sustainable and responsible alternative for integrating AI in a corporate linguistic department.

BIBLIOGRAPHY

Alla, P. B. (2025). Automation as Augmentation: Stories of Human-AI Collaboration in the Workplace. *European Modern Studies Journal*, *9*(4), 1019–1036. https://doi.org/10.59573/emsj.9(4).2025.96

Angelone, E., Massey, G., & Ehrensberger-Dow, M. (2024). Introduction: Contextualizing language industry studies. In G. Massey, M. Ehrensberger-Dow, & E. Angelone (Eds.), *Handbook of the Language Industry* (pp. 1–14). De Gruyter. https://doi.org/10.1515/9783110716047-001

Asimopoulos, D., Siniosoglou, I., Argyriou, V., Karamitsou, T., Fountoukidis, E., Goudos, S. K., Moscholios, I. D., Psannis, K. E., & Sarigiannidis, P. (2024). *Benchmarking Advanced Text Anonymisation Methods: A Comparative Study on Novel and Traditional Approaches* (arXiv:2404.14465). arXiv. https://doi.org/10.48550/arXiv.2404.14465

Autor, D. (2015). Why Are There Still So Many Jobs? The History and Future of Workplace Automation. *ResearchGate*. https://doi.org/10.1257/jep.29.3.3

Bachmann, M. (2025). *RapidFuzz* (Version 3.14.3) [Python]. https://github.com/rapidfuzz/RapidFuzz (Original work published 2020)

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., Do, Q. V., Xu, Y., & Fung, P. (2023). *A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity* (arXiv:2302.04023). arXiv. https://doi.org/10.48550/arXiv.2302.04023

Behnel, S. (2025). *lxml—Processing XML and HTML with Python* (Version 6.0.0) [Python]. https://lxml.de/index.html

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). *Enriching Word Vectors with Subword Information* (arXiv:1607.04606). arXiv. https://doi.org/10.48550/arXiv.1607.04606

Brandt, A. (2024). Chapter 7 Translation and localization project and process managers. In G. Massey, M. Ehrensberger-Dow, & E. Angelone (Eds.), *Handbook of the Language Industry* (pp. 143–178). De Gruyter. https://doi.org/10.1515/9783110716047-008

Breton, J., Billami, M. M., Chevalier, M., Nguyen, H. T., Satoh, K., Trojahn, C., & Zin, M. M. (2025). Leveraging LLMs for legal terms extraction with limited annotated data. *Artificial Intelligence and Law*. https://doi.org/10.1007/s10506-025-09448-8

Briva-Iglesias, V., & O'Brien, S. (2022). The Language Engineer: A Transversal, Emerging Role for the Automation Age. *Quaderns de Filologia - Estudis Lingüístics*, *27*, 17–48. https://doi.org/10.7203/qf.0.24622

Carmo, F. D., & Koponen, M. (2024). Chapter 9 Revisers and post-editors: The guardians of quality. In G. Massey, M. Ehrensberger-Dow, & E. Angelone (Eds.), *Handbook of the Language Industry* (pp. 203–224). De Gruyter. https://doi.org/10.1515/9783110716047-010

Chun, Y., Kim, M., Kim, D., Park, C., & Lim, H. (2025). *Enhancing Automatic Term Extraction with Large Language Models via Syntactic Retrieval* (arXiv:2506.21222). arXiv. https://doi.org/10.48550/arXiv.2506.21222

Conrado, M., Pardo, T., & Rezende, S. (2013). A Machine Learning Approach to Automatic Term Extraction using a Rich Feature Set. In A. Louis, R. Socher, J. Hockenmaier, & E. K. Ringger (Eds.), *Proceedings of the 2013 NAACL HLT Student Research Workshop* (pp. 16–23). Association for Computational Linguistics. https://aclanthology.org/N13-2003/

Cortesi, D., Bajo, G., Caban, W., & McMillan, G. (2025). *PyInstaller* (Version 6.17.0) [Computer software]. https://pyinstaller.org/en/stable/index.html

Csutoras, B. (2025, September 27). The Em Dash Dilemma: How a Punctuation Mark Became AI's Stubborn Signature. *Medium*. https://medium.com/@brentcsutoras/the-em-dash-dilemma-how-a-punctuation-mark-became-ais-stubborn-signature-684fbcc9f559

Declercq, C., & Egdom, G. van. (2023). No more buying cats in a bag? Literary Translation in the age of language automation: Editorial. *Tradumàtica Tecnologies de La Traducció*, (21), 49–62. https://doi.org/10.5565/rev/tradumatica.407

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (arXiv:1810.04805). arXiv. https://doi.org/10.48550/arXiv.1810.04805

Dhar, A. (2025, December 11). *PII Detection In Unstructured Text: Why Regex Fails (And What Works)*. https://www.protecto.ai/blog/why-regex-fails-pii-detection-in-unstructured-text/

Doherty, S. (2016). The Impact of Translation Technologies on the Process and Product of Translation. *International Journal of Communication*, *10*(0), 23.

ELIS. (2025). *EUROPEAN LANGUAGE INDUSTRY SURVEY 2025*.

Engel, C., Ebel, P., & Leimeister, J. M. (2022). Cognitive automation. *Electronic Markets*, *32*(1), 339–350. https://doi.org/10.1007/s12525-021-00519-7

Faes, F., & Massey, G. (2024). Chapter 1 Charting the language industry: Interview with an industry observer. In G. Massey, M. Ehrensberger-Dow, & E. Angelone (Eds.), *Handbook of the Language Industry* (pp. 17–32). De Gruyter. https://doi.org/10.1515/9783110716047-002

Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms:. The C-value/NC-value method. *International Journal on Digital Libraries*, *3*(2), 115–130. https://doi.org/10.1007/s007999900023

GDPR. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. 1–88.

Granell, X., & Chaume, F. (2023). Audiovisual translation, translators, and technology: From automation pipe dream to human–machine convergence. *Linguistica Antverpiensia, New Series – Themes in Translation Studies*, *22*. https://doi.org/10.52034/lans-tts.v22i.776

Hagos, D. H., Battle, R., & Rawat, D. B. (2024). Recent Advances in Generative AI and Large Language Models: Current Status, Challenges, and Perspectives. *IEEE Transactions on Artificial Intelligence*, *5*(12), 5873–5893. https://doi.org/10.1109/TAI.2024.3444742

Hidayat, M. Y., Yaqin, M. A., & Abidin, Z. (2025). Semantic-Enhanced News Clustering Using TF-IDF and WordNet with K-Means. *Journal of Information Systems and Informatics*, *7*(4), 3924–3951. https://doi.org/10.63158/journalisi.v7i4.1360

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*.

Huang, Z., Xu, W., & Yu, K. (2015). *Bidirectional LSTM-CRF Models for Sequence Tagging* (arXiv:1508.01991). arXiv. https://doi.org/10.48550/arXiv.1508.01991

Ide, Y., Nohejl, A., Tanner, J., Yanaka, H., Lindsay, C., & Watanabe, T. (2026). *Towards Automated Lexicography: Generating and Evaluating Definitions for Learner's Dictionaries* (arXiv:2601.01842). arXiv. https://doi.org/10.48550/arXiv.2601.01842

International Organization for Standardization. (2019). *ISO 1087:2019*. https://www.iso.org/standard/62330.html

Judea, A., Schütze, H., & Bruegmann, S. (2014). Unsupervised Training Set Generation for Automatic Acquisition of Technical Terminology in Patents. In J. Tsujii & J. Hajic (Eds.), *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 290–300). Dublin City University and Association for Computational Linguistics. https://aclanthology.org/C14-1029/

Jurafsky, D., & Martin, J. H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models* (3rd ed.). https://web.stanford.edu/~jurafsky/slp3

Kageura, K., & Marshman, E. (2019). Terminology extraction and management. In M. O'Hagan (Ed.), *The Routledge Handbook of Translation and Technology* (1st ed., pp. 61–77). Routledge. https://doi.org/10.4324/9781315311258-4

Kaleidoscope. (2025). *Quickterm* (Version 6.7.) [Computer software]. https://kaleidoscope.at/en/platform/quickterm/

Kappus, M. (2024). Chapter 6 Language technology developers. In G. Massey, M. Ehrensberger-Dow, & E. Angelone (Eds.), *Handbook of the Language Industry* (pp. 121–142). De Gruyter. https://doi.org/10.1515/9783110716047-007

Kocaman, V., Haq, H. U., & Talby, D. (2023). *Beyond Accuracy: Automated De-Identification of Large Real-World Clinical Text Datasets* (arXiv:2312.08495). arXiv. https://doi.org/10.48550/arXiv.2312.08495

Kucza, M., Niehues, J., Zenkel, T., Waibel, A., & Stüker, S. (2018). Term Extraction via Neural Sequence Labeling a Comparative Evaluation of Strategies Using Recurrent Neural Networks. *Interspeech 2018*, 2072–2076. https://doi.org/10.21437/Interspeech.2018-2017

Kumar, S., & Bonnefoy-Claudet, B. (2025). *python-dotenv: Read key-value pairs from a .env file and set them as environment variables* (Version 1.2.1) [Python]. https://pypi.org/project/python-dotenv/

Lafferty, J., McCallum, A., & Pereira, F. (2001, June 28). *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. International Conference on Machine Learning. https://www.semanticscholar.org/paper/Conditional-Random-Fields%3A-Probabilistic-Models-for-Lafferty-McCallum/f4ba954b0412773d047dc41231c733de0c1f4926

Marappan, S., & Vignesh, S. (2024). *Text Clustering for Topic Identification: A TF-IDF and K-Means Approach Applied to the 20 Newsgroups Dataset*. EasyChair Preprint 13917. https://easychair.org/publications/preprint/C6Ft?utm_source=chatgpt.com

Massey, G., Ehrensberger-Dow, M., & Angelone, E. (2024). *Handbook of the Language Industry: Contexts, Resources and Profiles* (1st ed., Vol. 20). De Gruyter, Inc.

Miličević Petrović, M., Bernardini, S., Ferraresi, A., Aragrande, G., & Barrón-Cedeño, A. (2021). *Language data and project specialist: A new modular profile for graduates in language-related disciplines*. https://doi.org/10.5281/zenodo.5030929

Mishra, K., Pagare, H., & Sharma, K. (2025). A hybrid rule-based NLP and machine learning approach for PII detection and anonymization in financial documents. *Scientific Reports*, *15*(1), 22729. https://doi.org/10.1038/s41598-025-04971-9

Mohamed, K., Valentini, C., Ralli, N., Barros, S., Löckinger, G., Vezzani, F., Salgado, A. C., Zhang, Z., Mahr, S., Carvalho, S., Fleischmann, K., & Costa, R. (2025). *Terminology Management Meets AI*. UniorPress. http://hdl.handle.net/10362/190326

Moorkens, J., Castilho, S., Gaspari, F., Toral, A., & Popović, M. (2024). Proposal for a Triple Bottom Line for Translation Automation and Sustainability: An Editorial Position Paper. *The Journal of Specialised Translation*, (41), 2–25. https://doi.org/10.26034/cm.jostrans.2024.4706

Moorkens, J., & Guerberof Arenas, A. (2024). Chapter 4 Artificial intelligence, automation and the language industry. In G. Massey, M. Ehrensberger-Dow, & E. Angelone (Eds.), *Handbook of the Language Industry* (pp. 71–98). De Gruyter. https://doi.org/10.1515/9783110716047-005

Moorkens, J., Way, A., & Lankford, S. (2024). *Sociotechnical Effects of Machine Translation*. https://doi.org/10.4324/9781003381280

Mozes, M., & Kleinberg, B. (2021). *No Intruder, no Validity: Evaluation Criteria for Privacy-Preserving Text Anonymization* (arXiv:2103.09263). arXiv. https://doi.org/10.48550/arXiv.2103.09263

Munich Re. (2025). *New multi-year strategy Ambition 2030: Munich Re focuses on sustained profit growth and high profit participation for shareholders*. https://www.munichre.com/en/company/media-relations/media-information-and-corporate-news/media-information/2025/media-release-2025-12-11.html

OpenAI. (2025). *How your data is used to improve model performance*. OpenAI Help Center. https://help.openai.com/en/articles/5722486-how-your-data-is-used-to-improve-model-performance

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V.,

Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2024). *GPT-4 Technical Report* (arXiv:2303.08774). arXiv. https://doi.org/10.48550/arXiv.2303.08774

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *30*(3), 286–297. https://doi.org/10.1109/3468.844354

Paschalidis, A. I. (2025). *AI and the great linguistic flattening | UNESCO*. https://www.unesco.org/en/articles/ai-and-great-linguistic-flattening

Pissarra, D., Curioso, I., Alveira, J., Pereira, D., Ribeiro, B., Souper, T., Gomes, V., Carreiro, A. V., & Rolla, V. (2024). *Unlocking the Potential of Large Language Models for Clinical Text Anonymization: A Comparative Study* (arXiv:2406.00062). arXiv. https://doi.org/10.48550/arXiv.2406.00062

Pym, A., & Torres-Simón, E. (2021). Is automation changing the translation profession? *International Journal of the Sociology of Language*, *2021*(270), 39–57. https://doi.org/10.1515/ijsl-2020-0015

Radford, A., & Narasimhan, K. (2018). *Improving Language Understanding by Generative Pre-Training*. https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford-Narasimhan/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035

Rajgarhia, H., Gupta, S., Shaik, A., Kumar, G. P., Santhoshraj, Y., Nishitha, S. N. T., & Mukherji, A. (2025). *An Evaluation Study of Hybrid Methods for Multilingual PII Detection* (arXiv:2510.07551). arXiv. https://doi.org/10.48550/arXiv.2510.07551

Ramshaw, L., & Marcus, M. (1995). Text Chunking using Transformation-Based Learning. *Third Workshop on Very Large Corpora*. https://aclanthology.org/W95-0107/

Reitz, K. (2025). *Requests: HTTP for Humans™* (Version 2.32.5) [Computer software]. https://requests.readthedocs.io/en/latest/

Rivas Ginel, M. I., & Moorkens, J. (2025). Translators' trust and distrust in the times of GenAI. *Translation Studies*, *18*(2), 283–299. https://doi.org/10.1080/14781700.2025.2507594

Russell, S. J., & Norvig, P. (with Chang, M., Devlin, J., Dragan, A., Forsyth, D., Goodfellow, I., Malik, J., Mansinghka, V., Pearl, J., & Wooldridge, M. J.). (2022). *Artificial intelligence: A modern approach* (Fourth edition, global edition). Pearson.

RWS. (2024). *Inline tags*. https://docs.rws.com/en-US/trados-studio-2024-1145319/inline-tags-532829

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2025). *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications* (arXiv:2402.07927). arXiv. https://doi.org/10.48550/arXiv.2402.07927

Schimansky, T. (2024). *CustomTkinter* (Version 5.2.2) [Computer software]. https://customtkinter.tomschimansky.com/documentation/

Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., Li, Y., Gupta, A., Han, H., Schulhoff, S., Dulepet, P. S., Vidyadhara, S., Ki, D., Agrawal, S., Pham, C., Kroiz, G., Li, F., Tao, H., Srivastava, A., … Resnik, P. (2025). *The Prompt Report: A Systematic Survey of Prompt Engineering Techniques* (arXiv:2406.06608). arXiv. https://doi.org/10.48550/arXiv.2406.06608

Shah, W. (2022). *Hybrid AI Models: Combining Symbolic Reasoning and Deep Learning for Enhanced Decision-Making*. https://doi.org/https://doi.org/10.13140/RG.2.2.11493.61920

Simonov, K. (2020). *PyYAML* (Version 5.3) [Computer software]. https://pyyaml.org/wiki/PyYAML

Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*. CoNLL 2002. https://aclanthology.org/W02-2024/

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). *LLaMA: Open and Efficient Foundation Language Models* (arXiv:2302.13971). arXiv. https://doi.org/10.48550/arXiv.2302.13971

Tran, H. T. H., Martinc, M., Caporusso, J., Doucet, A., & Pollak, S. (2023). *The Recent Advances in Automatic Term Extraction: A survey* (arXiv:2301.06767). arXiv. https://doi.org/10.48550/arXiv.2301.06767

van der Meer, A.-M. (2024). Chapter 2 Evolution of the language industry. In G. Massey, M. Ehrensberger-Dow, & E. Angelone (Eds.), *Handbook of the Language Industry* (pp. 33–48). De Gruyter. https://doi.org/10.1515/9783110716047-003

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Vieira, L. N. (2020). Automation anxiety and translators. *Translation Studies*, *13*(1), 1–21. https://doi.org/10.1080/14781700.2018.1543613

Warburton, K. (2024). Chapter 8 Terminology managers. In G. Massey, M. Ehrensberger-Dow, & E. Angelone (Eds.), *Handbook of the Language Industry* (pp. 179–202). De Gruyter. https://doi.org/10.1515/9783110716047-009

Watson, A., Meyer, Y., Van Segbroeck, M., Grossman, M., Torbey, S., Mlocek, P., & Greco, J. (2024). *A synthetic dataset for training language models to label and detect PII in domain specific formats*. https://huggingface.co/datasets/gretelai/synthetic_pii_finance_multilingual

Wissik, T. (2025). Impact of automatic term extraction on terminology work: A qualitative interview study in institutional settings. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *31*(1), 110–135. https://doi.org/10.1075/term.00085.wis

Xu, J., Bandyopadhyay, S., Pawar, N., & Touretzky, D. S. (2024). *Word Embedding Demo*. https://www.cs.cmu.edu/~dst/WordEmbeddingDemo/index.html

Yuan, Y., Gao, J., & Zhang, Y. (2017). Supervised learning for robust term extraction. *2017 International Conference on Asian Language Processing (IALP)*, 302–305. https://doi.org/10.1109/IALP.2017.8300603

Zhang, C., Wang, H., Liu, Y., Wu, D., Liao, Y., & Wang, B. (2008). *Automatic keyword extraction from documents using conditional random fields*. *4*, 1169–1180.

APPENDIX

A. CHAPTER 2: TEXT ANONYMIZATION

Table 2.14. CRF per entity scores.

Tag	Precision	Recall	F1 score
B-BANK_ROUTING_NUMBER	0.00	0.00	0.00
B-BBAN	0.74	0.30	0.42
B-CREDIT_CARD_NUMBER	0.00	0.00	0.00
B-CREDIT_CARD_SECURITY_CODE	1.00	0.12	0.22
B-CUSTOMER_ID	0.57	0.26	0.36
B-DATE_OF_BIRTH	0.00	0.00	0.00
B-DRIVER_LICENSE_NUMBER	0.62	0.36	0.46
B-EMAIL	0.75	0.58	0.65
B-EMPLOYEE_ID	0.72	0.36	0.48
B-IBAN	0.85	0.70	0.77
B-NAME	0.78	0.57	0.66
B-PASSPORT_NUMBER	0.56	0.23	0.33
B-PHONE_NUMBER	0.28	0.12	0.16
B-SSN	0.50	0.10	0.17
B-STREET_ADDRESS	0.55	0.30	0.39
B-SWIFT_BIC_CODE	0.73	0.51	0.60
I-CREDIT_CARD_NUMBER	0.00	0.00	0.00
I-CUSTOMER_ID	0.00	0.00	0.00
I-DATE_OF_BIRTH	0.67	0.30	0.41
I-DRIVER_LICENSE_NUMBER	0.00	0.00	0.00
I-EMAIL	0.12	0.05	0.07
I-IBAN	1.00	0.10	0.18
I-NAME	0.41	0.16	0.23
I-PASSPORT_NUMBER	0.00	0.00	0.00
I-PHONE_NUMBER	0.18	0.13	0.15
I-SSN	0.00	0.00	0.00
I-STREET_ADDRESS	0.77	0.65	0.70

Table 2.15. BiLSTM-CRF per entity scores.

Tag	Precision	Recall	F1 score
B-BANK_ROUTING_NUMBER	0.98	0.67	0.80
B-BBAN	0.96	0.50	0.65
B-CREDIT_CARD_NUMBER	0.98	0.89	0.93
B-CREDIT_CARD_SECURITY_CODE	1.00	0.76	0.86
B-CUSTOMER_ID	0.91	0.56	0.70
B-DATE_OF_BIRTH	0.99	0.80	0.89
B-DRIVER_LICENSE_NUMBER	0.96	0.77	0.85
B-EMAIL	0.99	0.63	0.77
B-EMPLOYEE_ID	0.96	0.66	0.78
B-IBAN	0.94	0.78	0.86
B-NAME	0.96	0.66	0.79
B-PASSPORT_NUMBER	0.99	0.72	0.83
B-PHONE_NUMBER	0.99	0.92	0.96
B-SSN	0.94	0.78	0.85
B-STREET_ADDRESS	0.99	0.87	0.93
B-SWIFT_BIC_CODE	0.97	0.51	0.67
I-BBAN	1.00	1.00	1.00
I-CREDIT_CARD_NUMBER	0.97	0.94	0.96
I-CUSTOMER_ID	0.89	0.34	0.50
I-DATE_OF_BIRTH	0.98	0.69	0.81
I-DRIVER_LICENSE_NUMBER	0.95	0.87	0.91
I-EMAIL	0.98	0.48	0.65
I-IBAN	1.00	0.79	0.88
I-NAME	0.98	0.79	0.87
I-PASSPORT_NUMBER	0.00	0.00	0.00
I-PHONE_NUMBER	0.99	0.95	0.97
I-SSN	0.93	0.83	0.88
I-STREET_ADDRESS	0.99	0.89	0.94

Table 2.16. Overview of libraries and packages.

Library/Package	Type	Purpose
`os, time, threading, re, json, collections, sys, getpass, pathlib`	Standard library	File handling, timing, threading, RegEx, JSON operations, system utilities
`CustomTkinter` (Schimansky, 2024)	Third-party	Graphical User Interface
`RapidFuzz` (Bachmann, 2020/2025)	Third-party	Fuzzy matching during anonymization
`lxml` (Behnel, 2025)	Third-party	Parsing and writing of SDLXLIFF files
`PyYAML` (Simonov, 2020)	Third-party	Reading and writing settings files in YAML
`spaCy` (Honnibal & Montani, 2017)	Third-party	Token counting for batch creation
`PyInstaller` (Cortesi et al., 2025)	Third-party	Bundling the application into an executable
`python-dotenv` (Kumar & Bonnefoy-Claudet, 2025)	Third-party	Managing .env files with API credentials
`requests` (Reitz, 2025)	Third-party	Communication with internal API endpoints

Table 2.17. Full PII detection system message.

You are an expert at detecting Personally Identifiable Information (PII). Return only a valid JSON that conforms to the provided schema. Do not include comments or explanations.

Table 2.18. Full PII detection prompt.

Task ## Your task is to analyze the following text segments and identify all Personally Identifiable Information (PII) within. ## Instructions ## The PII types you must detect are: - BANK_ROUTING_NUMBER, - BBAN,

- CREDIT_CARD_NUMBER,
- CREDIT_CARD_SECURITY_CODE,
- DATE_OF_BIRTH (do not include general dates like appointments and similar),
- DRIVER_LICENSE_NUMBER,
- EMAIL,
- EMPLOYEE_ID,
- IBAN,
- NAME (remember that titles like ‘Dr.’ or ‘Mr.’ can help you spot a name, BUT DO NOT include the title in the list),
- PASSPORT_NUMBER,
- PHONE_NUMBER (remember that a prefix like ‘Tel.’ can help you spot a phone number, BUT DO NOT include the prefix in the list),
- SSN_NUMBER (and similar),
- STREET_ADDRESS (only if it clearly refers to a person’s residence, NOT companies, institutions or public buildings),
- SWIFT_BIC_CODE,
- VEHICLE_PLATE,
- TAX_ID_NUMBER (and similar),
- ID_CARD_NUMBER.

These are also the exact LABELS you have to use. Only include PII that clearly match the LABEL definitions.

Do not include other PII entities or labels that are not in this list.

Only include PII that refer to PEOPLE, not companies or legal entities. Do not include PII that refer to famous or known people.

Remember than number sequences can be different PIIs, try to disambiguate them according to the context.

The input language of the text segments can vary. Keep in mind that not all PIIs are consistent across languages and identify the correct variation for each language. Do not change the name of the LABEL according to the input language, but keep it in English.

Output

Return a single JSON dictionary where each key is a LABEL and each value is the exact PII string found in the text. Group together PII that correspond to the same label. Preserve the exact spelling

and structure of the PII as it appears in the text. Do not normalize or simplify. Preserve casing and punctuation as well.

Each PII should appear only once with the following exceptions, where you have to record all variations:

- partial PII: when only a part of a PII appears (for example, “123 Main Street, New York” and “123 Main Street”)
- reordered PII: when a PII is written in a different sequence (for example, “James Bond” and “Bond James” and “Bond, James”)
- misspelt PII: when a PII has a spelling mistake (for example, “James Bond” and “James Bons”)
- same date written in a different format (for example, “01 January 2000” and “01.01.2000” and “01/01/2000”).

If you encounter these exceptions in the provided segments, add each variation of the PII as different entries, even if they refer to the same entity.

Remember to not include in the PII list titles or prefixes, but only the actual PII.

Return only a valid JSON object. Do not include explanations, comments or other text.

Read each text segment carefully.

Here are the segments:

B. CHAPTER 3: TERMBASE EXPANSION

Table 3.19. Term classification prompt.

TASK

You are an expert in INVESTMENTS. Here is a list with general finance-related terms in English, some with definitions. Read each term and return two lists: one including all terms STRICTLY BELONGING to the domain of investments and one with the EXCLUDED terms.

LIST 1: INCLUDE a term ONLY if it belongs to:

- investment instruments, asset classes, financial products
- investment strategies, portfolio/allocation concepts
- investment decision-making or investment processes (only concrete, domain-specific ones)
- investment mandates, guidelines, vehicles
- investment-specific risks directly tied to investment activity

LIST 2: EXCLUDE if it is:
- general finance not specific to investments
- regulatory/legal/supervisory
- organizational/governance/procedural
- generic business terms

For borderline cases: include only if an investment professional would recognize it as an investment concept and it stands alone meaningfully in an investment glossary.

Do not add any term that is NOT in the original list, do not invent terms.

Table 3.20. Extraction prompt.

TASK
Analyze the segments appended below and extract candidate terminology belonging strictly to the INVESTMENT domain.

TERM GUIDELINES
A term is a lexical unit designating a precise concept in a specialized domain.
Concepts may have spelling variants (e.g. behavior/behaviour), format variants (e.g. email/e-mail or CEO/Chief Executive Officer) or synonyms (e.g. car/automobile).
Termhood (domain relevance) and unithood (multi-word expression cohesion) are important indicators.
Terms are typically single nouns or noun phrases.

INCLUDE a term ONLY if it belongs to:
- investment instruments, asset classes, financial products
- investment strategies, portfolio/allocation concepts
- investment decision-making or investment processes (only concrete, domain specific ones)
- investment mandates, guidelines, vehicles
- investment specific risks directly tied to investment activity

EXCLUDE if it is:
- general finance not specific to investments (e.g., risk management, operational risk, GAAP, ratings)

- regulatory/legal/supervisory (e.g., Solvency II, audit process)
- organizational/governance/procedural (e.g., committee names, management bodies)
- generic business terms (e.g., objectives, structures, processes, agreements)
For borderline cases: include only if an investment professional would recognize it as an investment concept and it stands alone meaningfully in an investment glossary.

ACRONYMS
If a term is followed by an acronym in parentheses, output:
- the full term as one entry
- the acronym as a separate entry

OUTPUT
- Return JSON with: {"terms": [{"term": "..."}]}
- Do not output explanations.
- Preserve original spelling/punctuation.
- Lowercase all terms except: acronyms, full forms of acronyms, proper nouns of laws/institutions, roman numerals.
- Singularize terms unless they are plural-only nouns
- Return all surface variants that appear in the text.
- Do not return truncated words, misspellings, fragments, or incomplete tokens.

EXAMPLE
Input sentence:
"2.2.6 Diversification
This relates to the distribution of investments of all types across different issuers (debtors) or different properties.
The Risk Limit & Trigger Manual (RLTM) outlines the entire early warning system."
"
Output:
[
  {"term": "diversification"},
  {"term": "debtor"}
  {"term": "Risk Limit & Trigger Manual"},
  {"term": "RLTM"}

]

SENTENCES

Table 3.21. Concept consolidation prompt

TASK

You are given a list of terminology entries in YAML format pertaining to the investments domain.

Identify terms that refer to the SAME CONCEPT:

- full form and abbreviation
- official long name and short form

DO NOT group:

- related but different concepts
- broader / narrower terms
- anything you are not highly confident is the same concept

If a term has no equivalent, return it alone.

OUTPUT

Return a JSON dictionary containing concept groups:

```
{
  "concept_groups": [
    { "terms": ["Term A", "Term B"] },
    { "terms": ["Term C", "Term D", "Term E"] },
    { "terms": ["Term F"] }
  ]
}
```

Table 3.22. Relation prompt.

TASK

You are a terminological analyst.

You are given a set of concepts that belong to the same broad topical area. Each concept is represented by a term and a text field that may include a definition and/or a context sentence

Your task is to analyze whether there are explicit conceptual relationships between the concepts.

This task has TWO SEQUENTIAL STEPS. You must perform Step 1 before Step 2.

STEP 1 — RELATEDNESS CHECK
For each possible pair of concepts decide whether there is a meaningful CONCEPTUAL relationship between them and if so mark it as "related": true.
A conceptual relationship exists only if the relationship is semantic and explicit,
i.e. one concept would normally be used to define, classify, contrast, or systematically relate to the other in a terminology resource, ontology, or glossary.

Mere co-occurrence in the same domain, shared topical area or typical usage together do NOT constitute a conceptual relationship.

If no relationship exists or the relationship is weak, unclear, implicit, or based only on topic proximity, you must mark the pair as unrelated ("related": false).

STEP 2 — RELATION TYPE CLASSIFICATION
ONLY for pairs marked as "related": true, classify the relationship using EXACTLY ONE of the following values:
- "superordinate" (one concept is more general than the other)
- "subordinate" (one concept is more specific than the other)
- "associative" (concepts are directly connected and this connection would normally justify a cross-reference in a glossary)
- "opposite" (concepts are contrasting or opposing)
- "equivalent" (concepts are exact synonyms, spelling, wording or formatting variants referring to the same underlying concept"
For "superordinate" or "subordinate" relations, specify the direction explicitly. For "associative" and "opposite" relations, direction is null.

If you cannot determine the type of relationship, change "related": true into "related": false.

EXPLANATION
For every pair marked as "related": true, you must provide a brief explanation.

The explanation must consist of one clear sentence and explicitly state why the relationship exists.
If you cannot formulate a clear explanation sentence, the pair must be marked as "related": false.

RULES
- Do NOT invent relationships.
- Do NOT force every concept into a relationship. Unrelated concepts are valid and expected.
- Do NOT mark as equivalent terms that have the same ID; they already are equivalent concepts.

OUTPUT FORMAT
Return your result as valid JSON using the following structure:

```
{
 "cluster_id": <cluster_id>,
 "relations": [
   {
    "concept_1": { "id": <id>, "term": "<term>" },
    "concept_2": { "id": <id>, "term": "<term>" },
    "related": true | false,
    "relation_type": "superordinate" | "subordinate" | "associative" | "opposite" | null,
    "direction": "<concept_1 → concept_2>" | "<concept_2 → concept_1>" | null,
    "explanation": "<one-sentence justification>"
   }
 ]
}
```

Include "relation_type", "direction", and "explanation" only when "related": true.

Ensure the output is strictly valid JSON.

CLUSTER