Hybrid Representation Learning for Information Extraction

Dissertation
zur
Erlangung des Doktorgrades (*Dr. rer. nat.*)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Tobias Kurt Stefan Deußer
aus
Bad Homburg v. d. Höhe

Bonn 2025

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen
Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter/Betreuer:	Prof. Dr. Rafet Sifa
Gutachter:	Prof. Dr. Christian Bauckhage
Tag der Promotion:	06.02.2026
Erscheinungsjahr:	2026

Acknowledgements

Looking back at the last five years, I can't help but feel eternally grateful for all the fond memories and rewarding experiences I have had during the journey that has culminated in this document.

In the first place, I want to thank my supervisors, Prof. Dr. Rafet Sifa and Prof. Dr. Christian Bauckhage, for always offering exceptional guidance when I needed it and giving me the freedom to explore research ideas without any restrictions. Discussing research ideas with Rafet, whether we're in the office, on a Teams call, or at a conference, always turns into a bottomless well of new directions to explore, accompanied by plenty of laughter along the way. Christian, on the other hand, kept me grounded in reality and offered valuable advice on what truly matters in a researcher's career. I would also like to extend my gratitude to Prof. Dr. Lucie Flek and Prof. Dr. Claudia Wich-Reif, who completed my dissertation committee.

To my co-authors, I deeply appreciate your invaluable collaboration. Your insights greatly enriched the quality and impact of my work and without you, this thesis would not have been possible.

Another shout-out to all current and former colleagues from the University of Bonn and Fraunhofer IAIS. Thank you for making it memorable and so much fun. Our shared conference visits and road trips will always be something I treasure and simply talking and sharing laughs at the institute or university is something priceless. To drop a few names here: Thank you to Armin Berger, Maren Pielka, Lorenz Sparrenberg, Daniel Uedelhoven, Jana Birr, Thiago Bell, Max Hahnbück, David Biesner, Thore Gerlach, Priya Tomar, Robin Stenzel, Cong Zhao, David Leonhard, Maurice Günder, David Berghaus, Sandra Halscheidt, Max Lübbering, Katherina Bieber, Tobias Schneider, and Tobias Uelwer. Lars Hillebrand, who already had to endure me during our Masters in Copenhagen, deserves a special mention here, as he was the one who got me into this mess.

To my friends, thank you for providing ample opportunities to clear my head and not having to think about the next paper or the next research idea. Without this, I probably would have finished earlier, but would have been miserable throughout.

I also want to thank my family, my parents Cornelia and Peter, my sister Sandra and brother-in-law Simon, who are a constant source of encouragement

and support. Also, thank you to Enzo and Gino, simply for being an absolute delight.

Finally, to my girlfriend and partner, Lisa. Thank you for enduring these years and your unwavering support. Thank you for simply being there.

I am grateful for everyone who took a part in this wild ride of a PhD. It has been amazing, and I would always do it again.

P.S.: Throughout this thesis, I have used fictitious examples from the *Lord of the Rings* [1–3] universe, a book series and universe I am always happy to revisit.

Abstract

In the contemporary digital era, the exponential increase in unstructured and semi-structured data has made information extraction a cornerstone of modern data-driven research and application. The ability to transform such raw information into structured knowledge is crucial for enabling later downstream tasks. While traditional rule-based and statistical approaches to information extraction have demonstrated success in narrow, well-defined tasks, they lack the scalability and adaptability required to address the vastness and variability of present-day data. Conversely, deep neural models and especially large language models have shown remarkable capabilities in language understanding, yet they remain constrained by high computational costs and susceptibility to hallucination.

This thesis explores the unification of various symbolic, statistical, and neural paradigms into a cohesive hybrid framework. The central hypothesis is that by combining the strengths of data-driven representation learning with structural, rule-based, and multimodal knowledge, one can achieve information extraction systems that are more accurate, efficient, and reliable than their monolithic counterparts. To test this hypothesis, the thesis investigates a range of hybrid architectures across five key application domains.

In the financial domain, a hybrid contradiction detection framework integrates syntactic pre-training with transformer-based representations and clustering algorithms to identify inconsistencies within large-scale financial reports. For named entity recognition, the *iNERD* algorithm introduces rule-based constraints to guide large language models, producing syntactically valid, hallucination-free entity extractions. Thereafter, the anonymisation study leverages knowledge distillation to compress the language understanding capabilities of large decoder-only models into lightweight encoder-only architectures, enabling secure and efficient text anonymisation. In relation extraction, this work presents *KPI-BERT* and the open-source *KPI-EDGAR* dataset, combining contextual embedding models with recurrent layers and noise-based regularisation to extract key performance indicators from financial documents. Extending beyond text, the final empirical contribution introduces a multimodal dementia detection framework that fuses linguistic and acoustic representations, offering a robust approach to early, non-invasive diagnosis.

Together, these studies provide compelling evidence that hybrid representation learning constitutes an important paradigm for modern information extraction. This research demonstrates that hybrid systems can achieve higher precision, stronger generalisability, and improved efficiency while remaining adaptable to real-world constraints. The findings of this thesis therefore advance the field towards more trustworthy, sustainable, and application-ready artificial intelligence.

Contents

List of Figures

List of Tables

List of Abbreviations

AD . . . . . Alzheimer's disease
API . . . . . Application programming interface
ASCII . . . . . American Standard Code for Information Interchange
BERT . . . . . Bidirectional encoder representations from transformers
BoW . . . . . Bag-of-Words
CBOW . . . . . Continuous Bag-of-Words
CPU . . . . . Central processing unit
CRF . . . . . Conditional Random Field
DAT . . . . . Dementia of the Alzheimer's Type
EDGAR . . . . Electronic Data Gathering, Analysis, and Retrieval
FTD . . . . . Frontotemporal Dementia
GDPR . . . . . General Data Protection Regulation
GloVe . . . . . Global Vectors for Word Representation
GPT . . . . . Generative pre-trained transformer
GPU . . . . . Graphics processing unit
GRU . . . . . Gated recurrent unit
iNERD . . . . . Informed Named Entity Recognition Decoding
IE . . . . . Information extraction
IOB . . . . . Inside outside beginning
IOBES . . . . . Inside outside beginning end single
JSON . . . . . JavaScript Object Notation
KPI . . . . . Key performance indicator
LBD . . . . . Lewy Body Dementia
LLM . . . . . Large language model
LSTM . . . . . Long short-term memory

MD . . . . . Mixed Dementia

MLP . . . . . Multilayer perceptron

NER . . . . . Named entity recognition

NLI . . . . . Natural language inference

NLP . . . . . Natural language processing

POS . . . . . Part-Of-Speech

RAG . . . . . Retrieval-augmented generation

RAM . . . . . Random-access memory

ReLU . . . . . Rectified Linear Unit

RNN . . . . . Recurrent neural network

RE . . . . . Relation extraction

SNLI . . . . . Stanford Natural Language Inference

tanh . . . . . Tangens hyperbolicus

TF-IDF . . . . . Term Frequency-Inverse Document Frequency

VaD . . . . . Vascular Dementia

VRAM . . . . . Video random-access memory

List of Notations

$\oplus$	Concatenation operator for vectors.
$\lvert\cdot\rvert$	Measure of magnitude or size, including absolute value for real numbers, modulus for complex numbers, norm for vectors, and cardinality for sets.
a**, a, A**	Intercept or bias.
$\mathcal{A}$	Activation function.
b	Bucket or cluster.
β	A boolean/binary value.
c	Context, such as the context in which a word appears in a sentence.
C	A text corpus, e.g., a collection of documents, paragraphs, sentences, ...
$\mathcal{C}$	Cost function.
$\mathbb{C}$	Set of complex numbers.
d	Dimensionality of a vector.
e	Entity, i.e., the entity that is extracted during named entity recognition.
e	Mathematical constant approximately equal to 2.71828, often called Euler's number.
$\mathtt{e}$	Content of an entity, e.g., a set of strings or other identifiers.
$\mathcal{E}$	Encoder that transforms its input into a representation.
$\mathbb{E}$	Set of entity classes (e.g., person, location, organisation).
f	Frequency.
$\mathcal{F}$	A (generic) function.
γ	Focusing parameter in the Focal Loss [4].
$\mathcal{G}$	Gated Recurrent Unit (GRU) [5].
i, j	Index, identifier, or a position.

i	Imaginary unit, defined as $\mathrm{i}^2 = -1$.
$\mathbf{h}$	Hidden state vector.
η	Learning rate
$\mathbf{k}$, $\mathbf{K}$	Key variable in the attention calculation [6].
κ	Cohen's κ, see [7].
ℓ	Number of elements in a sequence, cluster, or bucket. The number of elements of a vector, i.e., its dimensionality, is denoted above as d.
λ, $\boldsymbol{\lambda}$, $\boldsymbol{\Lambda}$	Hyperparameter.
l	Layer in a neural network.
$\mathcal{L}$	Loss function.
$\mathcal{M}$	Multilayer perceptron (MLP).
n, m	Number of data points, samples, observations, ...
$\mathcal{N}$	Normal distribution.
$\mathbb{N}$	Set of natural numbers.
$\mathbb{N}_0$	Set of natural numbers, including zero.
$\mathfrak{O}$	Resulting set of an overlap/intersection of two sets.
p	Probability.
$\mathcal{P}$	Pooling operation.
$\mathbf{q}$, $\mathbf{Q}$	Query variable in the attention calculation [6].
$\mathbf{r}, \mathbf{R}$	Learned representation(s) of one or multiple items (e.g., a sentence, word, image, or other item).
$\mathbb{R}$	Set of real numbers.
$\mathbb{R}_{\geq 0}$	Set of non-negative real numbers, including zero.
s	Sequence of words (usually a sentence), each word w itself being a sequence of letters.
σ	Standard deviation.
τ	Threshold parameter.
$\mathcal{U}$	Uniformly distributed noise.
$\mathbf{v}$, $\mathbf{V}$	Value variable in the attention calculation [6].
$\mathbb{V}$	Vocabulary.

w	Word or a subword/token, i.e., a sequence of letters.
w**,** $\mathbf{w}$**,** $\mathbf{W}$	Weights, e.g., the weights of a neural network.
$\mathcal{W}$	Hann window [8].
x**,** $\mathbf{x}$**,** $\mathbf{X}$	Input features or independent variable.
y**,** $\mathbf{y}$**,** $\mathbf{Y}$	Target or dependent variable.
$\hat{y}$**,** $\hat{\mathbf{y}}$**,** $\hat{\mathbf{Y}}$	Predicted value of target variable.
z**,** $\mathbf{z}$	Holds intermediary results, e.g., the output of a neuron in a neural network before an activation function is applied.
$\mathcal{Z}$	Masking operation, usually done to set certain parts of the input to zero.

It's the job that's never started as takes longest to finish, as my old gaffer used to say.

— J.R.R. Tolkien in *The Fellowship of the Ring* [1]

1 Introduction

In the contemporary digital age, we are confronted with an unprecedented proliferation of data [9]. From financial reports [10–12] and legal contracts [13–15] to clinical records [16–18] and social media communications [19–21], the vast majority of this information exists in an unstructured or semi-structured format [22, 23], like text or audio. This flood of textual and multimodal data represents a profound challenge as well as a significant opportunity. Within this raw bulk of information lies the potential to uncover critical insights [24], drive informed decision-making [25], and automate complex analytical tasks [26]. The key topic of this thesis, information extraction (IE), a cornerstone of natural language processing (NLP), is dedicated to this very pursuit: transforming unstructured data into structured, machine- and human-readable knowledge.

1.1 Motivation

Historically, information extraction systems relied on handcrafted rules and statistical methods [27–29]. These methods, while effective for narrowly defined tasks, lacked the scalability and adaptability required to contend with the sheer volume, size, and heterogeneity of modern data [30]. Thus, the development of neural networks has revolutionised the field of information extraction by delivering substantial performance gains and broadening the field's applicability [31]. Representation learning via the transformer architecture [6] has emerged as a central paradigm, enabling models to learn dense, semantic vector representations of words, sentences, and entire documents directly from data, as demonstrated by

transformer based language models like BERT [32]. These learned representations capture the intricate nuances of language, like syntax, semantics, and context, in a way that was previously unattainable [32–35]. Furthermore, advances such as convolutional networks for vision [36], transformers for language, and multimodal architectures for integrating text, images, and audio [37] all share this common foundation: they succeed by learning internal representations that expose structure in the data, serving as a bridge between raw input and task objectives and enabling applications from medical diagnostics to legal contract analysis.

The rise of large language models (LLMs) has marked another paradigm shift, offering significant capabilities in language understanding and generation, as seen in models like OpenAI's GPT [38–42] or Google's Gemini [43] and the plethora of applications of LLMs across a wide range of fields [44–50]. However, their computational expense and susceptibility to "hallucination" might present significant obstacles for tasks demanding fast results, a small computational footprint, high precision, factual accuracy, or domain-specific consistency [51–54]. We therefore believe that simply scaling up language models is not the remedy to these issues, but theorise that the path towards robust and reliable information extraction lies in a more nuanced, hybrid methodology.

This thesis, titled *Hybrid Representation Learning for Information Extraction*, argues for and demonstrates the effectiveness of such hybrid approaches. Therefore, we explore the question:

> *How can we utilise hybrid approaches to representation learning to improve the performance, efficiency, and robustness of neural networks on downstream tasks like contradiction detection, named entity recognition, relation extraction, anonymisation, or dementia detection?*

To tackle this, we hypothesise that the next frontier in information extraction will be defined by the intelligent fusion of data-driven representation learning with other forms of knowledge and structure as well as the combination of various modalities into unified representations. This includes the integration of rule-based constraints to guide generative models (see Chapter 4), the combination of different model architectures and statistical methods to leverage complementary strengths (see Chapter 3 and 6), the distillation of knowledge from larger into smaller models (see Chapter 5), and the fusion of multiple data modalities to form a more complete understanding of the task at hand (see Chapter 7).

Across a diverse set of real-world information extraction tasks like ensuring consistency in financial audits, protecting privacy, and aiding in medical

diagnostics, this thesis will demonstrate that hybrid systems consistently yield solutions that are not only capable but also reliable and efficient. It is this hybrid combination that supports the contributions of the chapters to follow.

1.2 Thesis Outline and Contributions

This thesis is structured as a collection of chapters based on peer-reviewed studies, each addressing a distinct yet related challenge in the field of information extraction. Each chapter showcases a unique application of hybrid representation learning, contributing novel models, datasets, and methodologies to the academic and practical discourse. The individual contributions to each research paper are described in the Appendix A. The overarching chapter structure of this thesis is as follows.

Chapter 2: Overview on Representation Learning

This chapter establishes the theoretical groundwork for the thesis. It provides a comprehensive overview of the fundamental principles of machine learning models and representation learning, beginning with the architectures of neural networks, from multilayer perceptrons to recurrent neural networks and the modern transformer architecture. It then dives into the specific challenges of learning representations from textual data, charting the evolution from classical methods like Bag-of-Words and TF-IDF to the dense, static embeddings of Word2Vec and GloVe, and finishing with the contextualised representations generated or internalised by modern transformer-based models like BERT or GPT. This chapter serves as the essential guide for understanding the techniques developed and deployed in the subsequent chapters.

Chapter 3: Contradiction Detection

The integrity of financial reporting is paramount for market stability and investor confidence. This chapter tackles the crucial task of identifying contradictions within financial documents, an essential aspect of the financial auditing process. We introduce two complementary hybrid methodologies. The first is a novel transformer-based classifier architecture enriched with linguistic knowledge through informed pre-training on Part-of-Speech (POS) tags. This approach demonstrates how injecting syntactic structure improves semantic understanding of "smaller" language models (here: RoBERTa [55]). The second methodology

employs LLMs in combination with an embedding-based paragraph clustering technique to detect inconsistencies across entire documents. This work highlights the power of fusing the strength of LLMs with structured data processing pipelines to create a powerful, zero- or few-shot contradiction detection system, which can even detect contradictions in already published financial reports. This system has the potential to significantly enhance the efficiency and reliability of financial audits.

Chapter 4: Named Entity Recognition

Named entity recognition (NER) is a, if not *the*, foundational task in information extraction, paving the way for later tasks like relation extraction [11] or building knowledge graphs [56], yet traditional generative methods often struggle with the dual challenges of ensuring factual accuracy and adapting to new domains [51]. This chapter presents a significant contribution to modernising NER by reformulating it as a constrained generative task. We introduce the **Informed Named Entity Recognition Decoding (iNERD)** framework, a novel hybrid approach that leverages the sophisticated language understanding of decoder-only LLMs. The core innovation lies in its informed decoding algorithm, a rule-based guardrail that forces the model's generative output to adhere to the rigid syntax of the NER task. This fusion of a learned neural system with symbolic constraints guarantees hallucination-free output, a critical requirement for reliable and production-ready information extraction. The chapter further contextualises this contribution through a comprehensive comparative analysis of LLMs on the specialised and challenging domain of legal NER, demonstrating the limitations of unconstrained prompting and reinforcing the need for hybrid solutions like iNERD.

Chapter 5: Anonymisation

The development of powerful, API-based LLMs has created a significant privacy paradox: leveraging their capabilities often requires sharing sensitive data with third-party services, that might be hosted in foreign countries with differing data protection regulations and that might misuse said sensitive data in unintended or even malign ways. This chapter addresses this critical challenge by introducing a novel, resource-efficient framework for anonymising textual data. The methodology is rooted in knowledge distillation, a technique where the text understanding of a large "teacher" LLM is transferred to a

much smaller, lightweight "student" model. This distilled student model, an efficient encoder-only transformer architecture, is then integrated into a complete pipeline with rule-based algorithms to create a robust and production-ready anonymisation system. Our experiments demonstrate that this distilled approach significantly outperforms existing baselines, achieving superior performance while maintaining a small computational footprint, making it suitable for local or on-device deployment where privacy is paramount.

Chapter 6: Relation Extraction

Beyond identifying entities, understanding the relationships between them is crucial for extracting meaningful knowledge. This chapter focuses on the task of relation extraction (RE) within the financial domain, specifically automating the extraction of key performance indicators (KPIs) and their corresponding values from financial documents. Our investigation charts an evolutionary path, beginning with the development of **KPI-BERT**, an end-to-end system that combines a BERT backbone with a recurrent neural network to perform joint NER and RE. Building on this, we introduce **KPI-EDGAR**, a novel, open-source dataset for this exact task, and propose a word-level weighted F_1 score that better captures the fuzzy boundaries inherent in financial documents. Furthermore, the chapter investigates how controlled noise injection into various transformer layers can act as a regularisation technique to improve model performance. Finally, we evaluate the capabilities of few-shot KPI extraction, providing an in-depth comparison between nine LLMs, illustrating the trade-offs in this ever evolving landscape.

Chapter 7: Dementia Detection

The final empirical chapter extends the application of hybrid representation learning into the domain of healthcare, focusing on the early and non-invasive detection of dementia. Accurate and timely diagnosis is crucial for patient care, and this work explores the potential of leveraging multimodal data to improve diagnostic accuracy. We develop a deep learning framework that analyses audio recordings from dementia patients. The core of our approach lies in the fusion of two distinct modalities: audio features, captured via spectrograms, are combined with linguistic features extracted from text transcriptions. Our findings demonstrate that this multimodal approach, which combines acoustic and semantic representations, significantly surpasses the performance of single-

modality systems, underscoring the immense promise of hybrid, multimodal deep learning for creating more robust and accessible diagnostic tools.

Chapter 8: Conclusion

The final chapter of this thesis provides a comprehensive summary of the key findings and contributions presented. It reflects on how the hybrid representation learning methodologies developed throughout this thesis collectively advance the field of information extraction. Furthermore, the chapter discusses the broader implications of these findings, highlighting that the intelligent fusion of machine learning with expert, structural, and multimodal knowledge offers a robust path towards more capable, reliable, and efficient systems. It concludes by outlining several promising avenues for future research that build upon the hybrid paradigm, including unified frameworks, dynamic constraint learning, and extended multimodal applications in domains such as healthcare and strategic reasoning.

1.3 Chapter Structure and Mathematical Notations

The core chapters of this thesis that present our empirical work, from Chapter 3 to Chapter 7, follow a shared structure. Each chapter stands on its own as a self-contained study, beginning with an abstract to summarise its core contributions. From there, an introduction outlines and motivates the specific challenge at hand. The main part of each chapter is then the methodology section, where theoretical foundations for later experiments are laid, and the experiments section, where these methods are tested and evaluated. Finally, each chapter wraps up with a conclusion, reflecting on the results and discussing their significance.

A note on the mathematical notation used consistently throughout this thesis: scalar values are denoted by lowercase italic letters, e.g., $n \in \mathbb{N}_0$ for the number of observations in a dataset. Vectors are represented by lowercase boldface letters, e.g., $\mathbf{r} \in \mathbb{R}^d$ for a learned representation of an item. Higher-dimensional variables such as matrices and tensors are denoted by uppercase boldface letters, e.g., $\mathbf{X} \in \mathbb{R}^{n \times d}$ for the input to a system. Functions are written in calligraphic style, e.g., $\mathcal{A}(\cdot)$ for an activation function in a neural network, and sets are represented in blackboard bold, e.g., $\mathbb{R}$ for the set of real numbers. For an exhaustive overview of all symbols used, the reader is referred to the List of Notations preceding the main body of this thesis.

You can prove anything you want by coldly logical reason
– if you pick the proper postulates.

— Isaac Asimov in *I, Robot* [57]

2

Overview on Representation Learning

Contents

This chapter provides a comprehensive overview of the fundamental principles underlying representation learning, focusing on its applications to address various challenges in information extraction tasks. It delves into the theoretical foundations and inherent challenges of these methods, offering a framework that sets the stage for the advanced topics discussed in subsequent chapters of this thesis. Since most of the issues addressed later in this thesis involve textual data, much of the discussion emphasises representation learning within the context of natural language processing (NLP). In the first part of this chapter, we formally introduce a few key concepts of neural networks, including three model architectures and how neural networks can be trained. Thereafter, we give an

introduction into how one can generate meaningful representations from textual data and how such textual representations have evolved.

As established in Chapter 1, this thesis investigates hybrid approaches to representation learning for information extraction. Before exploring how different methods can be combined to create more robust and efficient systems, it is essential to understand the individual components that form the building blocks of such hybrid architectures. The neural network architectures presented in Section 2.1, ranging from multilayer perceptrons to transformers, each possess distinct strengths and limitations: multilayer perceptrons excel at learning complex nonlinear mappings [58] but struggle with sequential data [59, 60]; RNNs handle sequences naturally [61] but suffer from vanishing gradients [62] and limited parallelisation [63]; transformers overcome these issues through self-attention [6] but at significant computational cost [64, 65]. Similarly, the textual representation methods discussed in Section 2.2 capture different aspects of language: classical methods like Bag-of-Words [66] and Term Frequency-Inverse Document Frequency [67] provide sparse, interpretable features [68]; distributed embeddings from Word2Vec [69, 70] and GloVe [71] capture semantic similarities in dense vector spaces [72]; and contextualised transformer representations adapt dynamically to context [32] but may lack the explicit structural knowledge of rule-based systems [73].

The subsequent chapters of this thesis demonstrate that intelligently combining these approaches, such as integrating linguistic knowledge into transformer pre-training (Chapter 3), constraining generative models with rule-based decoding (Chapter 4), fusing multiple architectures for joint tasks (Chapter 6), distilling large models into efficient ones (Chapter 5), or merging multimodal representations (Chapter 7), yields systems that are more capable, reliable, and efficient than any single method alone. This chapter thus provides the foundations for how and why these hybrid combinations succeed.

2.1 Neural Networks

To establish a solid foundation for representation learning, this section introduces neural networks as the central computational building block of most modern approaches. We begin by reviewing fundamental architectures, from multilayer perceptrons to recurrent neural networks and transformers, highlighting how each addresses different challenges in modelling structured and sequential data. This overview situates neural networks within the broader context of representation

learning and provides the formal background necessary for understanding their role in information extraction tasks, as almost all models discussed in later chapters are based on such neural networks.

2.1.1 Multilayer Perceptron

Multilayer Perceptrons (MLPs), also called feedforward neural networks or deep feedforward networks, are the classic neural network, as stated in [74], which the upcoming paragraphs are based upon. The goal of such an MLP is to approximate some function $\mathcal{F}^*$, where the Universal Approximation Theorem even states that a sufficiently large MLP can approximate any continuous function, making it a universal function approximator [58, 74]. As an example for an MLP, given the binary classifier

$$y = \mathcal{F}^*(\mathbf{x}), \tag{2.1}$$

that maps an input $\mathbf{x} \in \mathbb{R}^{n_{\text{input}}}$ of size n_{input} to a class y, an MLP defines

$$\begin{aligned} \hat{\mathbf{y}} &= \mathcal{F}_{\text{MLP}}(\mathbf{x}, \mathbf{W}) && (2.2)\\ &= \mathcal{M}(\mathbf{x}), && (2.3) \end{aligned}$$

in which it learns[1] the value of the weight parameters $\mathbf{W}$, with

$$\mathbf{W} = \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \ldots, \mathbf{W}^{(n_l)} \tag{2.4}$$

denoting the set of n_l weight matrices across all layers, where each $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ connects layer $l-1$ to layer l.

An MLP, or any kind of neural network, consists of layers of neurons, that each computes a weighted sum $z \in \mathbb{R}$ of its inputs $x_1, x_2, \ldots, x_{n_{\text{input}}}$:

$$z = \sum_{i=1}^{n_{\text{input}}} w_i x_i + a, \tag{2.5}$$

where w_i is the weight associated with the input x_i and a is the bias term. Thereafter, a nonlinear activation function $\mathcal{A}$ is applied to this weighted sum, which determines the output of the neuron:

$$y_{\text{neuron}} = \mathcal{A}(z) \tag{2.6}$$

[1] Section 2.1.4 details how this "learning" works.

Common activation functions include the *Sigmoid Function* [75], defined as

$$\mathcal{A}_{\text{Sigmoid}}(z) = \frac{1}{1 + \mathrm{e}^{-z}}, \tag{2.7}$$

the *Rectified Linear Unit (ReLU)* [76], defined as

$$\mathcal{A}_{\text{ReLU}}(z) = \max(0, z), \tag{2.8}$$

or the *Tangens hyperbolicus (tanh)* [77], defined as

$$\mathcal{A}_{\tanh}(z) = \tanh(z) \tag{2.9}$$

$$= 1 - \frac{2}{\mathrm{e}^{2z} - 1}. \tag{2.10}$$

Another widely used activation function, especially in the output layer for multi-class classification problems, is the *Softmax function* [78]. It converts a vector of size n_{classes} into a probability distribution of n_{classes} possible outcomes. For a given input vector $\mathbf{z} \in \mathbb{R}^{n_{\text{classes}}}$, the softmax function is defined as:

$$\mathcal{A}_{\text{Softmax}}(z_i) = \frac{\mathrm{e}^{z_i}}{\sum_{j=1}^{n_{\text{classes}}} \mathrm{e}^{z_j}} \quad \text{for } i = 1, \ldots, n_{\text{classes}}. \tag{2.11}$$

These neurons are stacked into layers and each layer of neurons receives the output of the previous layer's neurons (see Equation (2.6)) as input. The first or input layer ingests the original input $\mathbf{x}$ and contains n_{input} neurons. This is followed by $n_{\text{hidden}} \in \mathbb{N}_0$ layers of any size $n_{i,\text{hiddensize}} \in \mathbb{N}$. These hidden layers learn new, intermediate representations of the input data and each successive layer has the potential to learn more abstract and complex features from the representations of the previous layer [79–81]. The final layer, usually named the output layer, maps the penultimate layer's output to the actual output and therefore, contains n_{output} neurons. Figure 2.1 illustrates a simple MLP as an example. The weight matrix $\mathbf{W}$ and bias matrix $\mathbf{A}$ are the trainable component of such a neural network and are being updated during the training, as seen in later in Section 2.1.4.

In the context of information extraction, MLPs are often used as a final classification layer on top of more complex feature extractors like recurrent neural networks or pre-trained transformers to make predictions based on the learned representations and are therefore a crucial tool in many information extraction use-cases [30, 32].

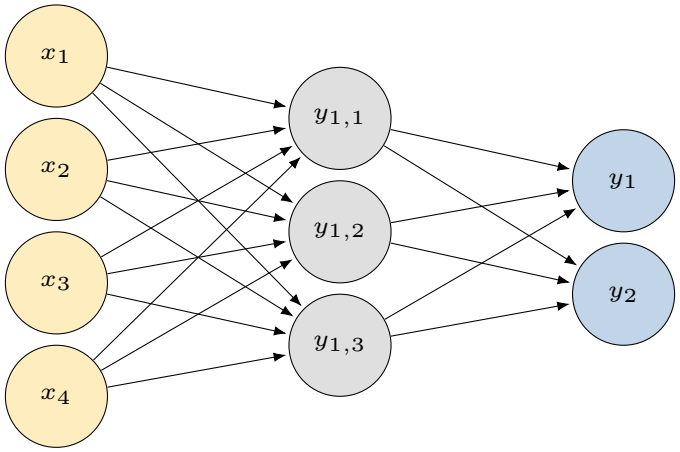


Figure 2.1: Exemplary depiction of an MLP with an input size of 4, a single hidden layer of size 3, and an output size of 2. Each node represents a neuron as defined in Equations (2.5) and (2.6).

2.1.2 Recurrent Neural Networks

While MLPs are powerful function approximators, they possess a significant limitation: they are designed to handle fixed-size inputs and assume that each input is independent of the others. This makes them inherently unsuitable for processing sequential data, such as text or time series, where context and order are crucial for understanding. For instance, in the two sentences "In the Battle of the Hornburg, Legolas and Gimli engaged in a contest" and "Legolas and Gimli engaged in a contest in the Battle of the Hornburg" we would like a model to recognise "Legolas" and "Gimli" as relevant persons, regardless of their position in the sentence [74]. If we train an MLP that processes sentences of a fixed length for this task, it would have separate parameters for each input feature/word position and therefore, would have to learn all the rules of the language separately at each position in the sentence [74]. To address this, recurrent neural networks (RNNs), like the Jordan network [82] or the Elman network [61], were introduced, a family of neural networks specifically designed for processing sequential data [74].

The defining feature of an RNN is its use of a hidden state, or memory, which allows it to persist information across time steps. At each step i in a sequence, the network takes the current input $\mathbf{x}_i$ and its hidden state from the previous step, $\mathbf{h}_{i-1}$, to compute the new hidden state $\mathbf{h}_i$ and, optionally, an output $\hat{\mathbf{y}}_i$. This recurrent connection creates a loop that enables the network to maintain a representation of the sequence seen so far, which is also illustrated in Figure 2.2.

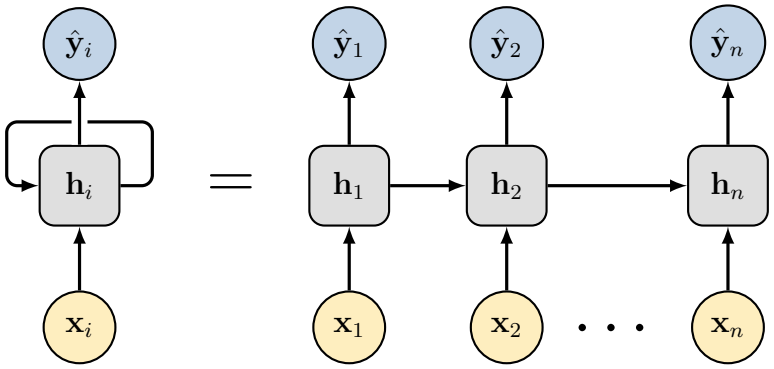


Figure 2.2: Exemplary depiction of the recurrent mechanism of a "vanilla" RNN. The figure is adapted from [83].

Elman network

The simplest form of an RNN calculates its hidden state and output using the following recurrence relations [74]:

$$\mathbf{z}_i = \mathbf{W}_{hh}\mathbf{h}_{i-1} + \mathbf{W}_{xh}\mathbf{x}_i + \mathbf{a}_h \tag{2.12}$$

$$\mathbf{h}_i = \mathcal{A}_h(\mathbf{z}_i) \tag{2.13}$$

$$\hat{\mathbf{y}}_i = \mathcal{A}_y(\mathbf{W}_{hy}\mathbf{h}_i + \mathbf{a}_y) \tag{2.14}$$

where $\mathbf{W}_{hh}$, $\mathbf{W}_{xh}$, and $\mathbf{W}_{hy}$ are the weight matrices for the hidden-to-hidden, input-to-hidden, and hidden-to-output connections, respectively. $\mathbf{a}_h$ and $\mathbf{a}_y$ are the corresponding bias terms. $\mathcal{A}_h$ is typically a nonlinear activation function like *tanh* (see Equation (2.9)), and $\mathcal{A}_y$ depends on the task (e.g., a sigmoid for binary classification or softmax for multi-class classification). The initial hidden state $\mathbf{h}_0$ is usually initialised to a vector of zeros. This structure can be visualised as being "unfolded" through time, as shown in Figure 2.2.

Despite their theoretical appeal, vanilla RNNs are difficult to train on long sequences due to the vanishing and exploding gradient problems [84, 85]. During backpropagation through time (see Section 2.1.4), gradients are multiplied at each time step, causing them to either shrink exponentially to zero (vanish) or grow uncontrollably (explode). This makes it nearly impossible for the network to learn long-range dependencies. To overcome this fundamental limitation, more complex gated RNN architectures were developed, of which two are shown herafter.

Solutions to exploding or vanishing gradient problem

The Long Short-Term Memory (LSTM) network, introduced by [86], was a solution to the vanishing gradient problem. LSTMs introduce a more sophisticated recurrent cell structure that includes a dedicated cell state and a series of gates that regulate the flow of information. These gates, the forget gate, input gate, and output gate, are small neural networks that learn which information is important to keep, add, or discard at each time step.

The Gated Recurrent Unit (GRU), introduced by [5], is a more recent and computationally simpler alternative to the LSTM. It merges the cell state and hidden state into a single hidden state vector $\mathbf{h}_i$ and uses only two gates: an update gate and a reset gate. The update gate determines how much of the past information to keep, while the reset gate decides how much to forget. By having fewer parameters, GRUs can be faster to train and sometimes outperform LSTMs on certain tasks, especially with less training data [87].

2.1.3 Transformers

While gated RNNs like LSTMs and GRUs significantly improved the ability of neural networks to capture long-range dependencies, they still retain a limitation inherited from their recurrent nature: they process information sequentially, as illustrated in Figure 2.2. This sequential processing hinders parallelisation across time steps, making them computationally intensive and slow to train on very long sequences [88]. Furthermore, even with gating mechanisms, the path length for information to travel between distant positions remains long, which can still pose a challenge for capturing very long-term dependencies [6].

To address these challenges, [6] introduced the transformer, a neural network architecture that removes recurrence entirely and relies instead on a mechanism called *self-attention* [6]. This allows the model to weigh the importance of all other words in the input sequence when processing a single word, enabling it to draw context from the entire sequence simultaneously.

Scaled Dot-Product Attention

The core component of a transformer is its attention mechanism. To calculate the scaled dot-product attention, introduced in [6], for each token in the input sequence, transformers computes three vector representations: a query and a key vector, $\mathbf{q}$ and $\mathbf{k}$, both of dimension d_k, and a value vector $\mathbf{v}$ of dimension d_v by multiplying its input $\mathbf{x}$ with three distinct weight matrices ($\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$):

$$\mathbf{q} = \mathbf{x}\mathbf{W}_q \tag{2.15}$$

$$\mathbf{k} = \mathbf{x}\mathbf{W}_k \tag{2.16}$$

$$\mathbf{v} = \mathbf{x}\mathbf{W}_v \tag{2.17}$$

The query vector can be seen to represent the current input token's request for information, the key vectors of all other tokens act as labels for the information they hold, and the value vectors contain the actual information of those tokens [72].

The attention score is calculated by taking the dot product of the query vector $\mathbf{q}$ with all the key vectors $\mathbf{k}$ in the sequence. This score is then scaled by the square root of the dimension of the key vectors, d_k. A softmax function is applied to these scores to obtain attention weights, which represent a distribution of importance over the entire sequence. Finally, the output for the query token is computed as the weighted sum of all the value vectors in the sequence. When this is computed for all tokens simultaneously using matrices $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, it is known as Scaled Dot-Product Attention:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathcal{A}_{\text{softmax}}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.18}$$

This self-attention mechanism allows the model to create context-aware representations by directly modelling the relationships between all pairs of words in a sequence, regardless of their distance from one another.

Multi-Head Attention and Positional Encoding

The transformer architecture enhances the self-attention mechanism through a procedure called *Multi-Head Attention*. Instead of performing a single attention function, the query, key, and value matrices are linearly projected multiple times into different, learned subspaces. Scaled dot-product attention is then applied in parallel to each of these "heads". This allows the model to jointly attend to information from different representation subspaces at different positions. The outputs of the parallel heads are then concatenated and linearly projected to produce the final output. Figure 2.3 illustrates this whole process in detail.

Since the self-attention mechanism is inherently permutation-invariant, as it does not naturally process the order of tokens, the model requires explicit information about the position of each token in the sequence. To this end, transformers add *positional encodings* to the input embeddings at the bottom of the encoder and decoder stacks. These encodings are vectors that give the model

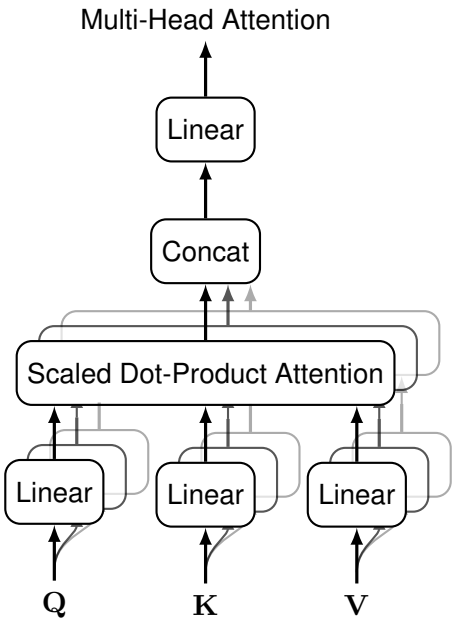


Figure 2.3: Multi-Head Attention in detail. The figure is adapted from [83].

information about the absolute or relative position of tokens. The original paper used sine and cosine functions of different frequencies for this purpose [6], but many extensions and adjustments were made of these [89–92].

Encoder and Decoder

The original transformer in [6] was designed for sequence-to-sequence tasks and consists of an encoder and a decoder stack, as depicted in Figure 2.4.

The encoder is a stack of identical layers. Each layer has two sub-layers: a multi-head self-attention mechanism and a simple, position-wise fully connected feed-forward network (which is an MLP applied to each position separately). A residual connection [93] followed by layer normalisation [94] is employed around each of the two sub-layers.

The decoder is also a stack of identical layers. In addition to the two sub-layers found in the encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. The self-attention sub-layer in the decoder is also modified to prevent positions from attending to subsequent positions, then called "masked self-attention", ensuring that the prediction for position i can depend only on the known outputs at positions less than i.

The parallelisable nature of transformers has significantly improved training efficiency and enabled the development of large pre-trained models. These

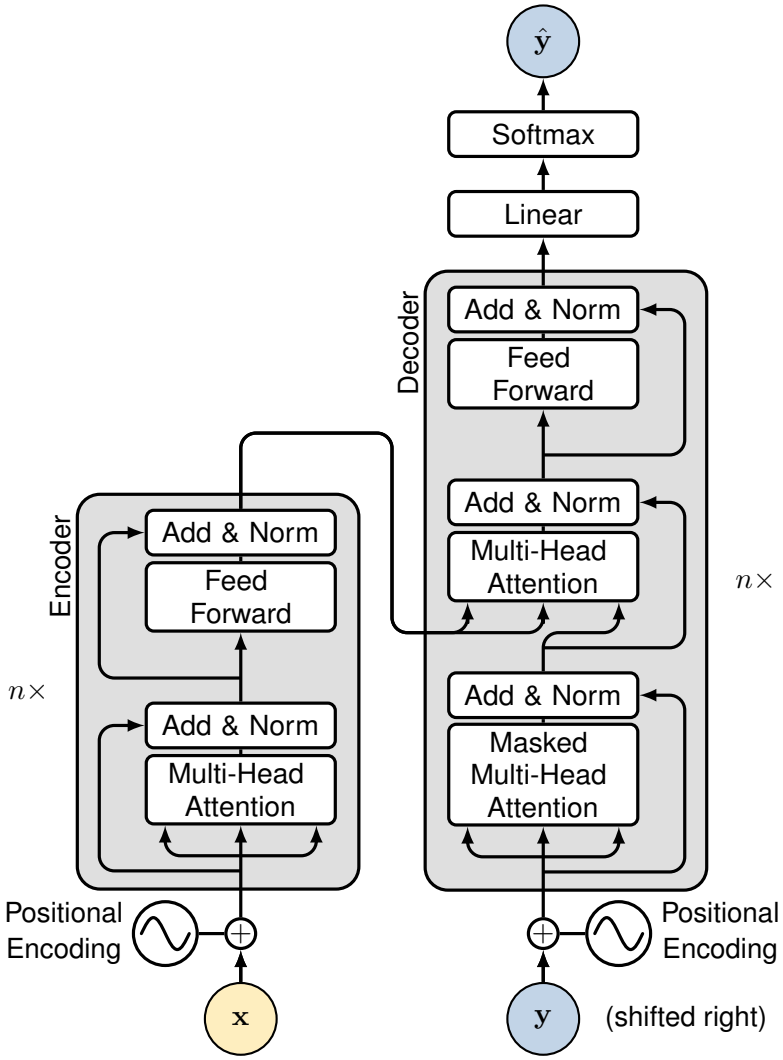


Figure 2.4: The architecture of the "vanilla" transformer model, implementing an "encoder-decoder" architecture. The figure is adapted from [83].

are typically categorised as either "encoder-only" models, which consist solely of encoder layers (e.g., BERT [32], RoBERTa [55], DIBERT [95]), "decoder-only" models, which consist only of decoder layers (e.g., GPT-1 to GPT-5 [38–42], Gemini [43], Llama [96–98]), or "encoder-decoder" models, which have both and thus, exhibit the exact structure as shown in Figure 2.4 (e.g., T5 [99], BART [100]).

Encoder-only models are typically used to generate contextualised representations of the entire input sequence, which can then be used for downstream tasks such as classification or extraction. In contrast, decoder-only models generate text autoregressively, predicting the next token conditioned on all previously generated tokens. As a consequence, encoder-only architectures are particularly well-suited for discriminative tasks such as sentiment analysis or named entity recognition,

while decoder-only architectures are naturally aligned with generative tasks such as summarisation, dialogue, or question answering.

In the context of information extraction, transformer models have become a de facto standard [30, 31]. By pre-training on large amounts of unlabelled text, these models learn rich, contextualised representations of language that can be fine-tuned to achieve exceptional performance on a wide range of tasks, including named entity recognition, relation extraction, and contradiction detection, as seen in later chapters.

2.1.4 Training a Neural Network

Training a neural network involves adjusting its parameters by updating the weight matrices $\mathbf{W}$ and biases $\mathbf{a}$, such that the network's predictions $\hat{\mathbf{y}}$ more closely match the target values $\mathbf{y}$ [74]. This is achieved by minimising a loss function $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$, which quantifies the difference between predicted and true outputs.

Gradient Descent

Gradient descent is an iterative optimisation algorithm used to update the parameters in the direction that most rapidly decreases the loss [101]. For a given weight matrix $\mathbf{W}$, the update rule is:

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}}, \tag{2.19}$$

and similarly for the biases $\mathbf{a}$:

$$\mathbf{a} \leftarrow \mathbf{a} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{a}}, \tag{2.20}$$

where η is the learning rate, a hyperparameter controlling the step size. The step size determines how far the parameters move in the direction of the negative gradient at each iteration: larger values enable faster learning but risk "overshooting" the minimum, while smaller values lead to slower yet more stable convergence. By repeatedly applying these updates across the training dataset, the network parameters gradually converge to values that minimise the loss.

Backpropagation

Backpropagation is the algorithm used to compute the gradients $\frac{\partial \mathcal{L}}{\partial \mathbf{W}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{a}}$ for all layers in the network [74, 75]. It relies on propagating the error from the output layer back through the hidden layers.

For a neuron with pre-activation value z (see Equation (2.5)) and activation $y_{\text{neuron}} = \mathcal{A}(z)$ (see Equation (2.6)), the gradient of the loss with respect to the weights $\mathbf{w}$ and connecting inputs $\mathbf{x}$ to this neuron is:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial \mathbf{w}} \tag{2.21}$$

$$= \frac{\partial \mathcal{L}}{\partial y_{\text{neuron}}} \cdot \mathcal{A}'(z) \cdot \mathbf{x} \tag{2.22}$$

and for the bias:

$$\frac{\partial \mathcal{L}}{\partial a} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial a} \tag{2.23}$$

$$= \frac{\partial \mathcal{L}}{\partial y_{\text{neuron}}}\mathcal{A}'(z) \tag{2.24}$$

where $\mathcal{A}'(z)$ denotes the derivative of the activation function with respect to its input. By recursively applying these calculations from the output layer to the input layer, we can compute gradients for all parameters to update them via gradient descent.

Backpropagation through time

While the standard backpropagation algorithm computes gradients layer by layer through a feedforward network, *backpropagation through time* extends this principle to recurrent architectures, where parameters are shared across time steps [74, 102]. Since an RNN processes sequential data by repeatedly applying the same set of weights (see Equations (2.13)–(2.14)), it can be conceptualised as a deep network unfolded through time, with one layer per time step.

During the forward pass, at each time step i, the network computes a hidden state $\mathbf{h}_i$ and possibly an output $\hat{\mathbf{y}}_i$. The total loss for a sequence of length ℓ is typically expressed as the sum (or average) of the individual time-step losses:

$$\mathcal{L} = \sum_{i=1}^{\ell} \mathcal{L}_i(\mathbf{y}_i, \hat{\mathbf{y}}_i). \tag{2.25}$$

In the backward pass, we compute gradients of this total loss with respect to all parameters by applying the chain rule through both the temporal and layer-wise dependencies. For the hidden-to-hidden weight matrix $\mathbf{W}_{hh}$, this means accumulating gradients from all time steps:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} = \sum_{i=1}^{\ell} \frac{\partial \mathcal{L}_i}{\partial \mathbf{h}_i}\frac{\partial \mathbf{h}_i}{\partial \mathbf{W}_{hh}}, \tag{2.26}$$

where $\frac{\partial \mathcal{L}_i}{\partial \mathbf{h}_i}$ depends not only on the current time step but also recursively on all future hidden states $\mathbf{h}_{i+1}, \ldots, \mathbf{h}_\ell$ due to the recurrent connections:

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{h}_i} = \frac{\partial \mathcal{L}_i}{\partial \hat{\mathbf{y}}_i} \frac{\partial \hat{\mathbf{y}}_i}{\partial \mathbf{h}_i} + \frac{\partial \mathcal{L}_{i+1}}{\partial \mathbf{h}_{i+1}} \frac{\partial \mathbf{h}_{i+1}}{\partial \mathbf{h}_i}. \tag{2.27}$$

This recursive dependency causes gradients to be repeatedly multiplied by $\frac{\partial \mathbf{h}_i}{\partial \mathbf{h}_{i-1}}$, which can lead to the *vanishing* or *exploding* gradient problem when training on long sequences (see discussion in Section 2.1.2).

2.2 Introduction to Textual Representation Learning

Even from a human point of view, many information processing or extraction tasks are either very easy or fairly hard to solve, depending on how the information is represented [74]. Take the following 6th game of the 1997 chess match[2] between IBM's Deep Blue [104] and Garry Kasparov, that can be represented as the following chess moves:

Deep Blue vs. Kasparov, Caro-Kann Defence, Steinitz Variation
1.e4 c6 2.d4 d5 3.Nc3 dxe4 4.Nxe4 Nd7 5.Ng5 Ngf6 6.Bd3 e6 7.N1f3 h6 8.Nxe6 Qe7 9.0-0 fxe6 10.Bg6+ Kd8 11.Bf4 b5 12.a4 Bb7 13.Re1 Nd5 14.Bg3 Kc8 15.axb5 cxb5 16.Qd3 Bc6 17.Bf5 exf5 18.Rxe7 Bxe7 19.c4 1-0 (Resignation)

This probably looks close to nonsense to a non-chess player, and even avid chess players likely find a better representation for this by "playing" the moves either "in their imagination" or on an actual board. So, for the first move "1.e4", an *internally visualised* or *physically set* representation would be:

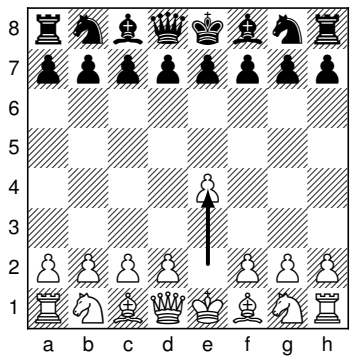


Figure 2.5: Deep Blue versus Kasparov, 1997, 1st move.

[2] See [103] for the complete chess match and analysis, from where this position was sourced from.

The final position from the move order above, in which Kasparov resigned, is:

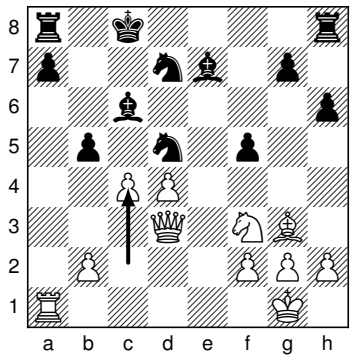


Figure 2.6: Deep Blue versus Kasparov, 1997, 19$^{\text{th}}$ move.

These representations are, to the human mind, more readable and comprehensible, highlighting that *learning* and generating good representations is paramount for the performance on downstream tasks.

Learning and exploiting representations of input data, such as text, audio, or images, is a fundamental aspect of all machine learning models discussed in subsequent chapters. Many methods integrate multiple approaches to enhance the quality of the resulting representations. However, the process of deriving meaningful representations varies considerably across modalities, both in terms of difficulty and methodology.

For instance, consider an image of resolution $d_W \times d_H$ pixels, where each pixel is represented by three integer values in the range from 0 to 255, corresponding to its red, green, and blue colour components in an 8-bit channel. This follows the additive colour model [105], commonly referred to as RGB24, and is illustrated with an example in Figure 2.7. The resulting representation, $\mathbf{X}_{\text{image}} \in \mathbb{N}_0^{d_W \times d_H \times 3}$, is in itself already computationally meaningful and can be directly employed in downstream tasks. For example, its concatenated vector form, $\mathbf{x}_{\text{image}} \in \mathbb{N}_0^{3d_W d_H}$, may be passed to an MLP to classify images into categories.

Similarly, audio data can be represented in a way that is directly amenable to machine learning models. A raw audio signal is fundamentally a one-dimensional waveform, representing the amplitude of the sound over time. While this raw waveform can be used as a direct input to a model, it is often more beneficial to transform it into a more informative, two-dimensional representation. A common approach is to compute a spectrogram, which visualises the spectrum of frequencies in the audio signal as they vary over time. We discuss such an

Figure 2.7: Representing an image as a matrix by leveraging an 8-bit per channel RGB representation. Note that each vector $\mathbf{x}_{i,j} \in \mathbb{N}_0^3$ represents the three colour channels red, green, and blue for $i = 1, \cdots, d_H$ and $j = 1, \cdots, d_W$, with the three-dimensional vector $\mathbf{x}_{1,1}$ corresponding to the top-left pixel and the remaining vectors defined analogously across the image grid.

approach in detail in Chapter 7, where we analyse how neural networks can be leveraged to detect dementia in speech recordings of patients.

In contrast, identifying an equally straightforward representation for text is less obvious. The closest analogue may be to assign each character in the input sequence a unique identifier, as in the character encoding defined by the American Standard Code for Information Interchange (ASCII) [106], which maps 128 characters to integers. For example, the string "Gandalf" can be represented in ASCII as the integer sequence "`71 97 110 100 97 108 102`". In principle, such a representation could be provided directly as input to an MLP. However, this encoding is unlikely to constitute a meaningful representation of text, and the model would therefore need to learn a more informative *internal* representation, similar to the earlier chess example.

Thus, a wide range of approaches has been developed to address the challenge of learning meaningful representations. In the following, we review some of the most common methods to represent textual data, beginning with the necessary pre-processing step of tokenisation, followed by a few key representation approaches, and concluding with transformer-based embedding models.

2.2.1 Tokenisation

As seen before, text cannot be directly represented as continuous numerical vectors in the same way as images or audio signals. Instead, it must first be decomposed into smaller units, a process known as *tokenisation*. A *token* is an atomic unit of text representation, which may correspond to a character, subword, or whole word, depending on the chosen scheme.

Example sentence: "Gandalf the Grey was determined to not let the Balrog pass."

Character-level tokenisation:

```
G | a | n | d | a | l | f | [ ] | t | h | e | [ ] | G
| r | e | y | [ ] | w | a | s | [ ] | d | e | t | e |
r | m | i | n | e | d | [ ] | t | o | [ ] | n | o | t
| [ ] | l | e | t | [ ] | t | h | e | [ ] | B | a | l
| r | o | g | [ ] | p | a | s | s | .
```

Word-level tokenisation:

```
[Gandalf] [the] [Grey] [was] [determined] [to] [not]
[let] [the] [Balrog] [pass] [.]
```

Subword-level tokenisation (BERT [32] WordPiece):

```
[G] [##anda] [##lf] [the] [Grey] [was] [determined] [to]
[not] [let] [the] [Ba] [##l] [##rog] [pass] [.]
```

Figure 2.8: Illustration of different tokenisation strategies applied to the sentence "Gandalf the Grey was determined to not let the Balrog pass.".

The simplest approach is *character-level tokenisation*, where each character is mapped to an integer identifier. While this ensures full coverage of any text, it often results in long sequences and weak semantic signals [107]. *Word-level tokenisation*, by contrast, assigns identifiers to entire words, but suffers from out-of-vocabulary issues when encountering unseen terms [108]. To balance vocabulary size and expressiveness, modern methods such as Byte Pair Encoding [109] or WordPiece [110] perform *subword tokenisation*, splitting words into frequently occurring units. Figure 2.8 illustrates these three tokenisation approaches.

2.2.2 Bag-of-Words

A simple approach to representing textual data numerically is the *Bag-of-Words* (BoW) model [66]. In BoW, a sentence is represented as a vector of word counts with respect to a predefined or "inferred" vocabulary. Each dimension of this vector corresponds to a unique word type in the vocabulary, and its value is the frequency with which that word occurs in the sentence. Formally, given a vocabulary $\mathbb{V} = \{w_1, \ldots, w_{|\mathbb{V}|}\}$ and a sentence s, the BoW representation of s is a vector

$$\mathbf{x}_{s,\text{BoW}} = \left[f_{w_1,s}, f_{w_2,s}, \ldots, f_{w_{|\mathbb{V}|},s} \right] \in \mathbb{N}_0^{|\mathbb{V}|}, \tag{2.28}$$

where $f_{w_i,s}$ denotes the raw frequency of word w_i in sentence s.

For example, the two sentences “Éowyn faced the Witch-king” and “The Witch-king faced Éowyn” would receive identical BoW representations, as the model ignores word order and syntactic structure. Despite this limitation, BoW has been widely used due to its simplicity and effectiveness in tasks such as sentence classification and clustering, especially when combined with linear classifiers [111, 112].

2.2.3 Term Frequency-Inverse Document Frequency

Building on the Bag-of-Words model, the *Term Frequency-Inverse Document Frequency* (TF-IDF) representation [67] introduces a weighting scheme that balances the frequency of a word within a sentence against its informativeness across a corpus. While Bag-of-Words represents sentences as high-dimensional count vectors, TF-IDF re-weights these counts to downplay very frequent but semantically uninformative words (e.g., “the”, “and”) and emphasise rarer, more distinctive ones.

Formally, let w denote a word and s a sentence. The *term frequency* (TF) of w in s is defined as

$$\mathrm{tf}(w, s) = \frac{f_{w,s}}{\sum_{w' \in s} f_{w',s}}, \tag{2.29}$$

where $f_{w,s}$ is the raw count of word w in sentence s, and the denominator is simply the total number of words in s.

The *inverse document frequency* (IDF) of w over a corpus C is given by

$$\mathrm{idf}(w, C) = \log\left(\frac{n_C}{1 + n_w}\right), \tag{2.30}$$

where n_C is the total number of sentences in the corpus and n_w the number of sentences where the word w appears. The constant 1 in the denominator avoids division by zero for words not present in the corpus.

The resulting TF-IDF weight of a word w in a sentence s in the context of a corpus C is

$$\mathrm{tfidf}(w, s, C) = \mathrm{tf}(w, s) \cdot \mathrm{idf}(w, C). \tag{2.31}$$

These weights replace the BoW frequencies in Equation (2.28) so that the TF-IDF representation of a sentence s is the vector

$$\mathbf{x}_{s,\text{TF-IDF}} = \big[\mathrm{tfidf}(w_1, s, C), \mathrm{tfidf}(w_2, s, C), \ldots, \mathrm{tfidf}(w_{|\mathbb{V}|}, s, C)\big] \in \mathbb{N}_0^{|\mathbb{V}|}. \tag{2.32}$$

Note that the choice of "sentence" versus "document" (or even "paragraph") depends on the level of analysis relevant to the task at hand. The mathematical formulation remains identical, with s denoting the chosen unit of analysis here.

2.2.4 Word2Vec

Representations such as BoW or TF-IDF are *local*: each word corresponds to a unique index, and all information is concentrated in that single dimension, as seen in the previous two subsections. Word representations generated by such algorithms are usually sparse and have a high dimensionality. In contrast, *distributed representations* [59, 113] encode words as dense vectors in a continuous space, where meaning is spread across multiple dimensions. Each dimension of such a vector contributes partially to the representation of a word, and similar words share overlapping components of their embeddings. This enables semantic and syntactic relationships to be reflected in geometric structures, such as distances or directions in the embedding space. For example, the words "Galadriel" and "Thranduil", both elven rulers, are orthogonal in a one-hot representation, yet should appear close together in a distributed embedding space due to their shared semantic properties.

A milestone in the development of such distributed word representations was the introduction of *Word2Vec* by [69, 70]. Unlike BoW or TF-IDF, which treat words as independent symbols, Word2Vec embeds words into a continuous vector space where semantic similarity is reflected in geometric proximity. The core intuition is that words appearing in similar contexts should have similar embeddings.

Two architectures were proposed: the *Continuous Bag-of-Words* (CBOW) model and the *Skip-Gram* model. In CBOW, the model predicts a target word w_i given its surrounding context words $\{w_{i-n_c}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+n_c}\}$ within a context window of size n_c. In Skip-Gram, the task is inverted, given a target word w_i, the model predicts its surrounding context words. Therefore, skip-gram and CBOW are basically the same in architecture. The only difference is the objective function utilised during training.

Formally, the Skip-Gram objective maximises the log-probability of observing context words given a centre word:

$$\mathcal{C}_{\text{Skip-Gram}} = \sum_{i=1}^{|C|} \sum_{-n_c \leq j \leq n_c,\, j \neq 0} \log P(w_{i+j} \mid w_i), \tag{2.33}$$

where $|C|$ is the corpus length in tokens, n_c the context window size, w_i the centre word at position i, and w_{i+j} a context word of w_i at relative position j. The conditional probability is parameterised using the softmax function:

$$P(w_{i+j} \mid w_i) = \frac{\mathrm{e}^{\mathbf{r}_{w_{i+j}}^\top \mathbf{r}_{w_i}}}{\sum_{w'=1}^{\mathbb{V}} \mathrm{e}^{\mathbf{r}_{w'}^\top \mathbf{r}_{w_i}}}. \tag{2.34}$$

However, this formulation is impractical as the cost of computing the conditional probability defined in (2.34) above is proportional to $|C|$, which tend to be quite large. To enable training on large vocabularies, approximations such as *hierarchical softmax* [114] and *negative sampling* [115, 116] were introduced.

Word2Vec was among the first models to demonstrate that distributed embeddings capture rich linguistic regularities. Famous examples include linear analogies such as

$$\mathbf{r}_{\text{king,W2V}} - \mathbf{r}_{\text{man,W2V}} + \mathbf{r}_{\text{woman,W2V}} \approx \mathbf{r}_{\text{queen,W2V}},$$

illustrating how semantic and syntactic relations are encoded in the learned vector space of distributed word representations.

2.2.5 Global Vectors for Word Representation

Building on the success of prediction-based models like Word2Vec, [71] introduced the *Global Vectors for Word Representation* (GloVe) model. Whereas Word2Vec relies on local context windows, GloVe leverages global statistical information about word co-occurrences across an entire corpus. The central idea is that ratios of co-occurrence probabilities contain rich semantic information, and embeddings should be trained to preserve these relationships.

Let f_{ij}^{GloVe} denote the number of times word w_j appears in the context of word w_i across a large corpus. GloVe learns word embeddings by minimising the following weighted least-squares objective:

$$\mathcal{C}_{\text{GloVe}} = \sum_{i=1}^{|\mathbb{V}|} \sum_{j=1}^{|\mathbb{V}|} \mathcal{F}_W\left(f_{ij}^{\text{GloVe}}\right) \left(\mathbf{r}_{i,\text{GloVe}}^\top \mathbf{r}_{j,\text{GloVe}} + a_i + \tilde{a}_j - \log\left(f_{ij}^{\text{GloVe}}\right)\right)^2, \tag{2.35}$$

where $\mathbf{r}_{i,\text{GloVe}}, \mathbf{r}_{j,\text{GloVe}} \in \mathbb{R}^{d_{\text{GloVe}}}$ are the target and context word embeddings of dimension d_{GloVe}, a_i and $\tilde{a}_j$ are bias terms, and $\mathcal{F}_W(\cdot)$ is a weighting function that downweights rare and overly frequent co-occurrences, in [71] defined as:

$$\mathcal{F}_W(x) = \begin{cases} \left(\frac{x}{\lambda_{\max}}\right)^{\lambda_\alpha}, & \text{if } x < \lambda_{\max}, \\ 1, & \text{otherwise.} \end{cases} \tag{2.36}$$

[71] proposed to set the Hyperparameters $\lambda_{\max} = 100$ and $\lambda_\alpha = 0.75$, which they found to work well across large corpora.

2.2.6 Learned Representations from Transformers

The introduction of transformer-based models (see Section 2.1.3) has fundamentally changed how representations of language are learned. Instead of relying on static embeddings, such as Word2Vec or GloVe, these models generate *contextualised representations* that adapt to the surrounding text. The embedding of a word is therefore not static but changes depending on the sentence in which it appears, capturing nuanced meaning shifts.

For example, static embeddings would assign the same vector to the word "bow", regardless of whether it appears in the sentence "Legolas' favourite weapon is the bow" or "The hobbits did not bow before Aragorn". In contrast, transformer-based models such as BERT [32] produce distinct embeddings for these two occurrences of "bow", since one context refers to a weapon to shoot projectiles with and the other to the action of bending forward at the waist. This dynamic adjustment of meaning in context is what enables transformer representations to capture such nuanced shifts.

Encoder-Only Models

Encoder-only architectures, such as BERT [32], RoBERTa [55], and their successors, learn contextual embeddings by pre-training on large text corpora using two main objectives: *masked language modelling* and, in the original BERT, *next sentence prediction*. In masked language modelling, random tokens in the input are replaced with a special `[MASK]` token, and the model is trained to predict the original tokens from their surrounding context. This encourages the encoder to build internal representations that integrate bidirectional context, unlike autoregressive models that rely on left-to-right predictions. The second objective, next sentence prediction, is a binary classification task where the model is given two segments of text and must decide whether the second segment is the actual continuation of the first or a randomly sampled sentence from the corpus. The goal of next sentence prediction is to make the model sensitive to inter-sentence coherence

and sentence-level relationships, which is particularly useful for downstream tasks such as question answering or natural language inference.

Analysis of BERT's layers [34, 117] reveals that its hidden representations encode a roughly hierarchical progression of linguistic features resembling the stages of a classical NLP pipeline. Early layers tend to capture surface-level information such as part-of-speech tags and basic syntactic information, while in the higher layers, representations become more specialised toward semantic role labelling and co-reference.

Decoder-Only Models

Decoder-only models, such as the GPT family [38–42], Gemini [43], and LLaMA [96–98], are trained autoregressively to predict the next token in a sequence given all previous tokens. Even though their training objective is a generative one, the internal hidden states of these models also serve as contextualised representations. Each layer of the decoder stack refines token embeddings by integrating longer-range dependencies and higher-level abstractions, such that intermediate activations encode increasingly abstract semantic and syntactic features [118–120].

These internal representations have been shown to encode information beyond surface-level word identity, including syntactic structure, factual knowledge, and latent in-context concepts relevant for reasoning [118, 121, 122]. For information extraction tasks, hidden states from decoder-only LLMs can thus be directly leveraged, fine-tuned, or adapted through prompting strategies to serve as high-quality feature vectors [123–125].

Doublethink means the power of holding two contradictory beliefs in one's mind simultaneously, and accepting both of them.

— George Orwell in *Nineteen Eighty-Four* [126]

3 Contradiction Detection

Contents

In this chapter, we unify two complementary approaches to detecting and correcting contradictions in financial reports, an essential information extraction task in auditing. First, we introduce a hybrid transformer-based architecture enriched with Part-Of-Speech (POS) knowledge through informed pre-training. By fine-tuning on both the Stanford Natural Language Inference Corpus (SNLI) [127] and a proprietary real-world financial contradiction dataset, the model achieves an outstanding contradiction detection F_1 score of 89.55%, substantially outperforming multiple baselines. Notably, financial-document-specific transformer models yield lower performance compared to more general embedding approaches in this setting. Second, we employ large language models (LLMs) in combination with an

embedding-based paragraph clustering methodology to enable the contradiction detection process across whole documents. Evaluations span three datasets, a publicly available annotated corpus, our aforementioned proprietary annotated dataset, and an unannotated corpus, demonstrating strong zero-shot performance. The results highlight the significant potential of automated contradiction detection as an information extraction strategy, substantially boosting auditing efficiency by reducing the time and effort required for a comprehensive and reliable financial report review.

3.1 Introduction

This chapter is based on our publications **"Contradiction Detection in Financial Reports"** (co-authored with Maren Pielka, Lisa Pucknat, Basil Jacob, Tim Dilmaghani, Mahdis Nourimand, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa) and **"Uncovering Inconsistencies and Contradictions in Financial Reports using Large Language Models"** (co-authored with David Leonhard, Lars Hillebrand, Armin Berger, Mohamed Khaled, Sarah Heiden, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa) published in the proceedings of the 4^{th} *Northern Lights Deep Learning Conference* [128] and the 11^{th} *IEEE International Conference on Big Data* [129], respectively.

Building on the theoretical foundation of hybrid representation learning introduced in Chapter 2, this chapter marks the first empirical exploration of how hybrid architectures can enhance the accuracy and interpretability of information extraction systems. In the domain of financial reporting, where consistency and factual precision are critical [130], we apply this paradigm to the detection of textual contradictions.

Contradictions in written text are widespread and can range from the merely amusing to the gravely consequential. While some may elicit a chuckle, as in the case of a newspaper article asserting that the "earth circles the moon in 365 and a fraction days"[1] when explaining the astronomy behind the summer solstice, others pose serious risks, particularly in financial reporting. In this chapter, we focus on the latter: contradictions in financial documents that, if left unrecognised or uncorrected, can lead to "bad operational decisions, reputational damage, economic loss, penalties, fines, legal action and even bankruptcy" [131].

Financial contradiction detection encompasses both *numeric* and *semantic* inconsistencies. Numeric contradictions arise when specific figures or values

[1] Printed in the article *Ottawa vs. the equator* by the *Ottawa Citizen* on the 20^{th} of June 2012.

differ within the same report (e.g., a net profit stated as $500 in one section but $600 in another). Several approaches exist to address this form of consistency checking [132, 133]. In this work, however, we concentrate on *semantic* contradictions, whose identification depends not purely on numerical comparisons but on the inferred meaning of, and relationships between, statements. Given the example from our paper [128]:

> "On 14th of March, 2020, we increased our capital by offering 5,000 new shares during a seasoned equity offering."
>
> "During 2020, we did not increase our total amount of equity and thus, it remained unchanged at $10,000,000."

Each statement in isolation is valid and numerically consistent, however the combination of both is contradictory as issuing 5,000 new shares would, in reality, increase the company's equity. Recognising such semantic contradictions requires both contextual and domain-specific financial knowledge.

3.1.1 Motivation and Context

In global finance and accounting, the accuracy and reliability of financial reports are paramount. These documents form the bedrock upon which key decisions are made by investors, analysts, regulators, and other stakeholders. Nevertheless, the financial reporting landscape is highly complex, characterised by a multiplicity of data sources[2] and reporting standards such as the International Financial Reporting Standards (IFRS), the German Commercial Code (*Handelsgesetzbuch*, HGB), or Japan's Modified International Standards (JMIS). In such a complex setting, any contradiction or inconsistency in published financial reports can lead to skewed investment decisions, regulatory compliance issues, and the erosion of market confidence [134].

Advances in NLP, particularly with LLMs such as the GPT family [38–42], Gemini [43], and LLaMA [96–98], offer promising avenues for automating the detection of these contradictions. Because LLMs can be trained to understand

[2]Examples of financial report data sources include the EDGAR database by the U.S. Security Exchange Commission (sec.gov/edgar/search-and-access), the SEDAR+ database by the Canadian Securities Administrators (sedarplus.ca), and the DART database by the South Korean Financial Supervisory Service (englishdart.fss.or.kr/).

complex financial relationships [44, 135, 136] and metrics [132], they are well-positioned to identify inconsistencies arising from the interplay of domain-specific concepts.

At its core, contradiction detection is the task of identifying when two statements contain information that cannot simultaneously be true. This task is a foundational challenge in natural language understanding because it requires a model to move beyond surface-level text similarity and perform deeper semantic reasoning. Contradictions can arise from various sources, such as antonymy, negation, numerical mismatches, or more complex world knowledge [137]. Identifying these inconsistencies is often difficult, as it likely demands nuanced contextual understanding and, in many cases, domain-specific expertise [138].

3.1.2 Our Contributions

Against this backdrop, our research explores multiple strategies for contradiction detection in financial reports, with a particular emphasis on *semantic* contradictions. In earlier works [132, 133], researchers have addressed numeric consistency in financial documents; however, here, we focus on the *semantic* dimension. Specifically, our contributions are:

- We introduce a new natural language processing task: detecting semantic contradictions in financial documents.
- We test various pre-trained language models, both *encoder-only* and *decoder-only*, on three datasets of real-world financial reports.
- The best *encoder-only* approach leverages a XLM-RoBERTa [139] model augmented with additional domain-oriented pre-training and achieves an F_1 score of 89.55%, while remaining locally deployable.
- The best *decoder-only* approach utilises a GPT-4 [41] model and achieves an F_1 score of 93.24%.

By leveraging language-model-driven methods as introduced in this chapter, companies as well as auditors can produce more accurate, reliable financial reports and thereby enable investors and creditors to make more informed and sound decisions. Ultimately, we believe such research on financial contradiction detection contributes to improving both the efficiency of the auditing process and the trustworthiness of global financial communications.

3.1.3 Structure of this Chapter

The remainder of this chapter is structured as follows. Section 3.2 reviews prior research in the fields of natural language inference and contradiction detection, establishing the context for our contributions. In Section 3.3, we detail the methodologies employed, covering our approaches for both sentence-pair and full-document contradiction detection. Thereafter, Section 3.4 describes the datasets used and presents the results of our experiments. Finally, Section 3.5 concludes the chapter by summarising our findings, discussing their implications, and outlining potential avenues for future work.

3.2 Related Work

The broader field of contradiction detection has its roots in natural language inference (NLI), also known as textual entailment. NLI aims to determine whether a given *hypothesis* can be deduced from a specific *premise*. Early interest in NLI was boosted by a challenge introduced in [140], which attracted 17 submissions. With the rise of deep, pre-trained transformer architectures, modern NLI methods commonly leverage models such as BERT [32], RoBERTa [55], and StructBERT [141], achieving state-of-the-art performance [142–147]. Interestingly, the use of LLMs in the field of NLI has proven to be difficult, as shown in [148] and [149].

Contradiction detection represents a more specific variant of NLI, in which the system must verify whether a hypothesis *directly contradicts* the premise. Earlier work frequently relied on rule-based and lexical methods. For instance, [150] drew on negation, antonymy, and pragmatic discourse information, while [151] explored a "contradiction-only" dataset, categorising contradictions into seven classes. Similarly, [152] built a rule-based framework combining shallow semantic representations with binary relations extracted via semantic role labelling.

As transformer models became more prevalent, many studies began using transformers such as BERT, RoBERTa, or GPT-like architectures [38, 39] to address contradictions in different domains. For instance, [153] identified inconsistencies in biomedical texts, [154] detected self-contradictions in an artificially-balanced Wikipedia-based corpus, and [155] sought to improve chatbot responses by locating contradictory content in preceding conversation turns. Furthermore, [156] employed semantic features and uncertainty indicators, and [157] examined sentence pairs in legal and historical domains.

Contradiction detection has also been extended to languages beyond English. [158] and [159] focussed on Spanish, [160] on Japanese, [161] on Persian, [162] on Arabic, and [163] on Chinese. Several works have also addressed German-language corpora [164–166].

While NLI and contradiction detection have proven versatile across domains, financial documents present unique challenges, including complex regulations, specialised terminology, and numeric or narrative inconsistencies. Previous work addressing numerical discrepancies include [132] and [133]. Prior to our work [128, 129], semantic contradiction in financial documents have, to the best of our knowledge, not been studied. Our works bridged this gap by proposing frameworks that apply pre-trained language models to the domain of financial reports, aiming to accurately recognise potential inconsistencies that may otherwise lead to costly misinterpretations.

3.3 Methodology

In this section, we describe our approach to address the two different scenarios for contradiction detection in financial reports. First, we investigate how one can identify contradictions within sentence-pair data. Then, we discuss the additional steps required to detect them in document-level data.

3.3.1 Sentence-Pair Data

Sentence-pair data is the usual form for many, if not the most, NLI and contradiction detection datasets, as seen in [127, 138, 140, 164, 167]. We tackle this scenario with an *encoder-only* setup, infused with additional pre-training, and a *decoder-only* setup.

Encoder-only

The overall architecture for the encoder-only setup comprises a pre-trained transformer language model $\mathcal{E}$, serving as the encoder of each sentence s and outputting a learned representation $\mathbf{r}_s$ in the form of a d-dimensional vector:

$$\mathbf{r}_s = \mathcal{E}(s) \in \mathbb{R}^d \tag{3.1}$$

This encoding step is followed by a feed-forward neural network, i.e., a multilayer perceptron (MLP), fine-tuned for the binary classification task of

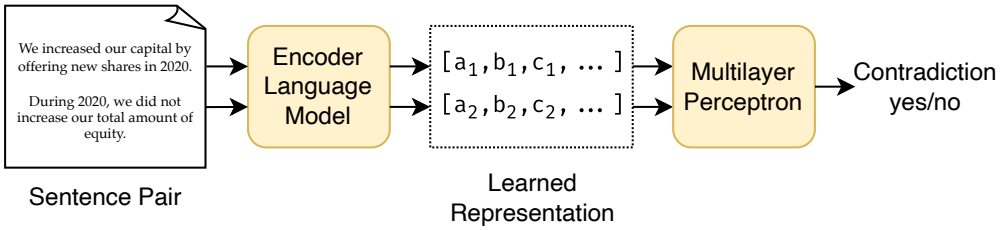


Figure 3.1: Architecture overview of our encoder-only contradiction detection approach.

contradiction detection. Formally, the MLP $\mathcal{M}$ outputs:

$$\hat{y} = \mathcal{M}(\mathbf{r}_{s_1} \oplus \mathbf{r}_{s_2}), \tag{3.2}$$

where $\mathbf{r}_{s_1}$ and $\mathbf{r}_{s_2}$ are the representations of the two input sentences, i.e., the result of Equation (3.1). This architecture is also illustrated in Figure 3.1.

We evaluate four different language models in two configurations for the encoder $\mathcal{E}$: *vanilla*, i.e., no further pre-training, and *enhanced*, i.e., augmented with additional pre-training as described further below. The four pre-trained models we evaluate are: XLM-RoBERTa [139], FinancialBERT [168], FinBERT [169], and a specialised RoBERTa variant known as Financial RoBERTa[3], which is trained on the Financial Phrasebank corpus by [170]. These models vary slightly in their architecture and hyperparameters.

XLM-RoBERTa is a multilingual transformer encoder trained on a masked language modelling task involving 100 languages. We leverage the `XLM-RoBERTa-large`[4] checkpoint, which sports an embedding dimensionality of 1024, incorporates 24 hidden layers, and uses 16 attention heads per layer, resulting in a total of 355 million trainable parameters. FinancialBERT and FinBERT are both based on the standard BERT [32] implementation, while Financial RoBERTa builds on RoBERTa [55]. They utilise the `bert-base`[5] and `roberta-base`[6] checkpoints, respectively. FinBERT and Financial RoBERTa underwent additional pre-training on the Financial PhraseBank corpus [170] for financial sentiment classification, whereas FinancialBERT is trained for next-sentence prediction and masked language modelling on a corpus of 3.39 billion tokens from the financial domain. All three models specialised on the financial domain employ an embedding dimensionality of 768 and feature 12 hidden layers with 12 attention

[3] huggingface.co/abhilash1910/financial_roberta
[4] huggingface.co/FacebookAI/xlm-roberta-large
[5] google-bert/bert-base-uncased
[6] huggingface.co/FacebookAI/roberta-base

heads each, totalling approximately 110 million trainable parameters. They are thus considerably smaller than the `XLM-RoBERTa-large` checkpoint.

Our primary motivation for the *enhancement*, i.e., additional pre-training, of an encoder-only transformer model is to improve the model's capacity for semantic understanding. To achieve this, we incorporate POS tagging as an auxiliary pre-training task, requiring the model to predict the syntactic role of each word in a sentence. Examples of possible labels include "noun", "verb", "adverb", and "determiner". Many words (e.g., "fly" or "break") can assume different syntactic roles depending on context, motivating deeper linguistic representations. We label each subword token according to the full word from which it originated. The following excerpt from our pre-training dataset illustrates a training example:

```
_We _classify _our _short  -  _term _investments _as _available  -  _for
 PRON     VERB    PRON   ADJ  PUNCT NOUN       NOUN          ADP    ADJ     PUNCT ADP
_sale  .
 NOUN  PUNCT
```

All POS tags are generated using the spaCy framework [171]. For additional implementation details, please refer to [137].

Decoder-only

To detect contradictions between two sentences, we employ a relatively straightforward approach: we query the decoder-only model, in our case an LLM, using a specific prompt and parse its output to derive a binary decision to determine whether the sentences contradict each other. The prompt we use is shown below:

> Given the following paragraph, check if there are any contradictions in it. Your answer has to start with "yes" or "no". Explain your answer.
> paragraph: "{paragraph}"

3.3.2 Document Data

When moving to the document level, an additional "bucketing" step is necessary to manage paragraphs and reduce computational overhead, as it is unfeasible and often impossible to process a whole document at once. At this bucketing step, we form sub-document buckets of thematically related paragraphs. Each bucket is then examined by an LLM to detect contradictions. A query to the LLM includes: (1) a prompt specifying processing and output instructions, and (2) the relevant paragraph collection (including paragraph IDs). The LLM responds

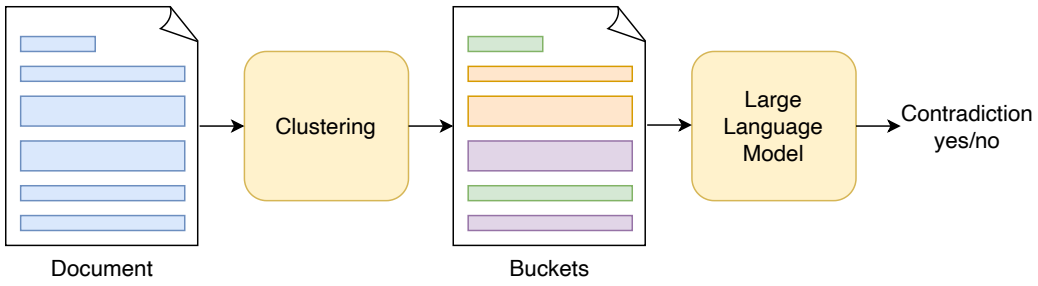


Figure 3.2: The workflow of our document-level contradiction detection approach. The figure is adapted from our paper [129].

with either "Yes" or "No" and provides a brief explanation if a contradiction is identified. Figure 3.2 illustrates this workflow. For German-language documents, the prompt is translated accordingly. An example prompt is shown below:

> Check the following document to see if it contains any contradictions. Justify your answer. Based on the justification: end your answer with "Yes" if the document contains contradictions, otherwise "No". If the document contains contradictions, cite the respective paragraph numbers in your justification. Document: "{document}"

In practice, most contradictions become apparent within one or two sentences, so the entire context of a paragraph may be unnecessary for discovering inconsistencies. Consequently, when multiple paragraphs are submitted to the LLM in a single query, irrelevant content may act as "noise" and reduce performance, as the truly relevant information then constitutes a smaller fraction of the overall prompt. We show empirical evidence of this effect in Section 3.4. Additionally, longer queries increase both computational time and costs, affecting both pay-per-query services (e.g., GPT-4 [41]) and self-hosted, open-source models (e.g., Llama [96]).

A naive solution to bucketing would be to compare each paragraph against every other paragraph and feed each pair into the LLM, effectively treating document data as a *sentence-pair* problem, as introduced in Section 3.3.1. However, the combinatorial complexity of paragraph-to-paragraph comparisons grows quadratically:

$$\begin{aligned} n_{\text{queries, paragraph-to-paragraph}} &= \frac{n_{\text{paragraphs}} \cdot \left(n_{\text{paragraphs}} - 1\right)}{2} \\ &\propto n^2_{\text{paragraphs}}. \end{aligned} \tag{3.3}$$

Since the financial reports we consider contain between 200 and 1200 paragraphs (see Section 3.4.3), this approach rapidly becomes infeasible due to

exorbitant query costs or computational demands. To mitigate these issues, we leverage paragraph representations and clustering.

Paragraph representations project text into a vector space, enabling similarity-based comparisons. While domain-specific, fine-tuned embeddings have been shown to excel in financial tasks [135], we use a more general, off-the-shelf approach via `text-embedding-ada-002` [172]. Cosine similarity is then used to group paragraphs into buckets, each containing paragraphs that are topically similar. The subsequent sections describe three different bucketing strategies for constructing these paragraph groupings.

Fixed-Length Buckets

To form buckets of a fixed size $\ell_{\text{fixed}} = n_k + 1$, we select the n_k most similar paragraphs based on cosine similarity and therefore, are applying the nearest neighbour algorithm [173]. This approach reduces the total number of LLM queries to $n_{\text{paragraphs}}$, in contrast to the paragraph-to-paragraph baseline. Furthermore, when

$$\ell < \frac{n_{\text{paragraphs}}}{2},$$

it also results in fewer total tokens passed to the LLM. Choosing a modest ℓ (on the order of 10) significantly reduces costs for commercial LLMs such as GPT-4, whose pricing is tied to the quantity of input and output tokens.

Variable-Length Buckets

The variable-length buckets approach seeks to further reduce the number of queries. Instead of selecting the n_k-nearest paragraphs, this method sets an embedding similarity threshold $\tau_{\mathbf{r}}$ of two paragraph representations $\mathbf{r}$, below which paragraphs are excluded from a bucket. Using `text-embedding-ada-002` from OpenAI, cosine-similarity scores typically approach 1, with a macro-average of 0.79 across all documents [172]. To prevent buckets from becoming excessively large, we impose a maximum size ℓ_{max}, and to avoid buckets that are too small when few paragraphs exceed $\tau_{\mathbf{r}}$, we set a minimum size ℓ_{min}. As later shown in Table 3.1, introducing a threshold $\tau_{\mathbf{r}}$ of 0.9 notably reduces the average bucket size. However, if some contradictory paragraphs do not reach the similarity threshold, they may be filtered out, leading to potential missed contradictions.

Merging Fixed-Length Buckets

The foundation of this approach is a set of buckets with a fixed size, as previously introduced. During implementation, initial experiments revealed that a single financial document frequently yields multiple buckets that share many of the same paragraphs. In our setup, each paragraph is assigned a unique integer, termed the *paragraph id*. When two fixed-length buckets of size n_k contain identical paragraph ids, they can be merged into a single bucket of size less than $2n_k$. This merged bucket is then the input for the LLM to determine if said bucket contains a contradiction.

Two primary parameters govern this merging procedure. The first is the bucket similarity threshold τ_{overlap}, which measures the overlap in paragraph ids (rather than embeddings) between buckets. The second is the maximum number of paragraph ids that a bucket may contain, denoted ℓ_{max}, which ensures that buckets, and thus the corresponding LLM prompts, do not become prohibitively large. Formally, if we consider a bucket b to be a set of integer paragraph ids, the similarity between two buckets b_1 and b_2 can be written as:

$$\text{Similarity}(b_1, b_2) = \frac{|b_1 \cap b_2|}{|b_{1,2}|}, \tag{3.4}$$

where $|b_{1,2}|$ is constant across all buckets, corresponding to the initial fixed-length bucketing. A third, implied hyperparameter is the minimum bucket size $\ell_{\text{min}} = n_k + 1$, enforced because merging occurs only after the n_k nearest neighbours have been identified. The $n_k + 1$ factor refers to one target paragraph plus its n_k neighbours.

Figure 3.3 offers a simple example of this merging process, and the full workflow for all merging methods appears in Figure 3.4.

Choosing the parameters for merging fixed-length buckets is not a trivial task. Our most important constraint is that, for the sake of reduced model output and query cost, we aim to merge paragraphs as often as possible. At the same time, we want our queries to remain as short as possible to minimise both complexity and noise in the model input, as too much noise can lead to performance degradation (see Figure 3.6). We choose the fixed bucketing method with $n_k = 10$ as a base. Then, we increase the number of additional paragraphs a bucket can hold from 0 to 9 in steps of one, resulting in $\ell_{\text{max}} \in [11, 20]$. Secondly, we iterate τ_{overlap} from 0.1 to 1.0 in steps of 0.1, thereby generating buckets for a total of 100 combinations of ℓ_{max} and τ_{overlap}. Figure 3.5 shows the percentage reduction in the number of queries for all 30 documents. The largest reduction occurs at a similarity threshold

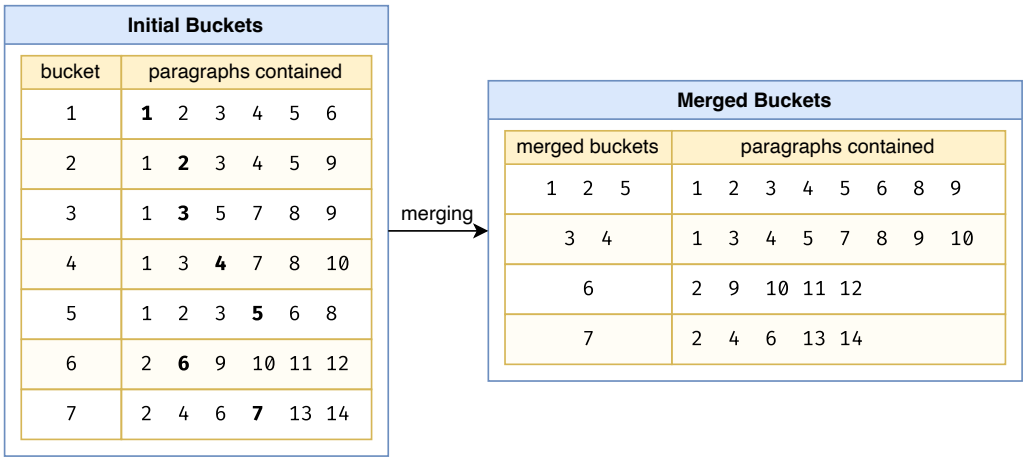


Figure 3.3: This fictitious example aims to demonstrate the idea of merging. Suppose there is a list of 7+ initial buckets with varying paragraph overlaps. Suppose further that the merging hyperparameters are $\ell_{\text{min}} = 6$, $\ell_{\text{max}} = 9$ and $\tau_{\text{overlap}} = 0.6$. Initially, there are 6 paragraphs in each bucket. Clustering the buckets according to their overlap shows that we group the first bucket with the second, the fifth, and the seventh. However, merging the second, fifth, and seventh all into the first bucket would exceed the maximum number of paragraphs ℓ_{max}. Thus, it will be only merged with the second and the fifth. The same goes for the second and the fifth buckets. This way, the data model can store the `paragraph ids` from each bucket. Given these buckets, the LLM is queried for each merged bucket only once. In the above example, the merging reduction would lead to four queries instead of the initial seven. The figure is adapted from our paper [129] and slightly altered in its layout.

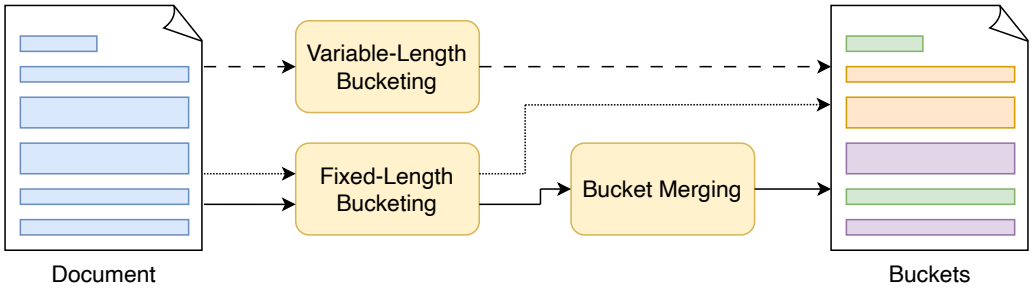


Figure 3.4: A detailed overview of the clustering procedure (Figure 3.2) when the bucket-merging approach is used. For comparison, the variable-length bucketing strategy (dashed line) and the fixed-length-only approach (dotted line) are also shown. The figure is adapted from our paper [129].

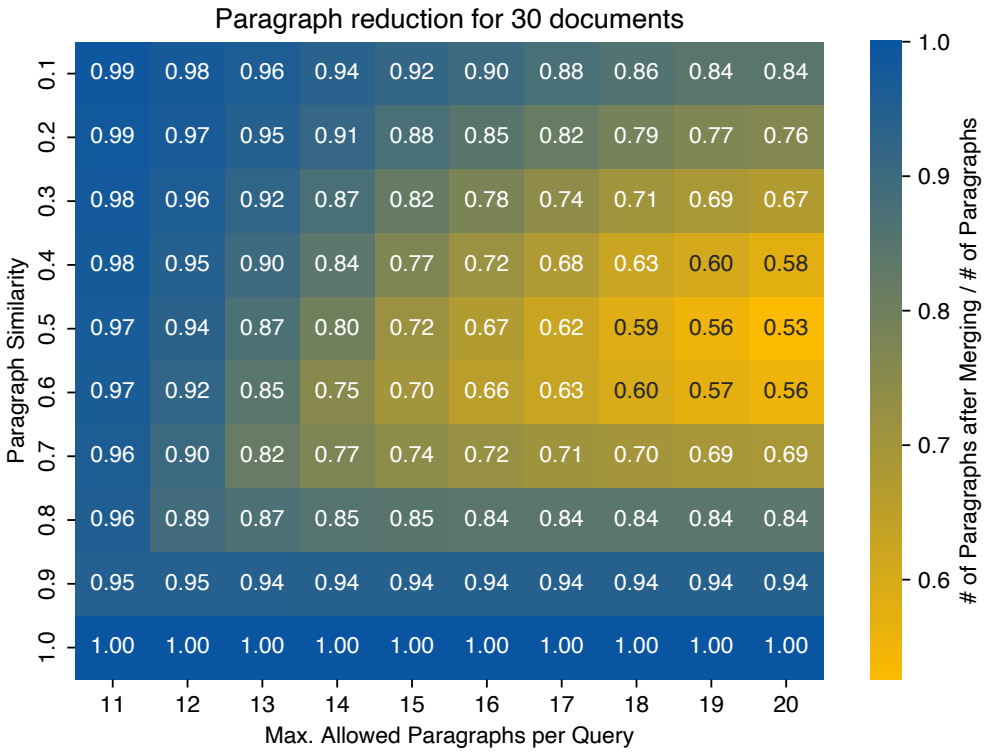


Figure 3.5: Heatmap to display the effectiveness of Merging Fixed-Length Buckets. A lower number signals a more extensive paragraph reduction, i.e., a value of x translates to a paragraph reduction of $n_{\text{paragraphs after reduction}} = n_{\text{paragraphs before reduction}} * x$. The figure is adapted from our paper [129].

of $\tau_{\text{overlap}} = 0.5$ and a value of $\ell_{\text{max}} = 20$. This outcome is expected, as a bigger maximum bucket size trivially allows for more mergers.

However, for our final evaluation, we set the parameters to $\ell_{\text{max}} = 17$ and $\tau_{\text{overlap}} = 0.6$ for two reasons. First, these values strike a balance between significantly reducing costs and output size while keeping LLM prompts relatively concise. Second, a higher similarity threshold probably has a positive impact on model performance, since merged buckets with greater similarity are likely to be more contextually cohesive.

Query Length and Cost Comparison

For a corpus of 30 documents, Table 3.1 compares both cost and query length in terms of the number of paragraphs passed to the model, in addition to a fixed prompt describing the task. In commercial services such as GPT by OpenAI [41], query fees are non-trivial; therefore, maintaining cost-effectiveness is paramount. As shown, the baseline method of comparing every possible pair of paragraphs is infeasible. The largest cost savings emerge from the bucket-merging approach

Bucketing	Bucketing Parameters	Cost in US$		Query Length
		GPT-3.5	GPT-4	
None	None	2539.28	68471.06	2
Fixed	$k = 10$	29.09	582.01	11
Variable	$\ell_{\text{max}} = 11, \ell_{\text{min}} = 3$ $\tau_{\mathbf{r}} = 0.85$	26.84	544.51	9.44
Variable	$\ell_{\text{max}} = 11, \ell_{\text{min}} = 3$ $\tau_{\mathbf{r}} = 0.9$	16.67	403.51	4.97
Merge	$\ell_{\text{max}} = 17, \ell_{\text{min}} = 11$ $\tau_{\text{overlap}} = 0.6$	**21.13**	**393.38**	**13.61**
Merge	$\ell_{\text{max}} = 20, \ell_{\text{min}} = 11$ $\tau_{\text{overlap}} = 0.5$	20.02	348.37	15.96

Table 3.1: Query length and cost comparison for the different paragraph bucketing methods. The bucketing keywords refer to the different subsections, in which the corresponding bucketing parameters are also explained. The *None* Bucketing refers to no bucketing method being used: each paragraph in a document is paired for a query with every other paragraph in the document. The cost is the estimated maximum cost for the dataset of 30 documents. The LLMs for which the theoretical maximum costs were calculated are GPT-3.5-turbo, denoted GPT-3.5, and GPT-4, using the values from openai.com/pricing as of 27 October 2023 (the time when the experiments were conducted). The actual query costs can be expected to be a bit lower, as the maximum number of input and output tokens was usually not reached in our experiments. The column *Query Length* displays the macro average of the number of paragraphs that were assigned to each query of the LLM through different bucketing methods. The bold configuration is the one used in the experiment. The table is adapted from our paper [129].

with parameters $\ell_{\text{max}} = 20$ and $\tau_{\text{overlap}} = 0.5$, while the second most efficient setting uses $\ell_{\text{max}} = 17$ and $\tau_{\text{overlap}} = 0.6$. These latter parameters are chosen for our final experiment as they deliver substantial savings while still keeping the large language model prompts relatively concise.

3.4 Experiments

This section presents the experiments undertaken to assess the performance of the methods introduced in the previous section. We use three contradiction detection datasets, described in detail in the following subsection. These include an annotated, sentence-pair-level dataset from [128], as well as an annotated and an unannotated document-level dataset introduced in [129].

3.4.1 Dataset 1: Annotated Sentence-Pair Data

Our dataset comprises 640 manually collected and annotated sentence pairs in English, derived from published financial documents (annual reports) and annotated by auditors at PricewaterhouseCoopers GmbH.

Two different procedures are used to build this dataset. In the first method, annotators are given a paragraph from a financial document and asked to formulate a plausible yet contradictory statement that could feasibly appear in another financial report. This approach is chosen because genuine contradictions in published documents are relatively uncommon, given that such documents have typically undergone auditing by the time they are public. Using this procedure, 145 examples are produced.

In the second method, annotators are presented with paragraph pairs that have been matched in advance. These pairs come from financial reports and satisfy two criteria: (i) they refer to the same legal requirement (based on prior annotations), and (ii) they surpass a text similarity threshold of 0.8, as computed by spaCy's [171] document similarity metric. The paragraphs in each pair may originate from different documents, creating a small but real possibility of encountering genuine contradictions. Annotators label each pair as `contradiction`, `no contradiction`, or `not related`. Pairs marked `not related` typically discuss unrelated facts or events and are thus not relevant for contradiction detection, so they are removed from the final dataset. This second process yields 495 examples.

Table 3.2 presents several anonymised samples from our dataset. Because the maximum sequence length is set to 512 tokens (covering premise, hypothesis, and separator tokens), a small number of data points exceed this length and must be excluded. Hence, our final dataset consists of 626 samples, of which 171 are labelled `contradiction` and 455 are labelled `no contradiction`, leading to a slightly imbalanced label distribution.

For the additional pre-training and enhancement of the encoder-only model, we draw on a collection of 47,000 paragraphs from English-language financial statements. This corpus, named the Financial Statement and Notes Data, is provided by the US Securities and Exchange Commission and is freely available online.[7]

[7] sec.gov/dera/data/financial-statement-and-notes-data-set.html

Paragraph 1	Paragraph 2	Label
Reversals of impairment losses recognized in previous years amounted to € in fiscal 2018 (2017: €). The largest reversal of impairment losses was recognized on in at € (2017: €) due to changed expectations regarding price developments.	As in the previous year, there was no requirement to recognise impairment losses or reversals of impairment losses on intangible assets in 2018.	`contradiction`
No significant events occurred after the end of the fiscal year.	No events have occurred since January 1, 2019, that will have a material impact on the net assets, financial position and results of operations of .	`no contradiction`
The total value of fixed assets in was € (previous year: €) of which, as in the previous year, none was pledged as collateral.	The total value of fixed assets in was € (previous year: €) of which, as in the previous year, € was pledged as collateral.	`contradiction`
As was the case at December 31, 2017, no treasury shares are held by at December 31, 2018.	The Executive Board is authorized, subject to the approval of the Supervisory Board, to increase the share capital by February 23, 2021, by up to € once or in several installments.	`not related`

Table 3.2: Example paragraph pairs from our financial sentence-pair contradiction dataset. Information that can be used to identify a company or individuals has been anonymized. The table is adapted from our paper [128].

Training Setup

For the encoder-only setup, as introduced and discussed in Section 3.3.1, we initialise the model parameters from their respective pre-trained checkpoints (XLM–RoBERTa–large[8], FinancialBERT[9], FinBERT[10], and Financial–RoBERTa[11]). We then perform a comprehensive grid search to identify optimal hyperparameters for each model, evaluating different parameter and pre-training configurations based on the *validation* contradiction classification F_1-score on SNLI and/or our proprietary financial contradiction dataset. Following this optimisation, we

[8] huggingface.co/FacebookAI/xlm-roberta-large
[9] huggingface.co/ahmedrachid/FinancialBERT
[10] huggingface.co/ProsusAI/finbert
[11] huggingface.co/abhilash1910/financial_roberta

Table 3.3: Test set evaluation of the contradiction detection sentence-pair dataset. We exclude the inferior configurations for FinancialBERT, FinBERT, and Financial–RoBERTa. The abbreviation *finCD* stands for our proprietary financial contradiction detection dataset. Whereas XLM-RoBERTa-large, FinancialBERT, FinBERT, and FinancialRoBERTa are encoder-only models and are thus fine-tuned in a supervised manner, GPT-3.5 Turbo and GPT-4 are decoder-only models and operate in a zero-shot environment. Both approaches are described in detail in Section 3.3.1. The table is adapted from our papers [128] and [129].

Configuration	Recall in %	Precision in %	F_1 in %
XLM-RoBERTa-large			
Fine-tuned on SNLI	70.59	68.57	69.57
Fine-tuned on finCD	67.65	76.67	71.88
Fine-tuned on SNLI & finCD	85.29	78.38	81.69
Pre-trained for POS-tagging and fine-tuned on SNLI	76.47	52.00	61.90
Pre-trained for POS-tagging and fine-tuned on finCD	82.35	80.00	81.16
Pre-trained for POS-tagging and fine-tuned on SNLI & finCD	**88.24**	**90.91**	**89.55**
FinancialBERT			
Pre-trained for POS-tagging and fine-tuned on SNLI & finCD	61.76	60.00	60.67
FinBERT			
Pre-trained for POS-tagging and fine-tuned on SNLI & finCD	64.71	56.41	60.27
Financial-RoBERTa			
Pre-trained for POS-tagging and fine-tuned on SNLI & finCD	35.29	44.44	39.34
GPT-3.5 Turbo	86.12	86.59	86.36
GPT-4	**93.01**	**93.47**	**93.24**

adopt the AdamW optimiser [174] with a binary cross-entropy loss and a linear warm-up of three epochs (for pre-training) or two epochs (for fine-tuning). The entire training procedure uses a learning rate of 5×10^{-6}. During fine-tuning, a dropout of 0.2 is applied.

Each model variant is trained for 15 epochs, and its best checkpoint is determined via early stopping[12]. For the custom part-of-speech pre-training, the model trains for a maximum of 25 epochs, since we observe slower convergence compared to fine-tuning.

For the decoder-only approach, no training setup is required, because we leverage a zero-shot framework, as described earlier.

Results

As shown in Table 3.3, our best-performing model achieves an F_1-score of 89.55% in financial contradiction detection, using an XLM–RoBERTa–large encoder with part-of-speech pre-training and fine-tuning on both the SNLI and our proprietary financial contradiction dataset.

[12]Our highest validation set F_1-score is observed at epoch 8.

More specifically, we find that the pre-training procedure leads to a significant improvement in performance. In addition, fine-tuning on both SNLI and the financial dataset yields further gains in predictive accuracy. Notably, the XLM–RoBERTa–large encoder demonstrates a clear advantage over all smaller models, even though they were specifically trained for financial documents.

Looking at the results from GPT–3.5 Turbo and GPT–4, which were evaluated in a prompting framework, the reported F_1 scores reveal that GPT–4 achieves the highest performance at 93.24%, whereas GPT–3.5 Turbo reaches 86.36%. These results confirm that GPT–4, even in a zero-shot scenario, outperforms the other models across all three metrics, including the supervised XLM–RoBERTa–large. Consequently, GPT–4 emerges as a promising option for tasks requiring highly accurate positive predictions and strong recall, showcasing its real-world utility without the need for domain-specific fine-tuning.

3.4.2 Dataset 2: Annotated Document Data

The document-level data for the second dataset is drawn from published reports following both International Financial Reporting Standards (IFRS) and the German Commercial Code (*Handelsgesetzbuch*, HGB). Five reports were selected under the criterion that most of their content is presented in paragraph form rather than in tables. Each report was then converted into machine-readable format, assigning a unique paragraph identifier (paragraph id) to every paragraph.

We introduced contradictions into the documents using three main strategies. In the first, annotators looked for redundancies, such that if two paragraphs conveyed the same information (e.g., "Aragorn is the rightful King of Gondor"), one was edited to create a contradictory statement (e.g., "Aragorn is only a simple ranger"). The second approach centred on numerical contradictions. An example would be a discrepancy in percentages that should sum to 100% (e.g., "The Morgul-host army consisted of Orcs of Mordor (95%), as well as Trolls (3%), Half-trolls of Far Harad (3%), Uruks of Mordor (2%) and the Nazgûl (0.001%)"). Numerical contradictions also include mistakes in enumerations, mismatched increases or decreases in values, and errors in simple calculations. Lastly, the third category, denoted *other*, includes contradictions that ostensibly demand a degree of general or world knowledge, such as placing a city in the wrong country (e.g., "Edoras is a city located in Mordor") or stating contradictory facts about sustainability (e.g., "Saruman felling trees on the border of the Fangorn Forest benefitted the environmental wellbeing of Isengard").

Table 3.4: Overview of generated contradictions for the annotated contradiction detection document-level dataset. The table is adapted from our paper [129].

Dataset	Documents	Contradictions		
		Redundancy-based	Numerical	Other
IFRS	5	12	13	8
HGB	5	6	16	13

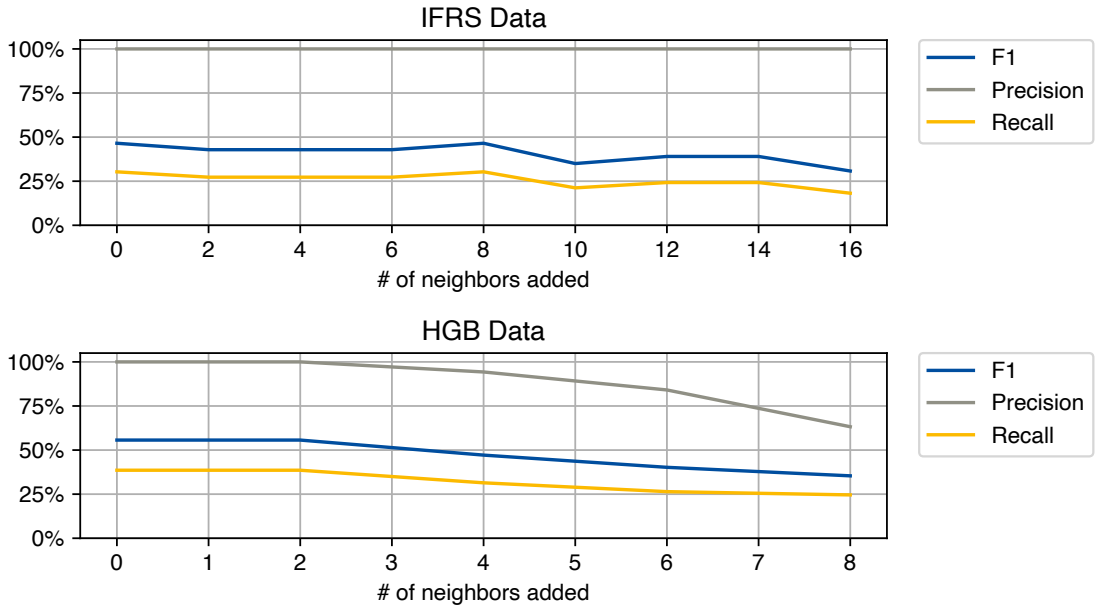


Figure 3.6: Results when we add noise, i.e., neighbouring paragraphs, to the input of the LLM. The figure is adapted from our paper [129].

All paragraphs containing these edited contradictions were recorded, permitting them to be extracted easily with or without additional noise for subsequent evaluation by the chosen large language models. Each of the five IFRS and HGB reports contains approximately 30 contradictions. A detailed overview is provided in Table 3.4.

Results

We leverage this dataset to examine the impact of noise on the performance of GPT–4. Here, "noise" refers to the inclusion of context irrelevant to the contradiction under scrutiny. To simulate this, we append "paragraph neighbours", i.e., paragraphs immediately adjacent to the contradictory text, to the query. The resulting performance outcomes are shown in Figure 3.6.

A key observation is that our model's precision in finding and identifying contradictions is significant; every contradiction it detects is indeed a valid one. In

Table 3.5: The ranking system employed by financial auditors when evaluating the output of the language model. This system is only applicable if no annotations exist, i.e., in the case of dataset 3. The table is adapted from our paper [129].

Score	Description
1	There is a contradiction in the paragraphs provided in the prompt.
2	There is no contradiction in the paragraphs provided in the prompt, but the formulation in the text is ambiguous and could be interpreted as a contradiction. Ideally, the formulation will be corrected to clear up any ambiguities. The artifact found is therefore relevant for the review.
3	There is no contradiction in the paragraphs provided in the prompt. However, this becomes apparent only in the context of missing information. Therefore, there is no contradiction, but it is justified to report the existence of a contradiction based on the information contained in the prompt.
4	There is no contradiction in the paragraphs provided in the prompt. The existing justification is technically incorrect, but this can be attributed to a lack of specific subject knowledge or faulty interpretations of terms on the part of the LLM.
5	There is no contradiction in the paragraphs provided in the prompt. The justification is also generally unusable, as it makes no sense, is not machine-readable because the output formatting specifications were not followed, or it is missing completely.

contrast, the recall is a bit lacking. Even so, without any added noise, we achieve F_1-scores of 46.51% and 55.67% for the IFRS and HGB datasets, respectively. We also confirm our expectation that including additional paragraphs (i.e., noise) in the query impairs performance. This outcome informs our decision on the maximum permissible number of paragraphs for the third and upcoming dataset and approach, as we already discussed in Section 3.3.2.

3.4.3 Dataset 3: Unannotated Document Data

The third dataset consists of 30 previously published and audited German financial reports in the *International Financial Reporting Standard* (IFRS) format. Because these reports are unannotated, we evaluate the results by presenting the model outputs to expert financial auditors, who score them based on how helpful they are for examining the financial report and identifying potential contradictions. The scoring system is detailed in Table 3.5.

After parsing, which includes removal of headings and single word paragraphs, as these provide insufficient context to hold contradictions, each document contains, on average, 261.1 paragraphs, with a median of 190.5. Furthermore, the macro-average and macro-median paragraph lengths are 46.2 and 35.0 words, respectively.

Table 3.6: Our contradiction detection model's output ranked by professional auditors. The table is adapted from our paper [129].

Score	Frequency (absolute)	Frequency (in %)
1	23	8.84
2	75	28.85
3	70	26.92
4	50	19.23
5	42	16.15

Results

Generally speaking, one would expect published and audited financial documents to be free of contradictions [175]. Therefore, based on this reasoning and the scoring system in Table 3.5, any contradiction reported by GPT–4 should to have been rated a **2** or higher by the professional auditors.

Nevertheless, Table 3.6 shows that even thoroughly audited and published financial reports can still contain genuine contradictions. Remarkably, almost 9% of the identified contradictions were confirmed as valid, and a substantial portion (28.85%) was deemed ambiguous but could be interpreted as holding contradictory statements.

In addition, the auditors tasked with reviewing the model's outputs expressed strong enthusiasm for the analysis provided, noting that such an approach could prove highly useful in their day-to-day auditing activities. Although this represents a more qualitative, "soft" metric, it underscores and emphasises the practical value of our framework.

It is important to note that, due to the datasets unannotated nature, only precision, i.e., whether a detected contradiction is indeed a contradiction, can be measured. Therefore, the recall, i.e., the proportion of all existing contradictions that are actually detected, remains unknown.

3.5 Conclusion

This chapter demonstrated how machine learning, particularly LLMs, can effectively identify contradictions in financial documents. We propose two complementary methodologies: the first leverages a transformer-based classifier, notably an XLM–RoBERTa–large model further pre-trained on POS tagging and then fine-tuned on both the SNLI [127] corpus and a proprietary financial contradiction dataset. This approach achieves an F_1-score of 89.55%, surpassing smaller

encoder models specifically pre-trained on and for financial data (FinancialBERT, FinBERT, and Financial–RoBERTa). Our findings suggest two possible reasons for this performance gap: First, the smaller size of the financial-domain models and, second, the limited transferability of their domain-specific pre-training to the task of financial contradiction detection.

Building on this classifier-based system, we also introduced a novel LLM-driven approach that tackles the challenge of detecting contradictions within full financial documents. A key innovation lies in our *bucketing* strategy, designed to mitigate the combinatorial explosion of pairwise paragraph comparisons for detecting contradictions. By intelligently grouping paragraphs, for example, via fixed-length buckets, variable-length buckets, or bucket merging, we reduce the number of necessary queries to commercial LLMs by up to 47%. Empirical evaluations on multiple datasets, including publicly available as well as confidential reports, show remarkable performance of our model and can confirm that even audited reports may occasionally contain genuine contradictions.

These results have several implications for both research and practice. First, they illustrate the power and potential of language models, either encoder-only or decoder-only, to uncover contradictions and inconsistencies beyond purely numeric mismatches, thereby enhancing the auditing process. Second, they underscore the importance of recognising the possibility of contradictions in published financial reports, suggesting that current auditing practices may benefit and further improve from additional automated contradiction checks. Third, they highlight the need for careful control of query length and complexity, particularly in commercial LLM settings where both token costs and response quality play key roles.

Looking ahead, we foresee several key avenues for future research. One priority is the integration of our contradiction detection pipeline into a production-level, machine-learning-enhanced auditing software custom-made for auditing firms. By doing so, we can continually collect real-world feedback and use these data to refine and improve the model's performance. Additionally, we plan to expand coverage to other languages other than German and English, especially for smaller companies that release annual financial reports exclusively in their language. Generating financial contradictions in a more automated manner, using a specialised or general generative model, may also alleviate the labour-intensive nature of manual annotation. A closely related challenge is *pre-filtering* paragraph pairs at scale; various strategies may be tested, including a three-way classifier (`contradiction`, `no contradiction`, or `not related`)

or two-step pipelines combining heuristic or learned filters with a contradiction detection model. Another promising direction is incorporating document-level context, for instance by linking a transformer model to a recurrent mechanism that reads paragraphs sequentially. Finally, adopting more advanced clustering algorithms, such as HDBSCAN [176] or Deep Gaussian mixture models [177], may also improve the bucketing step's effectiveness.

Taken together, we think these directions promise to strengthen automated financial auditing and ensure increased transparency, accuracy, and efficiency for the industry.

Beyond financial reporting, the methodologies and insights presented here have broader implications for contradiction detection across a diverse set of fields. The fundamental challenge of maintaining consistency in complex, multi-source documents extends to legal contracts, medical records, scientific publications, policy documents, and journalistic reporting, to name but a few. In each of these domains, contradictions can have serious consequences: legal disputes arising from inconsistent contract terms [178], patient safety risks from conflicting medical instructions [179], or erosion of public trust when policy statements contradict one another [180]. The hybrid approaches we have developed, combining transformer-based architectures with paragraph clustering, offer a generalisable framework that can be adapted to domain-specific requirements. Moreover, the insights gained from our bucketing strategies and our empirical analysis of how noise affects model performance provide valuable guidance for NLP researchers and practitioners working with long-form documents in any field where factual consistency is paramount. As LLMs continue to advance [48], we theorise that automated contradiction detection will become an increasingly essential component of quality assurance workflows across numerous sectors, helping to uphold the integrity and reliability of written communications in an era of abundant information.

Οὖτις ἐμοί γ' ὄνομα.

Noman is my name.

— Homer in *Odyssey* [181], translation by Pope [182]

4
Named Entity Recognition

Contents

This chapter explores advancements in Named Entity Recognition (NER) through hybrid representation learning. While ever-larger language models (LLMs) with ever-increasing capabilities are now well-established text processing tools, NER tasks have, until recently, largely remained reliant on the previous generation of encoder-only transformer models. Addressing this, the chapter first introduces a significant contribution: Informed Named Entity Recognition Decoding (iNERD), a novel and effective approach that reconceptualises NER as a generative process. iNERD leverages the sophisticated language understanding capabilities of recent generative models in a future-proof manner. It employs an informed decoding scheme that

incorporates the restricted nature of information extraction into open-ended text generation, thereby improving performance and efficiency whilst eliminating any risk of hallucinations. The iNERD model was coarse-tuned on a merged named entity corpus to strengthen its performance, and its evaluation across five generative language models on eight NER datasets demonstrated remarkable results, particularly in environments with an unknown entity class set, highlighting its adaptability. Subsequently, the chapter extends the investigation of LLMs in NER to a specialised and challenging domain: legal texts. We present a comprehensive comparative analysis of eleven state-of-the-art LLMs, including models such as GPT-4 and Llama-3, on legal NER tasks. The study spans seven diverse datasets across five languages. The findings reveal significant variability in LLM performance across different languages and legal contexts, with proprietary models like GPT-4 achieving the highest overall scores.

4.1 Introduction

This chapter is based on our publications **"Informed Named Entity Recognition Decoding for Generative Language Models"** (co-authored with Lars Hillebrand, Christian Bauckhage, and Rafet Sifa) and **"A Comparative Study of Large Language Models for Named Entity Recognition in the Legal Domain"** (co-authored with Cong Zhao, Lorenz Sparrenberg, Daniel Uedelhoven, Armin Berger, Maren Pielka, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa), both published in the proceedings of the *12th IEEE International Conference on Big Data* [14, 183].

Extending the hybrid paradigm established in the previous chapter, which combined specific pre-training, a clustering approach, and language models, this study applied this paradigm to named entity recognition (NER), a foundational task within information extraction. The landscape of natural language processing is now largely defined by the capabilities of generative large language models (LLMs) [184, 185]. Their proficiency in tasks requiring fluid text generation and deep contextual understanding is undisputed [129, 186]. Yet, the foundational task of NER, a cornerstone of information extraction, has lagged behind, often still relying on the previous generation of fine-tuned, encoder-only architectures, as seen in [187–190]. This chapter seeks to fill this research gap by exploring how the power of modern generative large language models can be effectively and reliably leveraged for the structured task of NER and introduces a novel constrained decoding strategy that we dubbed Informed Named Entity Recognition Decoding, shortened to iNERD.

4.1.1 Motivation and Context

At the heart of the challenge lies a fundamental problem: the open-ended, probabilistic nature of generative models clashes with the deterministic, high-precision requirements of information extraction. For NER, an output must be factually correct and structurally exact; there is no room for the creative confabulation or "hallucination" that might plague generative models [52, 191]. This makes the direct application of LLMs to NER a non-trivial problem that might demand more than simple prompting. The goal is to transform an unstructured sentence into a set of perfectly categorised entities, as shown in the example below.

Input: "Before the War of the Ring started, Buckland & Brandywine Brews (Organisation) sent a delegation from The Shire (Location) to meet with Queen Galadriel (Person) in Lothlórien (Location)."

The expected structured output from this process would be something in the form of:

(Buckland & Brandywine Brews, *Organisation*), (The Shire, *Location*), (Queen Galadriel, *Person*), (Lothlórien, *Location*)

To achieve this reliably with a generative model, we devise a new methodology, one that can leverage the model's strengths while strictly governing its output. This challenge forms the central motivation for the research presented hereafter.

4.1.2 Our Contributions

The work in this chapter makes two primary contributions towards modernising NER, approaching the problem from two distinct angles. The research introduces:

- **A Novel Generative Framework (iNERD):** We devise a novel framework that reformulates NER as a constrained generation task. The **Informed Named Entity Recognition Decoding (iNERD)** method fine-tunes decoder-only LLMs to produce a structured text output containing all entities and their types. Crucially, its hybrid nature is embodied by an informed decoding algorithm that acts as a guardrail, forcing the model to adhere to the rigid syntax of the NER task. This fusion of learned representations with rule-based constraints not only makes the system future-proof but guarantees hallucination-free output, a critical requirement for any reliable information extraction system. The system was introduced in our paper

"Informed Named Entity Recognition Decoding for Generative Language Models" [183].

- **A Comprehensive Benchmark for Legal NER:** To contextualise our primary contribution, we also map the performance landscape of applying LLMs to NER via the common method of few-shot prompting. This second study presents an exhaustive evaluation of eleven LLMs, both proprietary and open-source, on a set of seven legal NER datasets in five languages. By probing their "out-of-the-box" capabilities, we establish important baselines and expose the significant performance volatility and brittleness of prompting-based approaches, especially when faced with specialised language and strict formatting.

Through this dual-faced investigation, the chapter constructs a comprehensive argument for the potential next generation of NER systems. We begin by proposing and validating a powerful, fine-tuning-based solution that aligns generative models with the needs of information extraction. We then use a wide-ranging benchmark study to demonstrate the limitations of less-structured, prompt-based alternatives. Therefore, we reinforce the central theme of this thesis: that truly robust and advanced information extraction is best achieved through hybrid representation learning, where the formidable power of neural models is guided and controlled by explicit structural knowledge.

4.2 iNERD

Recent public releases of large language models (LLMs) with human-like writing skills have drawn unprecedented attention to natural language processing (NLP). Indeed, the performance of transformer-based LLMs increases notably, and they develop "emergent abilities", i.e., their performance increases significantly, when the number of their parameters exceeds a certain level [192].

On the other hand, tasks not based on generative transformers, such as sentiment analysis, contradiction detection, or named entity recognition, have been given lower priority in this latest push in NLP. At the time of writing the paper, they were usually and often are tackled using "encoder-only"[1] language

[1] "Encoder-only" refers to transformer models which only consist of encoder blocks. This contrasts with the original encoder-decoder structure proposed by [6] or the "decoder-only" structure of generative models [193].

models [128, 194, 195], which are typically much smaller than their "decoder-only" counterparts, or by prompting rather than fine-tuning such large language models, as seen in Section 4.3.

Here, we intend to narrow the gap between generative and extractive NLP and introduce a novel named entity recognition (NER) framework. Our Informed Named Entity Recognition Decoding (iNERD) approach has three main features: First, it utilises proven capabilities of "decoder-only" models. Our current approach works with the latest generative LLMs but can easily incorporate even better models once they become available and thus, keep pace with rapid release cycles [196], making it future-proof, computationally efficient, and quick to upgrade.

Second, we exploit the extensive pre-training and the resulting language understanding capabilities of state-of-the-art LLMs. Our approach involves an informed decoding algorithm that eliminates any hallucinations from which current models might suffer [197] and improves performance by ruling out impossible tokens during generation. To strengthen the model's understanding of the NER task, we "coarse-tune" each language model by training on a merged corpus of all NER datasets prior to fine-tuning.

Third, we propose a simple decoding strategy that allows for casting the extractive task of named entity recognition as a generative task. Our idea is to let the model generate extended texts of the following form:

> "King Théoden accepts Gondor's call to boycott exports from Mordor. **<CT>** Person **<TCS>** King Théoden **<ES>** Location **<TCS>** Gondor **<ES>** Location **<TCS>** Mordor **<ES>**",

where the special tokens inside angular brackets signal the start of the entity string (**<CT>** for **<C**ombine**T**oken**>**), separate entity type and entity content (**<TCS>** for **<T**ype**C**ontent**S**eparator**>**), and identify different entities (**<ES>** for **<E**nity**S**eparator**>**). During inference, we enforce this structure and therefore, reduce the complexity of the generation step.

Extensive evaluations show that this approach achieves remarkable performances in various NER settings, ranging from general-purpose over biomedical to finance.

In short, our contributions are the following:

- We propose a novel future-proof architecture to cast the extractive process of named entity recognition as a generative one, incorporating natural

language understanding capabilities of generative models into the process.

- We introduce a novel decoding strategy for such an architecture, which prevents the model from hallucinating and improves performance.
- We "coarse-tune" decoder-only models like Llama [96] or GPT-2 [39] on a merged named entity recognition dataset to further improve the contextual awareness of these models for NER tasks.
- We publicly provide our code as well as the weights of our best-performing model[2].

Next, we review recent related work. We then elaborate on our framework, our encoding scheme for named entities, and the corresponding informed decoding. Afterwards, we discuss our experimental protocol and the results obtained on eight benchmark datasets. Section 4.2.4 summarises our main results and provides an outlook on promising future work. We close with a statement on the potential ethical impact of our approach.

4.2.1 Related Work

Named entity recognition [198] is a fundamental task in text mining and natural language processing. Among others, it allows for anonymisation [199] or relation extraction [11] and, owing to its practical importance, has been studied with respect to standardised corpora early on (e.g., the CoNLL-2003 data collected by [200]).

Prior to the deep learning revolution, NER was usually tackled in a rule-based manner [201] or with unsupervised or feature-based supervised learning [202–205].

In their seminal paper on BERT, an encoder-only transformer, [32] achieved remarkable results on the CoNLL-2003 data by adding a classifier on top of the encoder and fine-tuning the model. Much subsequent work on similar approaches towards NER then focused on improved context awareness. To name but a few, [206] fused hierarchical contextualised representations with input token embeddings, [207] applied additional pre-training aimed at biomedical texts, and [208] added a conditional random field on top of BERT. Going even further, [209] forced entity extraction during pre-training and [187] added a co-regularisation framework for entity-centric information extraction to achieve state-of-the-art results. Nevertheless, all of these approaches are built upon an

[2] github.com/tobideusser/inerd

encoder-only transformer model and are unsuited to incorporating the decoder-only transformer architecture powering the recent popularity and success of natural language processing.

Closest to the ideas proposed in this paper, [210] formulated NER as an entity span sequence generation task, in which they added special tokens to their vocabulary to then generate entities and their types in an autoregressive fashion. [211] extended this to cover more tasks in the information extraction field. Similarly, [212] added "structure-building actions" to their vocabulary, to enable the model to generate entity boundaries and [213] employed a structured graph generation approach to generate graphs representing relations of entities. The advantage of our approach compared to the one mentioned is that we do *not* add the entity type tokens as special tokens, but as regular tokens already known to the model. In a way, this has been studied in [214] and in [215], where the extraction task was modelled as a translation task with an encoder-decoder setup, but both did not introduce a way to control for model hallucinations and used different templates. Furthermore, [216] utilised the GPT–3 [40] API to tag entities in a sentence in zero- and few-shot approaches. [217] introduced the concept of generative information extraction, which utilised generative language models to extract information in the form (subject, relation, object), and [218] applied a constrained decoding algorithm to extract events from textual data. Both applied a decoding algorithm similar to informed decoding but in a different setup and in an encoder-decoder framework. [219] used "Pattern-Exploiting Training" and a "verbaliser" to utilise an encoder-only setup to model various patterns, but did not study NER or a "decoder-only" setup.

In recent years, the field of generative language models expanded rapidly [48, 196, 220], driven by its prominent place in public discourse [221, 222], and many new models emerged and were studied, e.g., Llama [96] and its second [97] and third [98] iteration, RedPajama [223] or Mixtral [224]. A critical property of such a LLM is that performance experiences a remarkable increase once the model's scale, i.e., its parameter size, surpasses a certain threshold, dubbed "emergent abilities of LLMs", studied in [192] and [225]. Due to the sheer size of these models, reaching into the hundreds of billions, it is apparent that training and even fine-tuning them is costly and time-consuming [226–228]. To alleviate this and make the training of pre-trained LLMs accessible to a broader audience, [229] introduced LoRA, a framework that freezes the pre-trained model weights and injects trainable rank decomposition matrices into the transformer layers. Regardless, these LLMs are trained to be capable text generation tools and are,

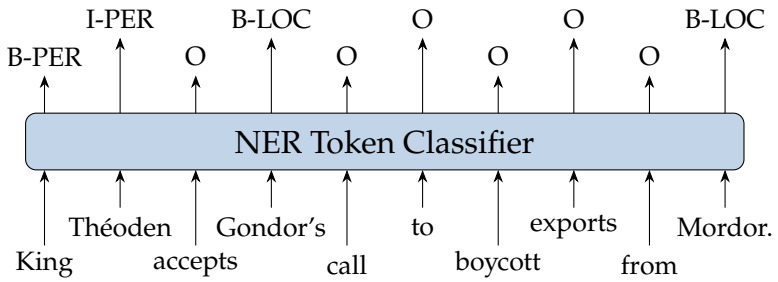


Figure 4.1: Illustration of Named Entity Recognition as a token classification task with the IOB tagging scheme. Each input token is classified either as B-Entity type, I-Entity type, or O. The figure is adapted from our paper [183].

at their state when our paper was written and without extensive prompting, mostly incompatible with other NLP tasks like NER [230, 231], a flaw that we alleviate with the iNERD approach introduced in this work.

4.2.2 Methodology

Here, we describe how we formulate named entity recognition (NER) as a task suited for generative language models. Following this, we shed light on how our algorithm for Informed Named Entity Recognition Decoding (iNERD) and the complete setup are defined and highlight its advantages compared to other approaches.

Named entity decoding

NER is usually formulated as a "token classification" task, as seen in [232] or [233]. In such a setup, an embedding of each token is generated using a text encoder, often an encoder-only transformer model like BERT [32]. This embedding is then fed into a classifier, which can be anything from a simple logistic regression to a more involved deep neural network to classify each token as either a part of an entity or not. This prediction generally has to include the entity start and entity end information, which can be achieved with, among others, the *IOB* tagging scheme [234]. Figure 4.1 illustrates this setup.

In contrast, we propose to model this task as a generative process, simplifying its machine-learning components to just one building block: a decoder-only transformer model.

To formalise this, we define the input x for our generative model during the training phase for n entities e as

$$\begin{aligned}\mathbf{x} &= s_{\text{entance}} \oplus \text{w}_{\text{CT}} \oplus \Big|\Big|^{n} (\text{w}_e \oplus \text{w}_{\text{CT}} \oplus s_e \oplus \text{w}_{\text{EST}}) \\ &= s_{\text{entance}} \oplus \text{w}_{\text{CT}} \oplus s_E,\end{aligned} \tag{4.1}$$

where

- $\oplus$ is the concatenation operator,
- s_{entance} the actual input sentence from which we intend to extract entities,
- w_{CT} the "combine token",
- w_e the type of entity e (from the set of entity classes $\mathbb{E}$),
- w_{TCS} the "type-content separator" token,
- s_e the actual entity,
- w_{EST} the "entity separator token",
- $\Big|\Big|^{n}$ concatenates its input along the number of entities n.

The concatenation $\Big|\Big|^{n} (\text{w}_e \oplus \text{w}_{\text{CT}} \oplus s_e \oplus \text{w}_{\text{EST}})$ is the entity string s_E of our input $\mathbf{x}$, i.e., what is unknown during inference and has to be predicted.

To make Equation 4.1 more accessible, we can illustrate it with the example presented in the beginning:

> "King Théoden accepts Gondor's call to boycott exports from Mordor. **<CT>** Person **<TCS>** King Théoden **<ES>** Location **<TCS>** Gondor **<ES>** Location **<TCS>** Mordor **<ES>**",

in which

- s_{entance} is the input sentence "*King Théoden accepts Gondor's call to boycott exports from Mordor*",
- w_{CT} the string "<*CT*>",
- w_e the entity types "*Person*" and "*Location*",
- w_{TCS} the string "<*TCS*>",
- s_e the actual entity content "*King Théoden*", "*Gondor's*" and "*Mordor*",

- w_{EST} the string "<*ES*>",
- s_E the entity string "<*CT*> *Person* <*TCS*> *King Théoden* <*ES*>*Location* <*TCS*> *Gondor* <*ES*> *Location* <*TCS*> *Mordor* <*ES*>".

We can then fine-tune the pre-trained decoder-only model to predict each token of the input I autoregressively using teacher forcing [235], i.e., the causal language modelling task is unchanged for these models. We calculate the loss on all predicted tokens after the combine token, w_{CT}.

Compared to the approach introduced in [210], the advantage of our framework for named entity decoding is that we do *not* add entity type tokens w_{EST} as special tokens, but as regular tokens already known to the model. Their approach, where the example sentence above becomes "King Théoden accepts Gondor's call to boycott exports from Mordor. <*PER*> King Théoden <*LOC*> Gondor's <*LOC*> Mordor", loses the meaningful embedding a transformer model has learned for w_{EST}, i.e., the model has to learn anew what the introduced special tokens mean. Furthermore, to eliminate any yet unknown special token from the training data, one could replace all special tokens with meaningful replacements already known to the model: the tokens "\n", ":", and ";" for w_{CT}, w_{TCS}, and w_{EST}, respectively.

Informed named entity recognition decoding

In the previous section on *Named entity decoding*, we only considered the training process, in which we apply teacher forcing to correct the model if it "makes a mistake" during the generation to accelerate convergence. However, during inference, applying teacher forcing would either be cheating or simply impossible if no ground truth exists.

Nevertheless, we do know quite a bit about what tokens to expect at a certain point during inference, described by these four rules:

1. After the combine token w_{CT} or the entity separator token w_{EST}, the entity type token w_e or the end-of-sequence token has to be predicted.
2. After predicting the entity type w_e, the type-content separator w_{TCS} has to be predicted.
3. After the type-content separator w_{TCS}, any token from the input s_{entance} may be predicted (signalling the start of the entity e).

Algorithm 1 iNERD for a batch size of 1

Input: Scores $\hat{\mathbf{y}}_{\text{score}}$ with the size of the vocabulary, input IDs s_{entance} holding the considered sentence and prior predictions
Parameters: Combine token w_{CT}, entity separator token w_{EST}, type content separator token w_{TCS}, set of entity type tokens $\mathbb{E}$
Output: Updated scores $\hat{\mathbf{y}}_{\text{iNERD}}$ with iNERD applied

Let w_p be the previously predicted token, i.e., the last token in the sequence $\mathbf{x}$.
Let β be a boolean value representing if we are in the "entity generation phase", i.e., if in the reversed sequence of $\mathbf{x}$ we can find the token w_{TCS} before we can find the token w_{EST}.
Let s_{entance} be the sentence considered, i.e., everything of $\mathbf{x}$ before the token w_{CT}.
if $\text{w}_p = \text{w}_{\text{CT}}$ or $\text{w}_p = \text{w}_{\text{EST}}$ **then**
    Mask $\hat{\mathbf{y}}_{\text{score}}$ to only allow entities in $\mathbb{E}$ or the end-of-sequence token.
else if $p \in \text{w}_e$ **then**
    Mask $\hat{\mathbf{y}}_{\text{score}}$ to only allow w_{TCS}.
else if β **then**
    if $\text{w}_p = \text{w}_{\text{TCS}}$ **then**
        Mask $\hat{\mathbf{y}}_{\text{score}}$ to only allow tokens present in s_{entance}.
    else
        Mask $\hat{\mathbf{y}}_{\text{score}}$ to only allow the token after w_p in s_{entance} or w_{EST}.
    end if
end if
Set $\hat{\mathbf{y}}_{\text{iNERD}}$ to $\hat{\mathbf{y}}_{\text{score}}$
return $\hat{\mathbf{y}}_{\text{score}}$

4. After a token from the input s_{entance} has been predicted, the only allowed tokens for prediction are either the entity separator token w_{EST} (signalling the end of the entity e) or the token following the previous token in the input s_{entance} (signalling the continuation of the entity e).

These four rules comprise the Informed Named Entity Recognition (iNERD) algorithm, as illustrated in Algorithm 1. This algorithm is implemented as a post-processing step and is executed after the model calculates the score over its vocabulary and before mapping this score to the actual token to be predicted.

The advantages of this approach are clear: First, the decoder-only model is unable to hallucinate, as any prediction that does not follow the decoding scheme introduced in Equation 4.1 is simply masked out, i.e., the score of this token is set to zero. Second, we can apply this model to unseen data and still expect reasonable results if we define our set of entity type tokens $\mathbb{E}$ beforehand, as later shown in the Experiments section.

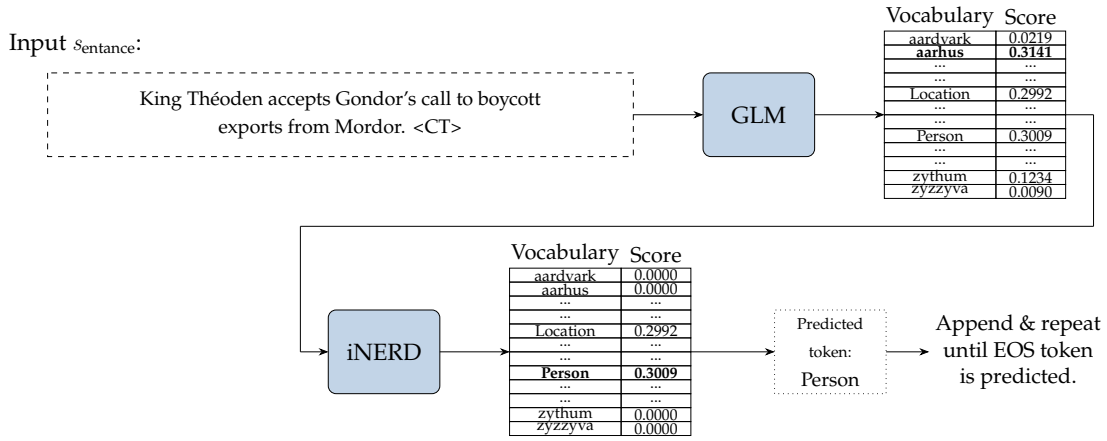


Figure 4.2: Illustration of the first step taken during the inference pipeline of our complete model setup. The model starts with the Input s_{entance} as shown in Equation 4.2, which is fed into the generative language model (GLM). This outputs a score vector $\hat{Y}_{\text{score}}$ over the vocabulary, which in turn is processed by the iNERD algorithm as described in Algorithm 1. The highest-scoring token from the vocabulary is then appended to the input, and the process starts anew. This is repeated until the model predicts the end-of-sequence token. The figure is adapted from our paper [183].

Complete Model Setup

Now, we can build our model with the blocks introduced previously. First, we transform our input to the structure described in Equation 4.1, which during training contains the entity string s_E, but becomes

$$I_{\text{Inference}} = s_{\text{entance}} \oplus \text{w}_{\text{CT}} \tag{4.2}$$

during inference. This is passed through the generative language model, which assigns a score $\hat{y}_{\text{score}}$ to each token in the vocabulary. The resulting score vector $\hat{\mathbf{y}}_{\text{score}}$ is the input to the iNERD algorithm, as described in Algorithm 1. This masks out impossible tokens for the current step, resulting in an updated score vector S_{iNERD}, which is then used to calculate the next token by taking the one with the highest score $\hat{y}_{\text{iNERD}}$. This token is concatenated with the input and the whole process is repeated until the model predicts the end-of-sequence token. Figure 4.2 illustrates this procedure for the first two steps.

4.2.3 Experiments

To practically evaluate the merits of our approach, we conducted experiments on eight datasets. Here, we describe our protocol, discuss data and results, and point out the strengths and flaws. We report the performance in *micro*-F_1 in %.

We use the model setup introduced in the previous section for all our experiments and compare it to various benchmarks from other approaches. We test various decoder-only language models for this setup, namely the 1.5 billion-parameter GPT2–XL [39], the 2.7 billion-parameter BioMedLM [236], the 3 billion-parameter RedPajama [223], and the 7 and 13 billion-parameter Llama [96]. Additionally, we apply LoRA [229], a framework that freezes the pre-trained model weights while integrating trainable rank decomposition matrices into the transformer layers, to every model with a parameter size above 3 billion.

Our general approach is as follows. We first coarse-tune on our merged NER dataset. We then evaluate each model *without additional fine-tuning* on the test set of each dataset and report the results in Table 4.1. This shows how our approach can generalise from a noisy dataset with a varying entity set $\mathbb{E}$ to a specific dataset with a clearly defined entity set. After coarse-tuning, we fine-tune on the respective training set. Additionally, we conduct an ablation study to highlight the improvements of each component of our approach.

Due to the sheer computational complexity, we only run each experiment once and do not test various seeds to take the average of each run. Furthermore, and for the same reason, we apply no hyperparameter tuning in this scenario. The fixed hyperparameters we used are an (accumulated) batch size of 16, a learning rate of 0.00001 with a weight decay of 0.01 for the Adam optimiser with weight decay [174]. The LoRA configuration, if applicable, is 8 for the rank of the update matrices, 32 for the scaling factor, and 0.1 for dropout. We are certain that the performance of our approach can be further improved if one focuses on a singular dataset and finds the optimal hyperparameter configuration for each dataset, but the computational cost of doing such a hyperparameter search is immense and beyond our financial scope and the general scope of this study, which aims to point out the general merits of our approach.

All experiments were run on a shared GPU cluster outfitted with the 40GB and 80GB versions of the Nvidia A100 GPU, an AMD EPYC 7742 CPU, and 512GB of RAM. The code is implemented in PyTorch and PyTorch Lightning, and the initial model weights were loaded from HuggingFace.

Data

We train and test on a total of eight datasets to show where our approach demonstrates notable and promising performance. Special attention is placed on the most prominent of these eight, the CoNLL–2003 dataset [200] featuring four

different entity classes and its second iteration CoNLL++ [237], which corrected 5.38% of the apparently wrongly annotated test sentences.

Furthermore, we include the OntoNotes [238] and Few–NERD [239] datasets, which are similar to CoNLL–2003 but have more granular entities (18 and 66 entity classes, respectively). For example, whereas in CoNLL–2003, we only have a coarse-grained entity type "Person", this is split into eight types in Few-NERD: "Actor", "Artist/Author", "Athlete", "Director", "Politician", "Scholar", "Soldier", and "Other".

Going a different route, the WNUT-17 [240] dataset features six different entity classes and focuses on identifying unusual, previously unseen entities in the context of emerging discussions. We also include three domain-specific datasets, two focusing on biomedical named entities (JNLPBA in [241] and NCBI-Disease in [242]) and one on financial ones (FiNER-ORD in [243]). The two biomedical datasets have five and one different entity classes, respectively, and the financial NER dataset has three.

The combined length of these datasets is 290,317 sentences for the training set, 42,016 for the validation set, and 60,477 sentences for the test set.

Coarse-tuning

As a first step, we merge all training splits of the previously discussed datasets and train a language model to predict the entity string s_E. We refer to this step as "coarse-tuning" the pre-trained language model, as it involves infusing the model with a general understanding of named entities.

It is important to note that the models must handle considerably noisy data because the set of entity type tokens $\mathbb{E}$ vary across datasets. For example, the CoNLL-2003 dataset has four different entity type tokens, whereas the Few-NERD dataset has 66. Despite this variability, we hypothesise that by exposing the model to the general structure introduced in Equation 4.1, it can gain valuable insights. Specifically, the model might learn to associate coarse-grained entity types, such as "Organisation" in CoNLL-2003, with their fine-grained subtypes, like "Company" in Few-NERD. This hypothesis is later confirmed in the ablation study.

Results without dataset specific fine-tuning

Looking at Table 4.1, it becomes apparent that strong performance across a set of diverse datasets is attainable without applying specific fine-tuning on the respective training dataset. An interesting observation is that a larger model size

Table 4.1: Results after coarse-tuning on our eight datasets. The Micro-F_1 score is reported in % on each test dataset before fine-tuning and after coarse-tuning each model on the complete dataset, except BioMedLM, which was only coarse-tuned on the bio-medical domain. We applied LoRA to all models with a size above 3 billion parameters. We do not report the performance of bio-medical coarse-tuned GPT2-XL and Llama-7b variations, as they show worse performances than the general coarse-tuned ones. The table is adapted from our paper [183].

(a) F_1 scores in % on CoNLL-03, CoNLL++, OntoNotes, and Few-NERD.

Model	CoNLL-03	CoNLL++	OntoNotes	Few-NERD
iNERD + ...				
GPT2-XL	89.57	90.54	83.39	50.95
BioMedLM	-	-	-	-
RedPajama	**91.06**	**92.09**	**86.93**	**51.25**
Llama-7b	90.33	91.83	83.19	51.01
Llama-13b	90.88	**92.09**	81.68	50.22

(b) F_1 scores in % on WNUT-17, JNLPBA, NCBI-Disease, and FiNER-ORD.

Model	WNUT-17	JNLPBA	NCBI-Disease	FiNER-ORD
iNERD + ...				
GPT2-XL	43.40	58.49	79.30	75.96
BioMedLM	-	**59.06**	**84.05**	-
RedPajama	**49.03**	57.65	81.17	**80.69**
Llama-7b	43.41	46.70	77.46	74.13
Llama-13b	39.90	57.85	75.37	75.56

does not consistently yield improved performance outcomes. Our largest studied model, the 13-billion parameter version of Llama, can mostly beat its smaller sister, the 7-billion version, but is largely overcome by the drastically smaller RedPajama (3-billion parameter). We theorise that the most likely explanation for this phenomenon is that during coarse-tuning, we apply LoRA to both Llama models to be able to train them in a reasonable time frame, which reduces the number of trainable parameters drastically. Therefore, for datasets with many entity classes $\mathbb{E}$, like Few-NERD and OntoNotes, models with LoRA applied struggle to learn the subtle nuances between different classes and thus fail to outperform smaller models, likely because their available updatable parameter size is simply too small to fit these nuances.

Another insight is that pre-training on a specific domain helps the model during named entity decoding immensely, as shown in the performance of the BioMedLM model. We see this as a vast opportunity for domain-specific pre-training of generative language models to make smaller models usable for the iNERD approach.

Table 4.2: Results after fine-tuning on each specific dataset. The Micro-F_1 score is reported in % on each test dataset after fine-tuning. Each table is divided into two parts. The first shows the performance of iNERD plus a generative language model. The second part shows the performances of various encoder-only approaches. BERT-Base and BERT-Large are taken from [32], BioBERT from [244], PL-Marker from [188], FiNER-LFs from [243], CrossWeigh from [237], and CL-KL from [208]. The table is adapted from our paper [183].

(a) F_1 scores in % on CoNLL-03, CoNLL++, OntoNotes, and Few-NERD.

Model	CoNLL-03	CoNLL++	OntoNotes	Few-NERD
iNERD + ...				
GPT2-XL	91.51	92.71	86.15	51.63
BioMedLM	-	-	-	-
RedPajama	91.06	92.09	**87.71**	**51.81**
Llama-7b	92.75	94.10	84.27	51.72
Llama-13b	**93.09**	**94.21**	84.58	51.13
BERT-Base	92.4	-	-	-
BERT-Large	92.8	-	-	-
BioBERT	-	-	-	-
PL-Marker	**94.0**	-	**91.9**	**70.9**
FiNER-LFs	-	-	-	-
CrossWeigh	93.43	94.28	-	-
CL-KL	93.85	**94.81**	-	-

(b) F_1 scores in % on WNUT-17, JNLPBA, NCBI-Disease, and FiNER-ORD.

Model	WNUT-17	JNLPBA	NCBI-Dis.	FiNER-ORD
iNERD + ...				
GPT2-XL	53.25	58.70	83.79	81.69
BioMedLM	-	**60.08**	**86.37**	-
RedPajama	55.26	59.38	85.75	82.82
Llama-7b	55.59	57.91	80.81	82.42
Llama-13b	**55.76**	59.27	85.07	**83.75**
BERT-Base	-	-	86.37	-
BERT-Large	-	-	-	-
BioBERT	-	**77.59**	**89.71**	-
PL-Marker	-	-	-	-
FiNER-LFs	-	-	-	**79.48**
CrossWeigh	50.03	-	-	-
CL-KL	**60.45**	-	88.96	-

Even though the performances reported are *not* zero-shot, as a small part of the coarse-tuning dataset consists of the training dataset of the respective dataset, this still demonstrates the impressive capabilities of such a model, the coarse-tuning routine, and the iNERD algorithm, as later shown in the ablation study.

Fine-tuning results

After evaluating the iNERD approach on its capabilities after coarse-tuning, we further fine-tune it on each dataset. The results of this can be seen in Table 4.2. Therein, we also report various competing approaches and their performances, taken from the respective papers.

A first observation is that iNERD is capable of performing on par with or better than the standard encoder-only approach reported for the BERT [32] model. A more general observation is that iNERD performs considerably well on datasets with a smaller entity class size, like CoNLL-2003 or NCBI-Disease. For our main focus, the datasets CoNLL-2003 and its corrected version, CoNLL++, iNERD is able to be almost on par with competing state-of-the-art encoder-only approaches [188, 237], which are complex implementations and are thus in stark contrast to our simple and still effective approach.

On the one hand, it struggles especially on Few-NERD and OntoNotes, where the entity class size is significantly larger. Furthermore, the fine variations of various biomedical terms in JNLPBA and the novel entities in WNUT-17 also seem to be a considerable hurdle for our approach. Of course, one could have simply excluded these datasets from this study, but we want to point out fields where our approach is struggling, where it might be improved upon with further research, and therefore, not simply ignore possible drawbacks of our method.

Nevertheless, on the other hand, we surpass the current best-performing model on FiNER-ORD, beating it by a considerable margin of more than 4% F_1 and establishing a new state-of-the-art for financial named entity recognition on this dataset.

In total, the results of our approach are promising for the concept of using generative language models for tasks for which they are not originally intended, as we show that our relatively simple approach can surpass the comparatively simple one proposed in [32].

Ablation study

To show the advantages of each component of our approach, we conduct an ablation study on the CoNLL-2003, CoNLL++, NCBI-Disease, and FiNER-ORD datasets. The results are shown in Table 4.3.

As can be seen there, each component of the iNERD approach adds to the overall performance. If we subtract the coarse-tuning as well as informed decoding steps, the micro-F_1 score falls to a mere 0% in the no fine-tuning setting, similarly

Table 4.3: Ablation study of our iNERD algorithm. Reported here are the micro-F_1 scores in % on the respective test set and the scores when we subtract either the informed decoding algorithm (see Algorithm 1), the coarse-tuning step, or both.

Approach	no fine-tuning	fine-tuning
CoNLL-2003		
iNERD + Llama-7b	90.33	92.75
- informed decoding	86.52	92.43
- coarse-tuning	0.0	91.81
- both	0.0	91.72
CoNLL++		
iNERD + Llama-7b	91.83	94.10
- informed decoding	87.81	93.71
- coarse-tuning	0.0	93.14
- both	0.0	93.01
FiNER-ORD		
iNERD + RedPajama	79.27	82.82
- informed decoding	69.11	81.51
- coarse-tuning	0.0	79.52
- both	0.0	78.74
NCBI-Disease		
iNERD + BioMedLM	84.49	86.37
- informed decoding	83.73	85.83
- coarse-tuning	0.0	85.95
- both	0.0	85.56

when we only exclude the coarse-tuning step. However, such a score can be expected, as the tagging scheme described in Equation (4.1) has not been seen during the model's regular pre-training and thus, is novel to the model. Not so momentous, but still significant, the informed decoding described in Algorithm 1 adds an improvement of approximately 4% for both CoNLL datasets, 2% for FiNER-ORD, and 1% for NCBI-Disease.

A similar, but not so severe, picture can be observed during fine-tuning, where the distance between each subtracted step shrinks, but is still present. In such a setting, we observe an overall improvement of more than 1% for the CoNLL datasets, 4% for FiNER-ORD, and 2% for NCBI-Disease when we compare the complete approach to the one with all components turned off. Additionally, we observe that both techniques introduced here, iNERD and coarse-tuning, are significant improvements to the decoding process.

Limitations

While the iNERD approach introduced in this chapter shows promising results in leveraging LLMs for NER tasks, several limitations should be acknowledged. Firstly, iNERD's effectiveness is inherently tied to the capabilities and limitations of the underlying language model, meaning biases or inaccuracies in the base model could directly impact the algorithm's performance. Similarly, the iNERD algorithm can only be applied when we can access the score vector $\hat{Y}_{\text{score}}$ (see Algorithm 1), which is inaccessible for proprietary closed-source language models like GPT-4 [41] or Gemini [43].

Additionally, while its adaptability has been demonstrated across various datasets, its generalisability to all types of NER tasks cannot be guaranteed, as already shown in Table 4.2. Furthermore, we only tested the performance on English datasets and, thus, cannot make any claims towards a similar performance on NER datasets in other languages.

With the application of coarse-tuning, the comparability to other approaches without coarse-tuning might not be given. This discrepancy arises because our models are exposed to a larger dataset during said coarse-tuning, which might give them an advantage, as also shown in the ablation study.

A further concern is the substantial computational resources required by the LLMs employed. This reliance may limit its accessibility for users with limited computational resources and has already proven to be an issue during hyperparameter tuning, or the lack thereof, in this study. Nevertheless, we focus on demonstrating the efficacy of the approach rather than extensive hyperparameter optimisation. Moreover, this effect is accentuated by the fact that a prompting strategy, as proposed in e.g., [216], does not require task-specific fine-tuning and is able to achieve reasonable results in a zero-shot setting.

The complexity of LLMs also poses challenges regarding transparency and accountability. The lack of clarity in how these models arrive at specific decisions can be problematic, especially in sensitive applications where understanding the reasoning behind an output is crucial.

Another limitation of our approach arises from how the generation process in Equation 4.1 is defined and the fact that it does not specify the position where the entity is located. Nevertheless, we argue that matching the extracted entity string to its position in the input is straightforward through simple string matching. However, in the extremely rare edge case where the entity appears multiple times in the input string, resulting in multiple string matches, iNERD's results may be

less accurate compared to a position-based approach, although we would argue that all string matches in such cases represent the entity.

Lastly, while we can show remarkable results in environments with variable entity class sets $\mathbb{E}$, the algorithm's ability to accurately identify and categorise entirely novel entities is an area that might warrant further exploration.

4.2.4 Conclusion

We introduced a novel approach for named entity recognition (NER) which leverages the outstanding language understanding capabilities of modern large language models (LLMs). Our Informed Named Entity Recognition Decoding (iNERD) algorithm is easy to implement and arguably as simple as an "encoder-only" transformer plus multilayer perceptron approach as proposed in the seminal BERT [32] paper. It builds on top of recent LLMs and is thus future-proof and computationally efficient, as the employed LLMs can easily be replaced by improved models whenever they become available. It incorporates an informed decoding scheme which further improves performance, eliminates any risk of hallucinations, and significantly increases adaptability. This informed scheme utilises the named entity decoding structure proposed herein to mask out disallowed tokens during the prediction phase.

Extensive experimental validation demonstrates the performance of our framework to be mostly on par with competing "encoder-only" approaches, if not better. Experiments further reveal considerable and outstanding adaptive capabilities and show that iNERD can react to changes in the underlying data distribution without any additional fine-tuning. This contrasts with "encoder-only" approaches, which dominate the current NER landscape, as these have to be retrained whenever their set of entity classes changes.

Furthermore, as the generative language models improve, so does our approach. Therefore, we believe that given a few iterations of these models, iNERD plus new versions of generative language models can outpace their competing systems built on smaller "encoder-only" transformer models, due to the indisputable language understanding capabilities of LLMs.

An obvious next step is testing the largest generative language models, like the 70 billion-parameter version of Llama-3 [98] or the Mixture of Experts model Mixtral-8x7b [224], or an architecture different from transformers, like Mamba [245]. Using these could improve the performance of the complete iNERD setup on each dataset even further. On the other hand, training these huge model variations

is extremely expensive and beyond our current computational capabilities. As already discussed in the Experiments section, applying LoRA [229], a method to freeze certain parts of the model to allow training large models, likely leads to a performance decrease. This is yet another interesting path to take for future research, as one could try pre-training large language models without this technique to improve the downstream performance further. Similarly, one could increase the size of the coarse-tuning dataset and include even more datasets.

Another promising starting point for future research is investigating how the various highly specialised named entity recognition techniques developed for encoder-only models like PL-Marker [188] or Co-Regularisation [187] can be applied to generative language models and iNERD. One could also test how regular expressions can be fused with the iNERD algorithm to combine the findings of this work with those from [246], in which they enforced a generation process with such expressions.

Different information extraction tasks like relation extraction or event identification are also candidates for future research, which we plan to tackle in a similar manner as iNERD, as these tasks are "rigid" like NER and would thus profit from an informed approach like the one we propose.

From a more practical standpoint, we plan to implement the iNERD approach in various real-world applications in the world of Financial Auditing and Biomedicine, for the advantages of our approach are clear: highly effective on unseen data with a variable entity set $\mathbb{E}$ (see Table 4.1) and easily upgradeable with the newest large language model.

4.3 LLMs for Legal NER

To contextualise the iNERD approach introduced in the previous section, which established a novel decoding strategy for LLMs, we investigate how they fare *without* such a constrained strategy and how they perform with a simpler prompting framework.

We evaluate general-purpose models to assess their effectiveness in recognising and classifying legal entities. Our research investigates the following questions:

1. How do general-purpose LLMs perform on legal NER tasks?

2. Can LLMs without specific fine-tuning like iNERD handle the archaic language, domain-specific jargon, and intricate syntactic constructions [247, 248] present in legal text?

3. How do different languages affect the performance of multilingual large language models?

4. Which model architectures are most effective for capturing the nuances of legal language in NER tasks?

By addressing these questions, we aim to provide insights into the strengths and limitations of current LLMs in legal NER applications. Specifically, we focus on evaluating eleven state-of-the-art LLMs across multiple datasets, including three English datasets and four datasets in other languages, to establish comprehensive benchmarks for legal NER tasks. The findings of this study are expected to guide future research and development efforts in legal NLP, assisting practitioners in selecting and optimising models for enhanced performance in legal text analysis.

The remainder of this section is structured as follows: we first discuss related work. Section 4.3.2 then describes our methodology, followed by our experiments and results in Section 4.3.3. Finally, we present our conclusion in Section 4.3.4.

4.3.1 Related Work

LLMs have shown remarkable adaptability across various domains, including legal texts, finance, and regulatory compliance, demonstrating their ability to detect inconsistencies, automate tasks, and improve efficiency [44, 129, 135]. Recently, LLMs have emerged as a key force in advancing NER, as demonstrated in [249–253], as well as the previous section.

In the legal domain, LLMs like GPT-4 [41] and Llama-3 [98] have proven highly effective in addressing the complexities of legal texts [13, 254, 255]. These models leverage their advanced language understanding capabilities for tasks such as regulatory compliance verification [44, 256, 257], contract analysis [258], and legal violation detection [259]. A study on legal violation identification demonstrated that LLMs, in addition to BERT-based models, detected legal violations with a high degree of accuracy, outperforming traditional models such as Conditional Random Fields (CRF) in legal contexts [259–261]. These models also handle long-range dependencies and context-rich information more effectively, leading to significant advancements in legal text processing and surpassing classical machine learning approaches [262, 263].

However, applying LLMs to legal NER remains challenging due to domain-specific terminology and intricate syntactic structures. Traditional NER models often underperform in legal contexts, as general language corpora fail to capture

the nuances of legal texts. Domain-specific datasets, such as E-NER [264], LeNER-Br [260], and German-LER [265], have been developed to address these challenges by providing detailed annotations that significantly enhance NER performance across different legal systems and languages [259, 261, 266].

4.3.2 Methodology

This section details our approach to extracting named entities from legal documents and paragraphs.

Dataset Preparation

To evaluate the performance of each LLM, the primary goal is to assign IOB (Inside-Outside-Beginning) [234] tags to each sentence. We explore three strategies for querying LLMs:

1. Directly retrieving IOB tags from the LLM,
2. Obtaining tuples that specify the entity and its start and end positions,
3. Generating a JSON output where entities are represented as keys and their corresponding classifications as values.

The first two methods often lead to considerable inaccuracies. When LLMs return IOB tags directly, they handle tokenisation internally, which frequently diverges from the dataset's ground-truth tokenisation, making the comparison between predicted and actual IOB tags unreliable. Similarly, when returning start and end indices for entities, LLMs struggle to identify the positions precisely, and even slight discrepancies result in a complete misalignment of the IOB tags.

Given these limitations, we adopt the third approach. Rather than relying on the LLM to determine entity positions, we implement a code-based solution to map entity positions within the sentence accurately. This approach ensures a robust alignment with the ground truth, enabling a reliable evaluation of the IOB tagging performance.

Mapping of Entities to Sentence Tokens

To map the model's returned entities to sentence tokens correctly, it is crucial to ensure proper alignment, as this directly impacts the accuracy of predicted IOB tags. Based on the dataset descriptions in relevant research, we replicate their tokenisation methods for consistency and to facilitate proper mapping. For

instance, we used the NLTK [267] library for word tokenisation in E-NER and LeNER-Br, SoMaJo [268] for German-LER, and SpaCy[3] for InLegalNER. For other datasets, where the tokenisation method is either unknown or was completed manually by domain experts, we apply space-based tokenisation. Since these datasets' sentences are pre-tokenised, we query the model with space-separated tokens and tokenise the returned entities similarly, using spaces. By adhering to these tokenisation practices, we ensure accurate mapping of returned entity text to sentence tokens, resulting in reliable IOB tag predictions.

We also compared the consistency of re-tokenised sentences with the originals. In rare cases, discrepancies arise even with identical tokenisation tools. To maintain accuracy in predictions and ground-truth comparisons, we exclude such sentences. Additionally, when the same word belongs to different entities with varied classes, conflicts occur. In these cases, we retain the longer entity text (e.g., a company name containing a person's name) to resolve overlaps, ensuring that the more informative entity is preserved. Furthermore, LLMs sometimes over-identify entities, producing entity classes absent from the dataset. In such cases, we disregard these extra entities to maintain dataset consistency.

Generation of JSON Output

During the generation phase, only three selected models (GPT-4o Mini [269], GPT-4o [41], and Mistral Large[270]) can enforce JSON outputs with their API calls. For models without this capability, we have to rely on the language model to produce a valid JSON output. If the model does not produce a valid JSON output, we evaluate this as if no entities were predicted.

Prompt Design

We design specific prompts for each dataset to ensure accurate entity extraction within distinct legal contexts. While the structure of the prompts remains consistent across datasets, we introduce dataset-specific instructions to optimise performance and address the unique challenges inherent to each. For instance, we emphasise three core requirements: maintaining a valid JSON format, strictly adhering to predefined entity classes, and handling ambiguities by excluding entities that did not match the provided categories.

However, we make tailored adjustments based on each dataset's specific requirements. In the *InLegalNER* dataset, titles or prefixes (e.g., "Mr.", "Sri." are

[3]spacy.io

excluded from annotated entities, so we omit them from the extracted names. In contrast, in the *TurkishLegalNER* dataset, titles (e.g., "Tetkik Hakimi" or "Review Judge") are part of the *Person* class, so we retain them to align with Turkish legal document norms. Similarly, in the *uk_ner_contracts* dataset, when extracting *Clause_Number*, we account for spaces between the number and period, ensuring any following periods are included in the entity. Datasets such as *LegalLensNER* focus on extracting violations and their legal context, thus requiring specific instructions to maintain consistency with legal terminology.

4.3.3 Experiments

In this section, we describe our experimental protocol, examine the datasets and results, and discuss the strengths and limitations of our approach. All model training was performed on a shared GPU node featuring eight Nvidia V100 GPUs, an Intel Xeon 6148 CPU, and 1 TB of RAM.

Data

We use several legal NER datasets in this study. The first three, E-NER, InLegalNER, and LegalLensNER, are English-language datasets covering a range of legal documents. The remaining datasets focus on other languages: LeNER-Br (Portuguese), German-LER (German), TurkishLegalNER (Turkish), and uk-ner-contracts (Ukrainian), each providing domain-specific annotations for their respective legal systems. Below, we shortly detail each of these.

- The **E-NER** [264] dataset consists of 52 filings from the US SEC EDGAR database with manually annotated named entities. For this study, we select the version of the dataset that contains four entity classes: *Person*, *Organisation*, *Location*, and *Miscellaneous*. Since the dataset is not pre-split into training, validation, and test sets, we randomly selected 20% of the data to be used as the test set.

- The **InLegalNER** [261] dataset, designed for legal NER in Indian legal texts, contains 46,545 annotated entities across 14 types, including *court names*, *petitioners*, *respondents*, and *statutes*. Since the dataset does not include IOB format annotations, we converted it by extracting the text and entity details (start/end positions, text, and entity class) for each sentence. Following the approach outlined in their paper, we utilised SpaCy to map entity positions to corresponding words using the "char_span" method, labelling each word

as the beginning (B-), inside (I-), or outside (O) of an entity. Partial manual verification showed that this method resulted in highly accurate mappings.

- The **LegalLensNER** [259] dataset was initially generated by LLMs, but it has been carefully validated by expert annotators to ensure its accuracy and reliability. The dataset includes four main entity types: *Law*, *Violation*, *Violated By* (the entity committing the violation), and *Violated On* (the victim). For our study, we used the entire test set, which consists of 617 sentences. However, one issue we identified with the dataset is the inconsistent labelling of the *Violated By* and *Violated On* entities. In some cases, unnecessary prepositions such as "to" or "on" are included, while in others they are omitted. We believe that excluding these prepositions results in more accurate entity labelling. Instead of modifying the dataset, we ensured that the examples in our prompts avoided such inconsistencies. This choice likely leads to a lower reported performance score but ensures comparability with other approaches.

- The **LeNER-Br** [260] dataset was created for NER in Portuguese, specifically in Brazilian legal texts, containing manually annotated documents from various courts. It includes six entity types: *Pessoa* (persons), *Organização* (organisations), *Local* (locations), *Legislação* (laws), *Jurisprudência* (legal cases), and *Tempo* (time). This comprehensive dataset supports precise NER tasks in Portuguese legal documents. For our study, we used the entire test set, consisting of 1,389 sentences.

- The German Legal Entity Recognition (**German-LER**) [265] dataset is based on German legal texts and contains around 54,000 manually annotated entities. These entities are categorised into two types of classification: fine-grained and coarse-grained semantic classes. For our study, we selected the coarse-grained classification, which includes the following categories: *Person*, *Location*, *Organisation*, *Legal norm*, *Regulation*, *Court decision*, and *Legal literature*. We used the entire test set, which consists of 6,673 sentences, for our experiments.

- The TurkishLegalNER [263] dataset consists of annotated legal texts from the Turkish Court of Cassation, with a total of 2,198 sentences and 5,311 named entities. The dataset includes various entity types relevant to the legal domain, such as *PER* (Person), *LOC* (Location), *ORG* (Organisation), *DAT* (Date), *LEG* (Legislation), *COU* (Court), *REF* (Reference), and *OFF*

(Official Gazette). For our study, we used the test set, which contains 439 sentences. However, due to privacy concerns, many sentences in the dataset contain the redaction "..." to obscure sensitive information. This particularly affects a significant portion of *PER* (Person) entities, as well as some *ORG* (Organisation) and *LOC* (Location) entities. To mitigate this issue, we replace this anonymisation technique with pseudo-anonymisation, i.e., we replace these ellipses with appropriate terms, such as randomly selected Turkish names and locations, ensuring the dataset remains suitable for inference while adhering to privacy regulations.

- The uk_ner_contracts[4] dataset classifies four key types of entities in Ukrainian legal contracts: "Clause_number", "Clause_title", "Contract_type", and "Definition_title". The dataset encompasses a wide range of legal documents across various domains, including employment, real estate, services, sales, and leases. All entities within the contracts have been manually labelled by legal experts, ensuring high-quality annotations. For our study, we used the test set, which consists of 494 sentences.

Evaluation Metrics

For evaluating the performance of LLMs, we employ standard classification metrics based on a strict comparison of predicted IOB sequences with the ground-truth labels. We enforce strict evaluation, requiring both the entity type and boundaries to match the annotations exactly.

We computed precision, recall, and F_1 scores for each entity class to assess the model's performance. For reporting, we focused on the micro-averaged F_1 score, which aggregates the contributions of all classes, providing a balanced view of the LLM's overall accuracy in identifying and classifying entities, without being influenced by class imbalance. All results for the micro-averaged F_1 score were rounded to two decimal places.

Results

The evaluation of LLMs on several legal NER datasets revealed varied performance outcomes. As presented in Table 4.4, F_1 scores for each model fluctuated across the seven datasets, reflecting differences in language and legal context.

GPT-4o [41] demonstrated the highest overall performance, consistently achieving top F_1 scores across multiple datasets. It excelled particularly in

[4]huggingface.co/datasets/lawinsider/uk_ner_contracts

Table 4.4: Few-Shot results for legal NER datasets. Performance is reported as the micro-averaged F_1 score (%). The table is adapted from our paper [14].

(a) English language datasets.

Model	**E-NER**	**InLegalNER**	**LegalLensNER**
Dataset language	*English*	*English*	*English*
GPT-4o Mini [269]	39.53	55.57	46.20
GPT-4o [41]	48.24	**60.56**	49.65
Mistral[270]	37.86	58.39	44.82
Qwen2-72B [271]	**51.62**	51.76	45.19
Llama-3 70B[98]	44.25	59.40	38.28
Llama-3.1 8B [98]	25.44	42.54	35.28
Llama-3.1 70B[98]	40.98	51.46	44.87
Mixtral 8x7B[224]	32.28	33.97	26.00
Gemma-2 27B[272]	42.50	50.69	**50.08**
Phi-3 14B[273]	34.77	41.42	35.90
Phi-3 3.8B[273]	20.16	27.90	31.89

(b) Portuguese, German, Turkish, and Ukrainian datasets.

Model	**leNER-br**	**German-LER**	**TurkishLegalNER**	**uk_ner_contracts**
Dataset language	*Portuguese*	*German*	*Turkish*	*Ukrainian*
GPT-4o Mini [269]	49.47	39.48	60.07	82.42
GPT-4o [41]	**63.88**	57.00	**77.35**	**90.32**
Mistral[270]	51.09	43.33	68.78	88.48
Qwen2-72B [271]	59.01	52.07	70.41	75.00
Llama-3 70B[98]	61.87	**58.84**	66.12	20.14
Llama-3.1 8B [98]	43.31	28.71	23.37	14.48
Llama-3.1 70B[98]	48.95	44.78	52.11	15.77
Mixtral 8x7B[224]	35.70	26.36	20.08	9.54
Gemma-2 27B[272]	52.88	44.16	58.73	87.09
Phi-3 14B[273]	38.25	35.52	20.85	4.94
Phi-3 3.8B[273]	24.65	21.30	11.00	2.65

the *uk_ner_contracts* dataset, achieving an F_1 score of 90.32%, and showed strong results in *InLegalNER* and *German-LER*. Similarly, Qwen2-72B [271] and Mistral [270] achieved competitive outcomes, particularly in *InLegalNER* and *TurkishLegalNER*, underscoring their capability to manage complex legal texts across varied languages and systems.

In contrast, the Llama-3 and Llama-3.1 models [98], which are typically strong performers [186], displayed inconsistent performance. They performed well on *LeNER-Br* with an F_1 score of 61.87% but struggled significantly on *uk_ner_contracts*, where they achieved only 20.14%. This inconsistency suggests that while certain models excel in specific legal domains, their adaptability across datasets remains limited.

A deeper analysis of the *uk_ner_contracts* dataset highlighted specific challenges

affecting model performance. The "Clause_number" entity class is the most prevalent, constituting around 80% of test cases, and requires precise extraction for high accuracy. Clause numbers frequently appear in formats such as "1 ." or "1.1.3 .", where spacing and periods are integral to the entity. Despite clear prompt examples illustrating this format, many models struggled to extract this entity reliably, leading to performance discrepancies. However, models such as GPT-4o and Mistral successfully adapted to these nuances, achieving superior results, which underscores the importance of precise formatting in legal NER tasks.

Furthermore, models like Mixtral 8x7B [224] and Phi-3 14B and 3.8B [273] struggled across most datasets, with notably low scores on *uk_ner_contracts* and *LegalLensNER*. This highlights the challenges that smaller models or certain architectures face when tackling complex legal language and entity structures.

Overall, these results indicate that model architecture, dataset characteristics, and prompt design significantly influence performance in legal NER tasks. While proprietary models like GPT-4o set a high standard, careful tuning and adaptation are crucial for consistent results across diverse legal datasets.

4.3.4 Conclusion

This section presented a comprehensive evaluation of eleven LLMs for NER in the legal domain, covering seven diverse datasets in multiple languages and legal contexts. Our findings demonstrate that LLM performance without a constrained framework like iNERD varies significantly across legal NER tasks, influenced by model architecture and dataset-specific characteristics.

The proprietary GPT-4o [41] model consistently achieved the highest scores, particularly excelling in complex datasets due to its ability to adapt to intricate formatting requirements. Competitive results from models such as Qwen2-72B [271] and Mistral [270] underscore the potential of specialised LLMs to manage multilingual and nuanced legal texts. However, models like Llama-3.1 [98] and smaller architectures, including Mixtral 8x7B [224] and Phi-3 14B [273], showed limitations in handling the syntactic and structural complexities unique to legal language, resulting in inconsistent performance.

Future work could focus on developing adaptive prompt strategies and exploring domain-specific model architectures to advance NER performance further in specialised legal contexts. One could also explore various fine-tuning techniques to better capture the legal language of the datasets we investigated or to improve the prompting strategy employed, as seen in, e.g., [274] or [275].

Overall, this section highlights the substantial potential of LLMs for legal NER tasks, comparing the performances of many state-of-the-art LLMs, showing the brittleness in their performances, and paving the way for other researchers to improve upon the baselines established in this study.

4.4 Conclusion

The work presented in this chapter has engaged directly with a central challenge in modern NLP: how to reconcile the power of generative large language models with the strict, high-precision demands of information extraction. Through two complementary studies, we have explored this issue from different angles, ultimately building a strong case for a hybrid approach to NER. The findings detailed here not only offer a novel and effective method for NER but also contribute to a deeper understanding of where LLMs excel and where they require careful guidance.

Our primary contribution, the **iNERD** framework, was created with a simple but powerful premise in mind: instead of forcing generative models to behave like the classifiers of a previous era, we should align the task with the model's native capabilities. By reformulating NER as a text-to-text generation problem, we directly leverage the deep linguistic knowledge encoded within the LLM's parameters. This conceptual shift allows the model to treat entity extraction as a form of structured translation, a task far more suited to its architecture than token-level classification.

The core innovation of iNERD, however, lies in how it manages this generative process. The informed decoding algorithm is the critical component that instils the reliability necessary for any serious information extraction system. It acts as a set of hard constraints, an overlay that forces the model's probabilistic output to conform to the rigid structure of the NER task. This fusion of a learned, neural system with a symbolic, rule-based one is the essence of the hybrid learning approach introduced in our research. The prevention of hallucinations is the most immediate advantage, moving the system from a "black box" that might produce anything to a predictable tool with a well-defined output space. Furthermore, our results demonstrated that iNERD is not just reliable, but also remarkably powerful, achieving performance that rivals and, in some cases, surpasses highly-specialised encoder-only systems. The success of the "coarse-tuning" strategy further underscored the adaptability of this generative foundation, showing that the

model could generalise effectively to new domains and entity sets, a feat that often requires complete retraining for more brittle, classification-based architectures.

Of course, fine-tuning a model with a method like iNERD is not the only way to apply LLMs to NER. To paint a complete picture, it was essential to investigate the far more common approach of few-shot prompting. Our second study, a comprehensive benchmark on legal NER, served this purpose. The legal domain, with its specialised jargon, complex syntax, and demand for absolute precision, represents a stress test for any language model.

On one hand, the leading proprietary models, particularly GPT-4o, demonstrated a remarkable ability to handle complex instructions and adapt to some of the nuances of multilingual legal text. This confirms that the raw capability for specialised text understanding is indeed present in these models. On the other hand, the overall performance was quite volatile accross models. We observed significant performance swings from one dataset to the next, and even between models from the same family. The difficulties many models had with the strict formatting of clause numbers in Ukrainian contracts, for instance, was a signal that a general understanding of language does not guarantee precision on specific, rule-based patterns. This benchmark study, therefore, serves as a crucial piece of context. It suggests that while prompting is an invaluable tool for rapid prototyping and general-purpose tasks, it may lack the consistency and reliability required for high-stakes, domain-specific information extraction. It highlights the very problem that iNERD was designed to solve.

In viewing this chapter's work as a whole, the two studies function as a compelling argument and its supporting evidence. The legal NER benchmark illustrates the challenges of using LLMs as-is, revealing their inconsistencies when faced with precise, structured tasks. The iNERD framework then serves as an effective response to those challenges. It demonstrates how, with the right methodology, one that embraces the model's generative nature but constrains it with explicit rules, we can build systems that are at once flexible and robust.

An interesting direction for future research lies in adapting specialised techniques from the encoder-only world to the new generative paradigm. For years, the community has developed powerful methods to improve encoder-based NER, such as the packed levitated marker approach of PL-Marker [188] for nested entities or the entity-centric co-regularisation framework proposed by [187], many based on BERT [32] or its variants. These techniques were designed to address specific challenges like overlapping entity spans and leveraging entity-level context. A research question might be how these concepts can be translated into

a generative, decoder-only setting. This might involve creating more complex generation templates that can represent nested structures or integrating novel regularisation terms into the fine-tuning loss function that encourage entity consistency. Successfully porting these advanced techniques could significantly elevate the performance of generative NER, allowing it to tackle even more complex extraction scenarios.

Another promising avenue involves strengthening the hybrid nature of the iNERD algorithm itself. The current informed decoding relies on a fixed set of rules. However, one could explore a more dynamic integration of symbolic knowledge. For example, the work of [246], which uses regular expressions to constrain text generation, could be fused with the iNERD approach. This would allow for the enforcement of highly specific, pattern-based constraints on entity content (e.g., ensuring a "Date" entity conforms to an "YYYY-MM-DD" format) directly within the decoding process. Such a fusion would create an even more powerful and precise extraction tool, combining the semantic understanding of the LLM with the formal rigour of regular languages.

Finally, another research direction is to extend the concept of NER beyond the textual domain into multimodal contexts. As large multimodal models that can process images, audio, and video become more prevalent (see [276, 277]), the task of identifying named entities within these modalities will become increasingly important. For instance, can a model identify a speaker's name from an audio waveform, recognise a company logo in an image, or extract the name of a location from a sign in a video frame? These present fundamental research questions. Can these modalities be processed directly, or must they first be transcribed into text, with NER performed as a secondary step? The former approach is likely far more ambitious, potentially requiring new methods to ground symbolic entity types directly within continuous, non-textual representations. Exploring how a generative framework like iNERD could be adapted for such tasks, perhaps by generating a mixed-modality output of something like `(text_description, bounding_box, entity_type)`, would represent a significant step towards truly comprehensive, multimodal information extraction. This line of inquiry not only pushes the boundaries of NER but also aligns with the broader vision of creating AI systems that can understand and structure information from the real world in all its varied forms.

Die Würde des Menschen ist unantastbar. Sie zu achten und zu schützen ist Verpflichtung aller staatlichen Gewalt.

Human dignity shall be inviolable. To respect and protect it shall be the duty of all state authority.

— Article 1 Paragraph 1 in *Grundgesetz für die Bundesrepublik Deutschland* [278]

5 Anonymisation

Contents

The proliferation of textual data alongside the ascent of powerful Large Language Models has created a dual challenge: while these models offer unprecedented analytical capabilities, their use often requires sharing sensitive information with external, API-based services, and exposes users to privacy risks. This chapter addresses this critical issue by introducing a novel, resource-efficient framework for anonymising textual data across arbitrary domains. Our methodology circumvents the need for manually labelled data and extensive computational power by employing knowledge distillation. We transfer the powerful text understanding of a large "teacher" LLM to a lightweight and efficient encoder-only "student" model, which is then combined with rule-based algorithms for a complete anonymisation pipeline. Our experiments demonstrate that this distilled approach significantly

outperforms established baselines, achieving superior recall and F_1 scores while maintaining a small computational footprint. This efficiency makes our system suitable for deployment on less powerful servers or even local computing devices. Ultimately, this chapter presents knowledge distillation as a scalable and pragmatic pathway for developing robust anonymisation solutions that balance the competing demands of privacy, data utility, and computational feasibility.

5.1 Introduction

This chapter is based on our publications **"Resource-Efficient Anonymization of Textual Data via Knowledge Distillation from Large Language Models"** (co-authored with Max Hahnbück, Tobias Uelwer, Cong Zhao, Christian Bauckhage, and Rafet Sifa) and **"A Survey on Current Trends and Recent Advances in Text Anonymization"** (co-authored with Lorenz Sparrenberg, Armin Berger, Max Hahnbück, Christian Bauckhage, and Rafet Sifa) published in the proceedings of the *31st International Conference on Computational Linguistics* [279] and the *12th IEEE International Conference on Data Science and Advanced Analytics* [280], respectively.

In an increasingly data-driven and AI-influenced world, the need to protect personal and sensitive information has become a critical concern across numerous domains, including, but not limited to, healthcare [281, 282], law [283–285], and finance [286], especially when leveraging large language models (LLMs) [287, 288]. Textual data often contains identifiable information that, if exposed, could lead to privacy violations and data breaches. Such privacy concerns might discourage the use of the most powerful LLMs, which are, at the time of writing, often only accessible via external API requests[1].

Having established robust methods for named entity recognition in Chapter 4, we now turn our attention to one of its practical applications: the protection of sensitive information through anonymisation. While the previous chapter focused on the fundamental challenge of accurately identifying and classifying entities within text, this chapter addresses the equally important question of what we should do with such entities once they have been found. To tackle this, we introduce a hybrid approach and pipeline to anonymise textual data from arbitrary domains. By leveraging knowledge distillation, named entity recognition, and regular expressions, our approach enables the anonymisation of sensitive information in a way that reduces computational overhead while maintaining the semantic integrity of the data. While we evaluate and train on

[1] See the LLM Leaderboard introduced in [186] and hosted at lmarena.ai.

English and German financial documents, our approach can easily be adapted to any new domain or other language. We explore the trade-offs between privacy preservation, model performance, and computational efficiency, demonstrating that knowledge distillation provides a promising pathway for scalable, resource-efficient anonymisation.

Traditional named entity recognition methods, though effective for anonymisation, often present challenges due to their high computational costs [230, 231] or reliance on manually labelled data [187, 188]. The former is problematic because local computational resources may be limited, and using cloud-based solutions may not be feasible, due to the similar reasons that hinder the use of remote LLMs in the first place. The latter poses a challenge because in many domains where state-of-the-art LLMs could offer the most benefit (and thus, require robust anonymisation), labour costs [289] are typically high, making manual data labelling an expensive and time-consuming process.

In this chapter, we shed light on the training pipeline for our anonymisation framework that can take an arbitrary unannotated text corpus and annotation guideline to produce high-quality anonymisation models, that leverage the knowledge and performance of LLMs like GPT-4 [41] or Gemini 2.5 [43] while being so small that they can be deployed on significantly less powerful servers or even conventional personal computing devices.

Our contributions in this chapter can be summarised as follows:

- We demonstrate how a small, lightweight model trained on text annotated by an LLM can be used to solve the underlying named entity recognition (NER) task of anonymisation.
- We build a production-ready anonymisation system that can be deployed either locally or as a service to handle API requests.
- We compare the effectiveness of distilling knowledge from different LLMs and benchmark our anonymisation system against existing solutions, namely Presidio [290] and GLiNER [291].

In the following sections, we start with discussing related work. Section 5.3 describes our methodology for distilling knowledge from a LLM into a smaller model as well as gives an overview on our complete anonymisation pipeline. Thereafter, we outline our experiments and results in Section 5.3. Finally, we end this chapter with a conclusion and an outlook on conceivable future work.

5.2 Related Work

The journey towards automated text anonymisation began with foundational, rule-based approaches that relied on dictionaries and predefined patterns to identify sensitive data [292, 293]. These early systems paved the way for more sophisticated machine learning techniques, with named entity recognition quickly becoming the workhorse of the field [294, 295]. Models based on recurrent neural networks, for instance, were among the first to bring the power of deep learning to the task of finding and flagging personal information in unstructured text [296, 297]. To this day, many practical tools still build on this core idea, often blending modern NER with heuristic rules to create robust, real-world anonymisation pipelines [298, 299].

The arrival of LLMs has been a disruptive force in this landscape, presenting a double-edged sword. On one side, LLMs serve as powerful anonymisers, capable of understanding context with a nuance that allows for highly effective zero-shot or few-shot PII detection [300]. Their general-purpose NER capabilities are so strong that they have inspired a new generation of specialised models, such as NuNER, which is trained on LLM-generated data [189], and highly efficient competitors like GLiNER, which challenges LLMs in zero-shot settings [291]. Yet, on the other side, this same analytical power makes them formidable adversaries in de-anonymisation, capable of inferring identities from subtle clues left behind in poorly anonymised text, a threat that forces the field to constantly re-evaluate its methods [301].

However, the power of these state-of-the-art LLMs comes at a cost. Their size and computational requirements can make them impractical for many real-world applications. This challenge has accelerated research into knowledge distillation, a technique for transferring the deep knowledge of a large "teacher" LLM into a much smaller and more efficient "student" model. This approach has been successfully applied to general NER tasks [274, 275] and offers a pathway to creating anonymisation tools that are both powerful and practical.

Beyond this central challenge of model efficiency, researchers are also tackling a wide array of specialised problems and are developing tailored solutions for the unique demands of specific domains, from the sensitive clinical notes in healthcare [302, 303] to the complex documents found in law [304, 305] and finance [286]. Others are pushing the theoretical boundaries by incorporating formal privacy guarantees like k-anonymity [306] or Differential Privacy (DP), which offers provable protection against data breaches [307, 308]. In a more niche

but equally important area, authorship anonymisation seeks to obscure a writer's unique linguistic style to protect their identity [309, 310].

Ultimately, for any of these techniques to be effective, they must be measurable and usable. This has led to the development of vital community resources, including open-source toolkits like Microsoft's *Presidio* [290] and rigorous evaluation standards like the *Text Anonymization Benchmark* [311]. These tools and benchmarks help researchers and practitioners navigate the fundamental trade-off that defines this field: the balance between protecting privacy and preserving the utility of the data for analysis [312]. This chapter is positioned at this crucial intersection, aiming to harness the power of LLMs in a resource-efficient manner to deliver a practical and effective solution for textual anonymisation.

5.3 Methodology

Our method involves three steps, detailed below:

1. We collect a large number of paragraphs from publicly available documents, which are then pre-processed using traditional methods (Section 5.3.1).
2. We generate training data by prompting large language models, i.e., GPT-4o and GPT-4o mini, to annotate the pre-processed paragraphs (Section 5.3.2).
3. We train a NER model on these annotated paragraphs (Section 5.3.3).

If the performance of step 3 is not satisfactory, we generate more training data by repeating step 2. Results for steps 1 and 2 are stored in a PostgreSQL database [313], whereas the final model of step 3 gets shipped in the form of a containerised environment after hyperparameter tuning has been completed. Figure 5.1 gives an overview of our approach. In the following subsections, we give more details about these three steps.

5.3.1 Data Acquisition

We start by collecting documents from five different sources in English and German. The documents are then split into sentences and subsequently into words to allow for filtering. In detail, we remove sentences that contain an excessive number of special characters or other textual artefacts, as such features suggest the sentence may not have been parsed correctly or may not be a valid sentence. The pre-processed sentences are then stored in a PostgreSQL database to be easily accessible for the following steps.

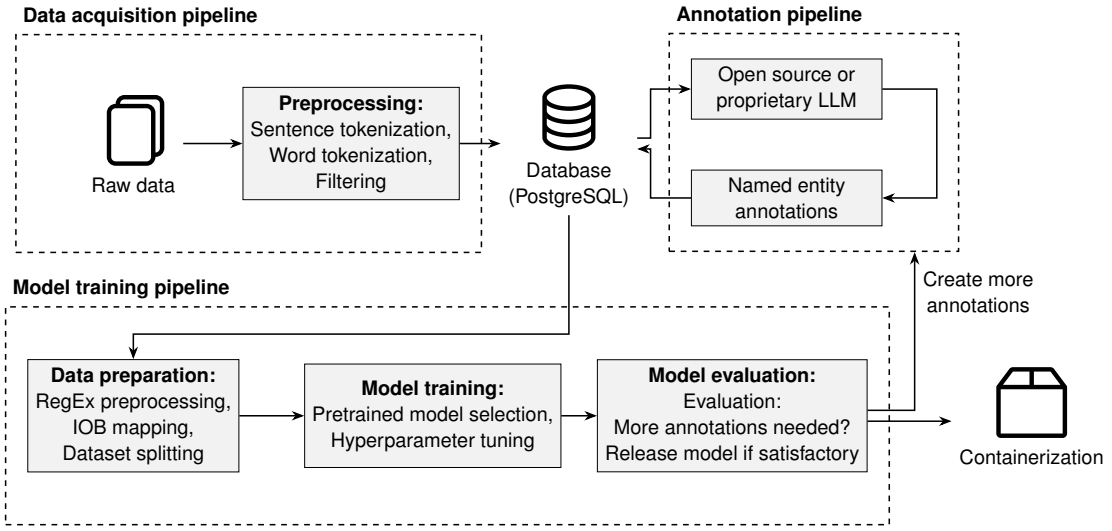


Figure 5.1: The different pipelines for our anonymisation framework. The figure is adapted from our paper [279].

5.3.2 Annotation

A central idea of our approach is to employ an LLM to annotate the collected sentences, thereby generating training data with which to train our lightweight model. We rely on GPT-4o [41] and GPT-4o mini [269], which we prompted using the provided API. However, we also tested Llama-3 70B [98], Mixtral 8x7B [224], and Mistral Large [270], which we found to be inferior to the GPT-4o models.

To find an optimal prompt, we use a comparatively small, annotated dataset composed of approximately 1,000 paragraphs and iteratively improve our prompt until we achieve satisfactory results. In the final prompt, we provide the model with nine different examples of input sentences and their corresponding expected outputs.

For the German datasets, we manually translate the prompt into German and adjust the examples. The annotated paragraphs are stored in the same database from which they were pulled. The entity classes used to train the model described in the following section are shown in Table 5.2. It is important to note that the set of entity classes is flexible and can be defined in advance, allowing customisation for any specific use case.

5.3.3 Model Training

During the model training phase, we first parse the previously created paragraphs and split them into training, validation, and test sets. We then tokenise the text and convert the entity annotations into the Inside-Outside-Beginning (IOB, see

[234]) format so that it can be used in the downstream task. IOB is a tagging scheme used in sequence labelling tasks, where each token in a sentence is tagged as either the beginning (B), inside (I), or outside (O) of a named entity.

The data preparation is followed by the actual training of an *encoder-only* model, e.g., BERT [32] or RoBERTa [55], with a classification head, i.e., a multilayer perceptron, on top. The encoder choice, depth, and layer size of the classification head, and more general model settings are tuneable hyperparameters in this setup.

During training, we leverage the focal loss [4] to allow for better control of how we can weight recall and precision, which is defined as

$$\mathcal{L}_{\text{focal}}(p_e) = -\lambda_e(1 - p_e)^\gamma \ln(p_e), \tag{5.1}$$

where λ_e is used to balance an entity class e, $\gamma \geq 0$ is the focusing parameter of the modulating factor, and $p_e \in [0, 1]$ is the model's estimated probability of entity class e. We theorise that with this loss, we can address the imbalance between the outside and actual entity classes. In an anonymisation framework, it is paramount to identify as many entities as is feasible without penalising precision too much, thus favouring a higher recall over precision. This favours underweighting the outside class, which is overrepresented in anonymisation (and many NER) datasets. To achieve this, we assign a smaller weight to λ_{outside} compared to all λ_{no}, where λ_{no} represents a balancing hyperparameter for all classes other than the outside class.

If we find that the performance after training is insufficient, we generate more annotations using the methodology previously described in Section 5.3.2, followed by repeating the model training step.

5.3.4 Application Development and Deployment

The model trained in Section 5.3.3 is combined with rule-based pre- and post-processing. This processing consists of the optional RegEx-based recognition of monetary values, email addresses, IBANs, phone numbers, and websites. IBANs are validated using `schwifty`[2], and only valid IBANs are anonymised.

The anonymisation model combined with the post-processing discussed above, is exposed as an API via FastAPI[3] and containerised with Docker[4].

[2] schwifty.readthedocs.io
[3] fastapi.tiangolo.com
[4] docker.com

Table 5.1: The datasets and sources we used for training the NER model. The table is adapted from our paper [279].

Name	Language	# Paragraphs	# Annotated paragraphs	Reference
Edgar	English	151k	96k	-
Financial News Articles	English	3.97M	172k	-
Bundesanzeiger	German	415k	38k	[314]
German News	German	201k	40k	[315]
Tagesschau	German	754k	39k	-

Sources:
Edgar: `https://www.sec.gov/search-filings`
Financial News Articles: `https://huggingface.co/datasets/ashraq/financial-news-articles`
Bundesanzeiger: `https://bundesanzeiger.de`
German News: `https://huggingface.co/datasets/community-datasets/gnad10`
Tagesschau: `https://huggingface.co/datasets/bjoernp/tagesschau-2018-2023`

5.4 Experiments

In this section, we describe our experimental protocol, review the data and results, and discuss the key advantages and limitations. All training runs were conducted on a GPU node equipped with eight Nvidia V100 GPUs (each with 32GB of VRAM), an Intel Xeon 6148 CPU, and 1 TB of RAM.

5.4.1 Data

During the data acquisition step, described in Section 5.3.1, we collected roughly 5.5 million paragraphs with a focus on the financial domain. From that pool of raw, unannotated paragraphs, we sampled 385,657 paragraphs, of which 268,756 are English and 116,901 are German, to be annotated with GPT-4o and GPT-4o mini (see Section 5.3.2). Table 5.1 gives an overview of each dataset and Table 5.2 shows all entity classes considered and their respective support in English and German after synthetic annotation. We split our dataset into 80% training data and 10% validation data, which were used for model training and hyperparameter tuning. The remaining 10% was reserved as a hold-out test set, on which we report the results presented in Table 5.3.

5.4.2 Results

When working with synthetic data generation, a key question arises: at what point is the amount of data generated sufficient? To address this, Figure 5.2 illustrates that our validation set performance jumps significantly from zero to approximately 70% after using just 2% of our English dataset, which is roughly five thousand paragraphs. Beyond this point, each additional paragraph

Table 5.2: Entity classes in our dataset and their support in English (en) and German (de). The table is adapted from our paper [279].

Label	Description	Support en	Support de
<PER>	Person	75,433	28,498
<LOC>	Location	95,538	41,799
<ORG>	Organisation	159,434	36,857
<PROD>	Product	20,865	4,603
<DATE>	Date or time	113,876	27,418
<MISC>	Miscellaneous	216,871	91,050

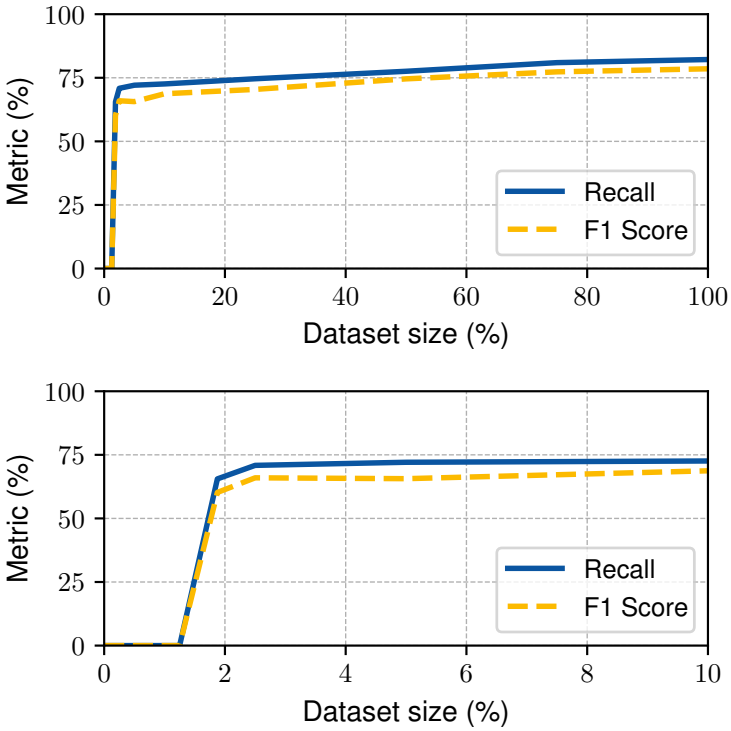


Figure 5.2: Diminishing Effect of dataset size on model performance. The underlying dataset is the English split of our data, as described in Table 5.1, totalling 268 thousand paragraphs. Note that both graphs show the same data, only with a differently scaled X-axis. The figure is adapted from our paper [279].

yields diminishing returns, and the performance plateaus when approximately 80% of the dataset is utilised.

Table 5.3 shows the results of our experiments on the test set. We test four different configurations of our *Anonymiser* system, each with a different pre-trained encoder backbone and various total model sizes. Our framework comfortably outperforms the two baselines, Presidio [290] and GLiNER [291].

As expected, larger models tend to exhibit superior performance. Nevertheless,

even our smaller models, with fewer than 200 million parameters, demonstrate satisfactory performance. Based on these findings, we propose a clear deployment strategy: smaller models are well-suited for on-device deployment due to their efficiency, while larger models, given their superior performance, are better positioned for server-based deployment.

Furthermore, we can observe that leveraging the focal loss [4] described in Equation 5.1 achieves our goal of favouring recall while keeping the overall F_1 score high, which is of significant importance when anonymising data.

5.4.3 Limitations

The "Miscellaneous" (MISC) category poses a unique challenge due to its highly heterogeneous nature. It serves as a catch-all for tokens that do not fit into other predefined categories, leading to a mix of relevant and irrelevant data, stemming from its definition: "Miscellaneous encompasses any significant information not covered by the other categories that might be used to de-anonymise". This lack of clear boundaries makes it difficult for the model to identify which tokens belong to this class consistently. Although dividing the MISC category into more detailed sub-categories might be possible, some tokens will always resist clear classification. Additionally, classification is subjective, depending on the user's context and model application. Despite these challenges, we have chosen to retain the MISC category in our six-class schema for its balance of manageability and relevance.

This ambiguous nature is illustrated by the following example sentence from the *financial-news-articles* dataset, with annotations below each entity:

"Francisco Palmieri (PER), acting Assistant Secretary of State (MISC) for Western Hemisphere Affairs (MISC), said the Cuban (LOC) government was responsible for the security of U.S. (LOC) diplomatic personnel on the island (MISC) 'and they have failed to live up to that responsibility.' Asked whether it was possible that the Cuban government (ORG) would have been unaware of any attacks, he said: 'I find it very difficult to believe that.'"

The entities tagged as MISC illustrate the ambiguous nature of this class, highlighting the difficulty for models to learn this entity class. One could

Table 5.3: Results on the hold-out test set. Ano N, S, R, and L refers to our *Anonymiser* framework, as described in Section 5.3, with different encoder models. The number following the model identifier is the corresponding total model parameter count. Ano S and L feature the GLiNER model variant (*only* the actual, raw transformer model without the classification head) introduced by [316] in the respective small and large size, whereas Ano R represents the setup with a RoBERTa-Large previously finetuned with the OntoNotes dataset [238] introduced by [317] and Ano N has the NuNER-v2.0 model [189] as its encoder. Each setup was subjected to hyperparameter tuning on the validation set before being evaluated on the test set. We add results from Presidio [290] and GLiNER [291] as a baseline. Note that Presidio only supports anonymising persons, locations and dates out-of-the-box. The table is adapted from our paper [279].

(a) F_1 Scores in %

	Model	Person	Location	Org.	Product	Date	Misc.	Micro avg.	excl. Misc.
English	Presidio	74.39	66.59	-	-	52.62	-	39.01	48.97
	GLiNER	68.40	62.85	60.62	12.17	75.87	03.89	51.20	61.54
	Ano N 146M	93.61	90.90	87.88	62.74	86.52	54.63	77.69	87.95
	Ano S 163M	93.49	90.34	88.37	59.84	85.40	52.22	77.28	87.58
	Ano R 377M	93.51	91.11	88.69	64.89	87.31	55.03	78.07	88.63
	Ano L 456M	**94.47**	**91.45**	**89.33**	**66.60**	**87.82**	**55.47**	**78.98**	**89.32**
German	Presidio	06.11	25.94	-	-	41.81	-	11.83	13.94
	GLiNER	60.41	65.49	47.65	23.33	68.39	04.35	45.48	56.70
	Ano N 146M	87.11	84.48	79.62	55.82	88.82	49.36	69.71	83.60
	Ano S 163M	88.05	86.13	80.96	55.58	88.37	47.11	70.20	84.58
	Ano R 377M	89.00	86.58	82.38	60.58	89.53	49.24	70.86	85.77
	Ano L 456M	**92.62**	**89.84**	**85.69**	**68.19**	**93.57**	**53.50**	**74.43**	**89.33**
English & German	Presidio	30.69	50.34	-	-	51.02	-	29.05	35.62
	GLiNER	66.86	63.50	57.29	21.09	74.44	04.03	49.71	56.64
	Ano N 146M	91.20	88.95	86.48	62.14	87.29	52.77	75.90	87.24
	Ano S 163M	92.68	89.84	87.89	63.11	88.27	54.10	76.82	88.18
	Ano R 377M	**92.80**	**90.26**	**88.41**	**64.67**	88.21	54.05	76.62	**88.62**
	Ano L 456M	92.69	90.10	88.37	63.21	**88.49**	**55.55**	**77.78**	88.51

(b) Recall Scores in %

	Model	Person	Location	Org.	Product	Date	Misc.	Micro avg.	excl. Misc.
English	Presidio	78.95	70.62	-	-	67.16	-	30.14	43.97
	GLiNER	91.37	78.04	85.74	52.80	76.49	02.45	56.54	81.35
	Ano N 146M	95.40	**94.47**	91.95	**65.64**	89.12	54.54	79.58	91.15
	Ano S 163M	95.73	93.16	90.91	63.95	88.77	48.49	77.29	90.46
	Ano R 377M	95.23	93.58	**92.21**	61.97	89.30	**56.00**	79.88	90.91
	Ano L 456M	**96.02**	94.35	91.68	64.84	**89.99**	55.94	**80.45**	**91.39**
German	Presidio	31.61	41.16	-	-	33.36	-	15.46	25.60
	GLiNER	86.65	79.52	79.33	61.96	75.32	02.54	49.01	79.49
	Ano N 146M	88.54	86.96	84.20	59.96	90.67	**52.07**	72.65	86.41
	Ano S 163M	92.12	89.98	83.01	59.00	90.61	46.37	71.32	87.69
	Ano R 377M	89.59	87.68	84.17	57.30	90.25	51.34	72.51	86.67
	Ano L 456M	**92.83**	**91.45**	**85.54**	**66.48**	**93.76**	50.60	**72.85**	**89.66**
English & German	Presidio	65.39	62.02	-	-	60.29	-	26.38	39.69
	GLiNER	89.75	78.87	84.00	53.27	76.42	02.48	54.49	80.71
	Ano N 146M	94.82	91.53	91.49	62.72	90.43	52.38	77.69	90.68
	Ano S 163M	94.31	92.23	89.75	63.15	89.82	54.82	78.19	89.94
	Ano R 377M	94.87	92.70	89.94	65.15	**90.76**	**57.19**	79.29	90.63
	Ano L 456M	**95.38**	**93.95**	**92.21**	**68.52**	90.51	54.95	**79.44**	**91.83**

also argue that they may not necessarily require anonymisation, as they lack definitive identifying information.

Another limitation of our approach is the requirement to train a model. Other approaches incorporating large language models or solutions like GLiNER [291] or Presidio [290] are designed to function in a zero-shot environment without any additional training. Nevertheless, such solutions are either computationally intensive, accessible only via an API, and/or exhibit weaker performance (see Table 5.3).

5.5 Conclusion

We have introduced a novel text anonymisation approach that balances privacy preservation with computational efficiency by distilling knowledge from large language models into smaller, encoder-only models using named-entity recognition and rule-based algorithms. Our lightweight system operates without the need for manually labelled data or extensive computational resources and is suitable for deployment on less powerful servers or personal computing devices. It can easily be adapted to any domain and is currently deployed for the anonymisation of financial documents and texts.

Our experiments demonstrate that our method outperforms existing solutions like GLiNER [291] or Presidio [290], achieving higher F_1-scores and, more importantly, higher recall overall and in all entity classes. Even our smaller models with fewer than 200 million parameters showed satisfactory and superior performance, indicating their practicality for on-device deployment where computational resources are limited and anonymisation is paramount.

In conclusion, our findings suggest that knowledge distillation offers a scalable, customisable, and resource-efficient pathway for text anonymisation. By harnessing the capabilities of LLMs, our approach holds significant promise for enhancing privacy preservation in textual data across various domains. Furthermore, with the continuous development of new LLMs, we can enhance our framework by updating the teacher, i.e., the LLM, of our NER models.

Future work could shift the focus from the financial domain to different languages or domains, such as social media, healthcare, or law, which require a different set of entities but can likely be solved with the same framework as introduced here. Additionally, one could test whether a performance degradation is observed after replacing the raw, real-world data (see Section 5.3.1 and 5.4.1) with synthetic data generated by an LLM, as seen, for example, in [318]. Another

interesting avenue to explore is the effect anonymisation has on the performance of LLM-powered downstream tasks like contradiction detection [129], factual consistency evaluation [319], or automated regulatory compliance verification [44], or on the direct, actual performance of LLMs, as evaluated by benchmarks like the Open LLM Leaderboard [320].

5.5.1 Ethical Considerations

Our work focuses on enhancing privacy by anonymising sensitive information in textual data across various domains. While our approach aims to protect personal data and mitigate the risk of privacy breaches, it is important to acknowledge that no anonymisation method, including manual anonymisation, can provide a 100% guarantee of complete confidentiality, and our method is no exception, as shown in Table 5.3.

Additionally, the opposite is also possible: if one applies the same approach as the one in our model, the identification of sensitive information and entities from arbitrary chunks of text could lead to the easier retrieval of said personal information, which is an inherent risk of all named-entity recognition models.

Got to love computers. They do all the thinking for you so you don't have to.

— Andy Weir in *Project Hail Mary* [321]

6 Relation Extraction

Contents

We present *KPI-BERT*, an end-to-end system that combines a Bidirectional Encoder Representations from Transformers (BERT) [32] backbone with a recurrent neural network (RNN) and conditional label masking to sequentially tag entities and classify their relations to automatise the extraction of Key Performance Indicators (KPIs) from financial documents. Building on these efforts, we introduce *KPI-EDGAR*, a novel open-source dataset of financial documents uploaded to the Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system, which provides a novel joint Named Entity Recognition (NER) and Relation Extraction (RE) benchmark open to all researchers. Additionally, we propose a new, word-level weighting scheme that refines the conventional F_1 score, better capturing the inherently fuzzy

boundaries of entity pairs in financial texts. Next, we investigate the impact of introducing controlled noise into the fine-tuning process of pre-trained language models to improve performance on joint NER and RE tasks, demonstrating how targeted perturbations can yield favourable gains on the *KPI-EDGAR* dataset. Finally, we extend our exploration to Large Language Models (LLMs), examining their potential for tackling KPI extraction without additional fine-tuning. We evaluate both proprietary and open-source LLMs for their suitability in linking KPIs to corresponding values and attributes, highlighting the technical challenges and substantial opportunities these models present. Collectively, these studies illustrate the evolution of KPI extraction techniques in financial domains, from tailored neural architectures to zero-shot LLM-based approaches, illuminating the path towards more efficient and accurate information extraction pipelines.

6.1 Introduction

This chapter is based on our publications **"KPI-BERT: A Joint Named Entity Recognition and Relation Extraction Model for Financial Reports"** (co-authored with Lars Hillebrand, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa), **"KPI-EDGAR: A Novel Dataset and Accompanying Metric for Relation Extraction from Financial Documents"** (co-authored with Syed Musharraf Ali, Lars Hillebrand, Desiana Nurchalifah, Basil Jacob, Christian Bauckhage, and Rafet Sifa), **"Controlled Randomness Improves the Performance of Transformer Models"** (co-authored with Cong Zhao, Wolfgang Krämer, David Leonhard, Christian Bauckhage, and Rafet Sifa), and **"Leveraging Large Language Models for Few-Shot KPI Extraction from Financial Reports"** (co-authored with Cong Zhao, Daniel Uedelhoven, Lorenz Sparrenberg, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa). These were published in the proceedings of the *26th International Conference on Pattern Recognition* [11], the *21st and 22nd IEEE International Conference on Machine Learning and Applications* [12, 322] and the *12th IEEE International Conference on Big Data* [323], respectively.

Building on the hybrid mechanisms for representation learning explored in the contradiction detection and named entity recognition chapters, here we investigate how hybrid architectures can be used to find structured relationships within financial documents. This relation extraction task is, like the anonymisation task introduced in the previous chapter, an extension to named entity recognition, and it provides an ideal proving ground for the hybrid paradigm, as it requires models to jointly reason about multiple entities and their interdependencies, as seen later in this chapter. Here, we explore how key performance indicators

(KPIs) can be extracted and connected by combining contextual embeddings with recurrent layers and regularisation strategies. In doing so, this chapter reinforces the central argument of the thesis: that hybrid representation learning yields information extraction models that are simultaneously data-efficient, interpretable, and domain-adaptable.

In the context of business administration and accounting, KPIs, which are defined as "vital navigation instruments used by managers to understand whether their business is on a successful voyage or whether it is veering off the prosperous path" [324], are typically understood to be quantitative measures that facilitate descriptive, comparative, and predictive analyses as well as informed decision-making [325, 326]. Examples include values such as *revenue*, *interest expenses*, *profit*, and *loss*, which feature prominently in financial documents including income statements, business plans, or *10-K* annual reports. Extracting and linking such indicators in an automated fashion can give businesses significant competitive advantages by reducing the time needed for investors, analysts, or auditors to parse large volumes of text [164].

6.1.1 Motivation and Context

To illustrate how the tasks of joint NER and RE allow the identification of KPIs and their corresponding numerical value in financial texts, consider the following example:

> "In 1971, the Khazad-dûm Mining Company increased its revenue$_{\text{KPI}}$ to \$2.5$_{\text{CY}}$ billion (prior year: \$2.1$_{\text{PY}}$ billion) while the total operating costs$_{\text{KPI}}$ decreased to \$1.3$_{\text{CY}}$ billion (prior year: \$1.5$_{\text{PY}}$ billion)."

In this snippet, the model is expected to identify the two KPIs (KPI) and link them appropriately with their associated *current year* (CY) and *prior year* (PY) values. The desired outcome is:

> revenue$_{\text{KPI}}$ – 2.5$_{\text{CY}}$, revenue$_{\text{KPI}}$ – 2.1$_{\text{PY}}$, total costs$_{\text{KPI}}$ – 1.3$_{\text{CY}}$, total operating costs$_{\text{KPI}}$ – 1.5$_{\text{PY}}$

However, the automatic extraction of KPI-related information from unstructured textual data poses considerable challenges. Several approaches to extracting structured data from financial documents have been explored, although these have often been rule-based and inflexible [327], limited to tabular data [326], or focused exclusively on numerical checks [133].

6.1.2 Our Contributions

In order to address the above-mentioned limitations, we introduce multiple approaches for extracting KPIs and their relationships from both German and English financial reports using state-of-the-art natural language processing (NLP) techniques for joint NER and RE. Our research on this topic has been presented in four main publications, covering:

1. **KPI-BERT** [11]: An end-to-end architecture combining a BERT [32] backbone with an RNN and conditional label masking to sequentially tag entities and classify their relations, aimed at automating KPI extraction in financial documents. We compare our system against multiple baselines, which also build on BERT-encoded word embeddings but utilise different entity tagging schemes, namely span-based tagging [328], sequential Conditional Random Field (CRF) tagging [329] and standard linear tagging [330].

2. **KPI-EDGAR** [12]: A publicly available dataset of *10-K* financial reports from EDGAR, coupled with a refined, word-level weighting scheme for the conventional F_1 metric to better reflect the partial correctness and fuzzy borders of predicted and ground-truth relations.

3. **Controlled Randomness in Fine-Tuning** [322]: A study examining how injecting targeted noise into pre-trained language models to avoid overfitting and instability during the fine-tuning process can improve the downstream performance, leading to improved results on the KPI-EDGAR dataset.

4. **LLMs for KPI-Extraction** [323]: An evaluation of both proprietary and open-source LLMs in a zero-shot or few-shot setting, highlighting their potential and limitations for extracting KPIs without additional fine-tuning and comparing them to our findings of the previous studies.

Collectively, our studies illustrate the evolution of automated KPI extraction in the financial domain. Starting from a specialised BERT-based system for German financial reports, our investigation extended to an open-source English dataset with a novel metric for partial correctness, the introduction of controlled noise to stabilise training, and finally, the move towards LLMs that can operate in a zero-shot setting. Our studies highlight both how traditional neural architectures for representation learning and classification can be tailored to solve domain-specific challenges with high accuracy, how recent advances in language modelling

may enable more flexible and efficient approaches, and how more "traditional" representation learning, supervised fine-tuning, and comparatively smaller RE systems fare against state-of-the-art, zero-shot LLM-enabled relation extraction. In doing so, we provide a comprehensive examination of the methodologies, practical challenges, and potential future directions in extracting KPIs and their associated relations from a variety of financial documents.

6.2 Related Work

Research in the area of automatic information extraction from text has frequently focused on either NER, RE, or the *joint* learning of both tasks. Early studies, such as [331] and [332], investigated hierarchically combining NER and RE in a pipeline and showed empirically that information extracted in one task can benefit the other. More recent work has reinforced the advantages of formulating them jointly rather than as standalone processes (e.g., [188, 328, 333–343]). In particular, many modern approaches leverage transformer-based language models like BERT [32] or RoBERTa [55] to encode contextual information and improve overall performance on NER and RE tasks. Further enhancements include strategies for reducing annotation efforts [335], making models more compact [334], or applying more advanced architectures such as graph neural networks [344], multi-head selectors [345, 346], or triple classification [347]. [348] and [349] fine-tuned LLMs to do few-shot extraction of relations in the legal and scientific domain, respectively. For a more thorough review of the latest advances, we refer to recent survey papers on joint NER and RE [350, 351].

Within the realm of financial texts, several lines of research have adapted these techniques to extract key financial data. [352] utilised MLPs to build interpretable accounting structures, albeit with pre-processed accounting variables rather than fully unstructured text. [327] proposed a rule-based NER approach for financial documents, and [326] introduced a hybrid system that recognises paragraphs and tables before extracting targeted key performance indicators (KPIs). More recent work has further demonstrated the benefits of transformer-based architectures in handling domain-specific subtasks such as cross-checking financial formulas [133], anonymising sensitive content [279, 353], or embedding auditing functionalities into software tools [164].

Similar attempts at regularisation during neural network training as done in our work [322], such as weight decay [174], dropout [354], or smart scheduling of learning rates [355], have been proposed to mitigate overfitting. Although

these approaches have historically been applied broadly in deep learning [356, 357], the explicit injection of noise into certain parts of pre-trained language models [358] has only recently been explored.

In summary, the field of joint NER and RE has benefited greatly from modern transformer architectures, and the financial domain provides a valuable use case for demonstrating the real-world impact of these methods. Ongoing developments in regularisation and zero-shot approaches promise further improvements in performance, stability, and breadth of applicability, ultimately paving the way for more automated and reliable data extraction pipelines in finance.

6.3 Methodology

In this section, we present the approaches corresponding to each contribution outlined in Section 6.1.2. First, we introduce KPI-BERT, our BERT-based [32] framework for joint NER and RE. We then propose a refined F_1 metric designed to capture the fuzzy boundaries common in financial texts. Next, we examine how targeted noise injection into the model parameters can act as an effective regularisation mechanism. Finally, we show how LLMs can tackle the same tasks addressed by KPI-BERT.

6.3.1 KPI-BERT

Our joint model for NER and RE, named *KPI-BERT*, comprises three interconnected modules, trained end-to-end via gradient descent [359]. First, a BERT-based encoder transforms the input sentence into a latent representation. Next, a Gated Recurrent Unit (GRU) [5] decoder employs conditional label masking and its tagging history to sequentially identify entities. Finally, a relation extraction (RE) decoder links the predicted entities to form relations.

BERT-based Sentence Representation

Given a WordPiece [110] tokenised input sentence consisting of n subwords, we use a pre-trained BERT model to obtain a sequence of $n+1$ encoded token representations, $(\mathbf{r}_c^{\mathcal{E}}, \mathbf{r}_1^{\mathcal{E}}, \mathbf{r}_2^{\mathcal{E}}, \ldots, \mathbf{r}_n^{\mathcal{E}})$. Here, $\mathbf{r}_c^{\mathcal{E}} \in \mathbb{R}^{d_{\mathcal{E}}}$ serves as a context embedding for the entire sentence, $\mathbf{r}_i^{\mathcal{E}} \in \mathbb{R}^{d_{\mathcal{E}}}$ represents the embedding of the i-th token, and $d_{\mathcal{E}}$ is the dimensionality of the chosen encoder $\mathcal{E}$, i.e., the pre-trained BERT model. To reduce model complexity and align with word-level annotations, we apply a pooling function, $\mathcal{P}(\cdot)$, that aggregates the subword embeddings corresponding

to each word. Specifically, if the j-th word w_j consists of m subwords, then we define the representation $\mathbf{r}_j^{\mathrm{w}}$ of this word as:

$$\mathbf{r}_j^{\mathrm{w}} := \mathcal{P}(\mathbf{r}_i^{\mathcal{E}}, \mathbf{r}_{i+1}^{\mathcal{E}}, \ldots, \mathbf{r}_{i+m-1}^{\mathcal{E}}), \quad \mathbf{r}_j^{\mathrm{w}} \in \mathbb{R}^{d_{\mathcal{E}}}. \tag{6.1}$$

While we also evaluate simple max- and mean-pooling methods, we employ a more sophisticated trainable RNN-pooling mechanism built on a bidirectional GRU. In particular, the subword embedding sequence $(\mathbf{r}_i^{\mathcal{E}}, \mathbf{r}_{i+1}^{\mathcal{E}}, \ldots, \mathbf{r}_{i+m-1}^{\mathcal{E}})$ is processed by forward and backward GRUs, $\mathcal{G}^{\text{forward}}$ and $\mathcal{G}^{\text{backward}}$, yielding the final hidden states

$$\mathbf{h}_j^{\text{forward}} = \mathcal{G}^{\text{forward}}\big(\mathbf{r}_i^{\mathcal{E}}, \mathbf{r}_{i+1}^{\mathcal{E}} \ldots, \mathbf{r}_{i+m-1}^{\mathcal{E}}\big), \tag{6.2}$$

$$\mathbf{h}_j^{\text{backward}} = \mathcal{G}^{\text{backward}}\big(\mathbf{r}_{i+m-1}^{\mathcal{E}}, \mathbf{r}_{i+m-2}^{\mathcal{E}}, \ldots, \mathbf{r}_i^{\mathcal{E}}\big), \tag{6.3}$$

where $\mathbf{h}_j^{\text{forward}}, \mathbf{h}_j^{\text{backward}} \in \mathbb{R}^{d_{\mathcal{E}}/2}$. Finally, we concatenate these two hidden states to obtain the representation of the j-th word,

$$\mathbf{r}_j^{\mathrm{w}} = \mathbf{h}_j^{\text{forward}} \oplus \mathbf{h}_j^{\text{backward}}. \tag{6.4}$$

NER Decoder

By utilising the BERT-encoded and the pooled word embedding sequence $(\mathbf{r}_1^{\mathrm{w}}, \mathbf{r}_2^{\mathrm{w}}, \ldots, \mathbf{r}_n^{\mathrm{w}})$, we employ a NER decoder to classify named entities within the sentence. Specifically, we introduce a sequential GRU-based tagger with conditional label masking, based on the IOBES annotation scheme, which extends the IOB scheme introduced by [234]. In this scheme, each word is tagged with a prefix among *I*- (inside), *B*- (begin), *E*- (end), or *S*- (single), while *O* (outside) represents words not belonging to any entity. If we denote $\mathbb{E}_{\text{entities}}$ as the set of possible entity types, including the *none* class, as detailed in Table 6.2, the actual number of IOBES entity tags is

$$|\mathbb{E}_{\text{IOBES}}| = 4\,(|\mathbb{E}_{\text{entities}}| - 1) + 1. \tag{6.5}$$

To illustrate the IOBES tagging scheme in detail, consider the following example:

> "The total net revenue of Rohan Express Couriers was $3.14 billion this year and $2.72 billion last year."

The corresponding IOBES label sequence should be:

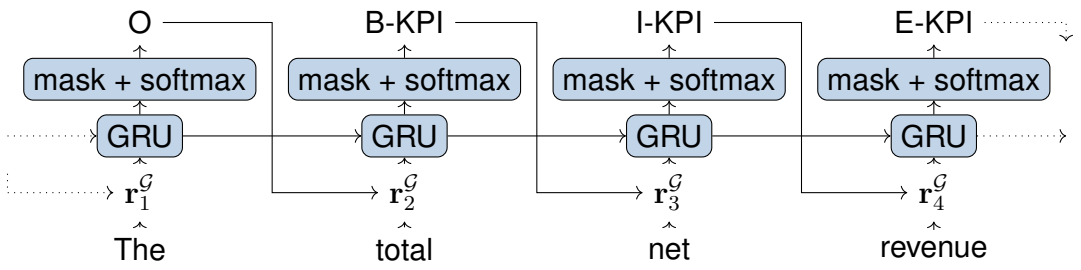


Figure 6.1: Sequential IOBES tagging (each entity class is prepended with the prefixes *I-* (inside), *B-* (begin), *E-* (end) or *S-* (single), while *O* (outside) represents the *none* class) leveraging a gated recurrent unit (GRU) and conditional label masking. $\mathbf{r}_j^{\mathcal{G}}$ represents the concatenation of the previously predicted label embedding with the word embedding at position j. Along with the previous hidden state vector it gets passed to a GRU that, followed by label masking and a softmax layer, predicts the next IOBES tag. The figure is adapted from our paper [11].

"O, B-KPI, I-KPI, E-KPI, O, O, O, O, O, O, S-CY, O, O, O, O, O, S-PY, O, O, O, O".

To account for the sequential nature of entity tagging, we use a GRU in combination with conditional label masking to sequentially predict IOBES tags, taking into consideration previously predicted tags. This decoding process is visualised in Figure 6.1 and described below.

First, we define a matrix $\mathbf{R}_{\text{label}} \in \mathbb{R}^{|\mathbb{E}_{\text{IOBES}}| \times d_{\text{IOBES}}}$ which contains learnable d_{IOBES}-dimensional representations of every IOBES label.

Second, we concatenate the pooled word embedding $\mathbf{r}_j^{\text{w}}$ with the embedding of the previously predicted tag $\mathbf{r}_{j-1}^{\text{label}}$, which yields the decoding input representation of word j,

$$\mathbf{r}_j^{\mathcal{G}} = \mathbf{r}_j^{\text{w}} \oplus \mathbf{r}_{j-1}^{\text{label}}, \tag{6.6}$$

where $\mathbf{r}_{j-1}^{\text{label}} \in \mathbb{R}^{d_{\text{IOBES}}}$ represents the label embedding of the previously predicted IOBES tag. We set $\mathbf{r}_0^{\text{label}}$ to the embedding of the *O* label, as using a dedicated beginning-of-sequence embedding did not improve results empirically.

Third, we feed $\mathbf{r}_j^{\mathcal{G}}$ alongside the previous hidden state $\mathbf{h}_{j-1}$ into a GRU $\mathcal{G}^{\text{NER}}$, obtaining the hidden state

$$\mathbf{h}_j = \mathcal{G}^{\text{NER}}\left(\mathbf{r}_j^{\mathcal{G}}, \mathbf{h}_{j-1}\right). \tag{6.7}$$

To generate the IOBES tag prediction for word j, we linearly transform $\mathbf{h}_j$, then apply a masking function $\mathcal{Z}$ followed by a softmax operation:

$$\hat{\mathbf{y}}_j = \text{softmax}\left(\mathcal{Z}\left(\mathbf{W}_{\text{lin}}\mathbf{h}_j + \mathbf{a}_{\text{lin}}\right)\right), \tag{6.8}$$

Table 6.1: Comprehensive overview of all allowed relations and their uniqueness. "`1:1`": One entity of type 1 can only be linked to one entity of type 2, "`1:n`": One entity of type 1 can be linked to many entities of type 2. "`-`": No relation possible. The table is adapted from our paper [11].

	KPI	CY	PY	INCREASE	DECREASE	DAVON	DAVON-CY	DAVON-PY
KPI	-	1:1	1:1	1:1	1:1	1:n	-	-
CY	1:1	-	-	-	-	-	-	-
PY	1:1	-	-	-	-	-	-	-
INCREASE	1:1	-	-	-	-	-	-	-
DECREASE	1:1	-	-	-	-	-	-	-
DAVON	n:1	-	-	-	-	-	1:1	1:1
DAVON-CY	-	-	-	-	-	1:1	-	-
DAVON-PY	-	-	-	-	-	1:1	-	-

where $\mathbf{W}_{\text{lin}}$ and $\mathbf{a}_{\text{lin}}$ are learnable parameters of the linear transformation. The masking function $\mathcal{Z}$ rules out invalid IOBES tag predictions based on the previously predicted label. For instance, if $\arg\max(\hat{\mathbf{y}}_{j-1})$ is *O* or has the prefix *S* or *E*, then all *I*- and *E*- labels are masked out for the current prediction. Conversely, if $\arg\max(\hat{\mathbf{y}}_{j-1})$ begins with *B* or *I*, the next tag must continue the same entity type with *I*- or *E*-, effectively reducing the decision to binary classification in that specific step.

Next, we map the predicted IOBES tags and corresponding word embeddings $\mathbf{r}_j^{\text{w}}$ to entity-level spans by applying the same pooling function as described in Equation (6.1). Finally, we concatenate these pooled entity representations with a span size embedding $\mathbf{r}_m^{\text{width}}$. This embedding is taken from a dedicated embedding matrix $\mathbf{R}_{\text{width}} \in \mathbb{R}^{\ell \times d_{\text{width}}}$ of fixed-size representations of dimensionality d_{width} for span lengths from 1 to ℓ. Specifically, the representation for an entity e of width m is:

$$\mathbf{r}_e := \mathcal{P}(\mathbf{r}_j^{\text{w}}, \mathbf{r}_{j+1}^{\text{w}}, \ldots, \mathbf{r}_{j+m-1}^{\text{w}}) \oplus \mathbf{r}_m^{\text{width}}, \tag{6.9}$$

where $\mathbf{r}_e \in \mathbb{R}^{d_{\mathcal{E}}+d_{\text{width}}}$.

RE Decoder

We define a single relation type, *matches*, to link two entities. This relation is symmetric, as it makes no practical difference whether a KPI is matched to its value or vice versa. Furthermore, we limit candidate entity pairs to those marked as valid in the *relation matrix* (see Table 6.1), which lists permitted entity combinations.

After the model has processed an input, we further enforce the uniqueness constraints from Table 6.1. Specifically, if two entities are predicted to form

a unique relation more than once, we retain only the pairing with the higher confidence score. For example, if two KPI entities are matched to a single current-year value, we keep only the KPI–value pair with the higher score.

Following established practice, we obtain candidate pairs e_i and e_j, where $i < j$, from the set of all *valid* entity combinations in the input sentence. For each pair, we concatenate their representations with a local context embedding $\mathbf{r}_{\text{local}}$. In contrast to the global sentence-level embedding $\mathbf{r}_c$, $\mathbf{r}_{\text{local}}$ is derived by pooling the encoded word embeddings between e_i and e_j, defined as

$$\mathbf{r}_{\text{local}}(e_i, e_j) := \mathcal{P}(\mathbf{r}_i^{\mathcal{E}}, \ldots, \mathbf{r}_j^{\mathcal{E}}), \quad \mathbf{r}_{\text{local}}(e_i, e_j) \in \mathbb{R}^{d_{\mathcal{E}}}. \tag{6.10}$$

As observed by [328], this local context proves more effective for relation classification than the global BERT context token $\mathbf{r}_c$. Therefore, we define:

$$\mathbf{x}_{\text{RE}}(e_i, e_j) := \mathbf{r}_{e_i} \oplus \mathbf{r}_{\text{local}}(e_i, e_j) \oplus \mathbf{r}_{e_j}, \tag{6.11}$$

where $\mathbf{x}_{\text{RE}}(e_i, e_j) \in \mathbb{R}^{3d_{\mathcal{E}}+2d_{\text{width}}}$. Since the *matches* relation is symmetric, we do not compute $\mathbf{x}_{\text{RE}}(e_j, e_i)$.

Our binary relation classifier, a logistic regression, is then given by

$$\hat{y}_{\text{RE}} = \mathcal{A}\Big(\mathbf{w}_{\text{RE}}^{\top}\, \mathbf{x}_{\text{RE}}(e_i, e_j) + a_{\text{RE}}\Big), \tag{6.12}$$

where $\mathcal{A}(\cdot)$ is the sigmoid function, and $\mathbf{w}_{\text{RE}} \in \mathbb{R}^{3d_{\mathcal{E}}+2d_{\text{width}}}$ and $a_{\text{RE}} \in \mathbb{R}$ are learnable parameters. If the output of (6.12), i.e., $\hat{y}_{\text{RE}}$, exceeds a confidence threshold τ_{RE}, we consider e_i and e_j to form a valid match.

6.3.2 Regularisation by Injecting Noise

Let $\big[\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_n\big]$ denote the parameter matrices of a language model, where n indicates the total number of distinct parameter types. Following [358], we define the perturbed version $\tilde{\mathbf{W}}_i$ of a parameter matrix $\mathbf{W}_i$ as:

$$\tilde{\mathbf{W}}_i := \mathbf{W}_i + \mathcal{U}\left(-\frac{\lambda_{\text{noise}}}{2}, \frac{\lambda_{\text{noise}}}{2}\right) \cdot \sigma\big(\mathbf{W}_i\big), \tag{6.13}$$

where $\mathcal{U}(x_1, x_2)$ denotes uniformly distributed noise ranging from x_1 to x_2, λ_{noise} is a tunable hyperparameter controlling the noise intensity, and $\sigma(\cdot)$ computes the standard deviation of its argument. Therefore, parameters with higher variance receive more noise.

In addition to simply adding such noise to all parameters, as seen in [358], we investigate how the performance of the downstream task is affected when we only partially inject the noise term into certain parts of the model.

While [358] applies this perturbation globally, we investigate how selectively adding noise to certain parts of the model affects downstream performance. Specifically, we insert noise only into biases, only into weights, or into both at different intensities. In the context of relation extraction, we additionally inject noise into the residual connection and the layer-normalisation parameters. We also split the BERT [32] encoder into two separate *layer zones*, injecting noise with different intensities in each. This partitioning is inspired by [117], who theorised that BERT encodes different linguistic features at various layer depths.

Therefore, we modify Equation 6.13 by only perturbing a certain part of the parameter matrix $\mathbf{W}_i$:

$$\tilde{\mathbf{W}}_i^{\text{loc}} = \mathbf{W}_i^{\text{loc}} + \mathcal{U}\left(-\frac{\lambda_{\text{noise}}}{2}, \frac{\lambda_{\text{noise}}}{2}\right) \cdot \sigma\left(\mathbf{W}_i^{\text{loc}}\right), \tag{6.14}$$

where $\mathbf{W}_i^{\text{loc}}$ denotes the localised, i.e., restricted to certain parts of the model, section of $\mathbf{W}_i$, and $\tilde{\mathbf{W}}_i^{\text{loc}}$ is its perturbed counterpart.

6.3.3 Few-shot KPI Extraction with LLMs

To compare our previously introduced, fine-tuned approaches to state-of-the-art decoder-only setups, we evaluate how effectively LLMs can extract KPI-related information in a few-shot setting by crafting prompts to enforce JSON-formatted output. We then parse the generated JSON, aligning predicted relations with ground-truth references.

Generation of JSON output

Among the models tested, only GPT-3.5 Turbo [40], GPT-4 [41], and Mistral Large [270] explicitly support JSON output through their APIs. Other models must produce valid JSON autonomously. Outputs failing to comply with the JSON format or the KPI-EDGAR schema, detailed in Table 6.7, automatically receive an F_1 score of zero.

Extraction of Ranges from Generated Entities

In mapping each predicted entity to the corresponding text span, we start by attempting a full-span match between the model's predicted entity and the ground truth. Specifically, we compare each token in the predicted entity against the tokens in the sentence, checking sequentially from the first token to the last. If an exact match in token content and order is found, we retrieve the matching token indices (the *range*) and stop.

If no full-span match is detected, we gradually reduce the span length by creating all possible sub-spans of size $\ell - \iota$, where ℓ is the original span length, and ι is incremented from 1 to $\ell - 1$. Each sub-span undergoes the same sequential token-level comparison. This process is inspired by the adjusted F_1 metric from Equation (6.21). If none of these sub-spans align with the ground truth, we consider the entity to be invalid and assign an F_1 score of zero for that match.

Fuzzy Word Matching

In practice, word-level comparison must allow for certain exceptions to preserve essential information without sacrificing overall accuracy. Ideally, two strings are deemed equal if they match exactly. However, generated or ground-truth entities can contain noisy variations, such as enumerations, different tenses, or inconsistent numerical formatting (e.g., different floating-point precision, or stating "1.5 million" rather than "1,500,000"). In such cases, we still treat the strings as matching because the key information remains intact.

Prompt Design

Our prompt explicitly requests the extraction of KPIs and their corresponding values in JSON format. We specify the different entity classes that can appear in the dataset and provide multiple example sentences with their ground-truth entities. Each entity category is showcased at least once to improve the model's internal representation of, and response to, each type.

We iteratively refined the prompt using observations from model outputs to further enhance performance. This included ensuring numerical formatting that aligns with the ground truth, stressing that text extracted must match the original content verbatim, reiterating the requirement of one-to-one mappings, and insisting upon a JSON-formatted answer. The prompt satisfying these requirements is available in the Appendix B.1.

6.3.4 Baselines

In this subsection, we briefly outline the additional baselines used for comparison. We present the span-level baselines, followed by the CRF baselines, and conclude with further token-level baselines. The latter involves models that replace the encoder, including EDGAR–W2V [360], GloVe [71], and tf-idf vectorisation [67].

Span-Level Baseline

At the core of span-level model, we use the same architecture as presented in [328], where they named the model *SpERT*, a designation we also adopt for our approach. Passing a byte-pair encoded input sentence of length n through BERT [32], we obtain $n + 1$ token representations, $(\mathbf{r}_c^{\mathcal{E}}, \mathbf{r}_1^{\mathcal{E}}, \mathbf{r}_2^{\mathcal{E}}, \ldots, \mathbf{r}_n^{\mathcal{E}})$, where $\mathbf{r}_c^{\mathcal{E}}$ represents the contextual embedding for the entire sentence and $\mathbf{r}_i^{\mathcal{E}}$ the embedding for the token i.

In contrast to our KPI-BERT approach, we classify entire entity spans instead of individual tokens. Thus, we generate all possible token subsequences from the encoded BERT output, up to a maximum span length l, set to $l = 10$ to remain consistent with the original configuration of [328]. For example, given the input `(Gimli son of Glóin)`, we form the spans:

- `(Gimli)`,
- `(son)`,
- `(of)`,
- `(Glóin)`,
- `(Gimli, son)`,
- `(son, of)`,
- `(of, Glóin)`,
- `(Gimli, son, of)`,
- `(son, of, Glóin)`,
- `(Gimli, son, of, Glóin)`.

Thereafter, each span is classified into entity types, and for each possible entity pair, we predict whether it is linked.

Unlike the approach in [328], we additionally filter overlapping entity spans by removing those with a lower classification score. This is because overlapping spans are not permissible in our application of extracting KPIs from financial documents. To illustrate, consider the previous example. If we assume that the span `(Gimli)` and `(Gimli, son, of, Glóin)` are both classified as entities, the former with a score of 0.5 and the latter with 0.75, then in our approach, we would only retain the entity `(Gimli, son, of, Glóin)`.

CRF Baseline

As a competing approach to KPI-BERT, we implement a CRF layer, with the final sequence of NER tags determined using the Viterbi algorithm [361]. CRFs are probabilistic models particularly well-suited for sequence labelling tasks like NER [362]. Unlike models that make independent classification decisions for each token, a CRF considers the entire input sequence to predict the most probable sequence of labels globally. This inherent ability to model dependencies between adjacent labels allows the CRF to learn and enforce constraints on the output tag sequence, leading to more coherent predictions [363–365]. For example, such a CRF can learn that an "I-KPI" tag is unlikely to follow an "O" tag without an intervening "B-KPI" tag.

The input to our CRF layer consists of the contextualised token embeddings generated by the preceding sentence encoder. The Viterbi algorithm then searches through all possible tag sequences to find the one with the highest overall probability, given the representations from the tokens and the learned transition scores between tags within the CRF model.

Further Token-Level Baselines

As previously defined, KPI-BERT consists of three main components: a sentence encoder, an NER decoder, and an RE decoder. To generate additional baselines, the sentence encoder of KPI-BERT is replaced with EDGAR–W2V [360], GloVe [71], and tf-idf vectorisation [67] models. The remainder of the architecture remains unchanged. For these baselines, sentences are tokenised into words rather than WordPiece tokens.

For EDGAR–W2V and GloVe, word embeddings are extracted from the pre-trained embeddings provided by their respective authors. For tf-idf, the model is trained on the tokenised sentences from the training set of our data. Subsequently, word embeddings are obtained by first generating the tf-idf embedding for the entire sentence. Then, the tf-idf score of each word is extracted from this sentence embedding and stored in a sparse vector representation, where all entries are zero except for the entry representing the extracted tf-idf score. For words that are not present in the vocabulary in any of these three approaches, a random vector is substituted.

6.3.5 Adjusted F_1 Metric

To better capture the actual performance of relation extraction in financial texts, we propose an adjusted F_1 metric that awards partial credit when parts of an entity are correctly identified.

Given the following example sentence and ground truth relations,

> "This year, Palantír Communications Network (PCN) recorded $\$3.2_{\text{CY}}$ million in infrastructure maintenance expenses$_{\text{CY}}$ and $\$1.8_{\text{CY}}$ million in Palantír channelling costs$_{\text{KPI}}$."
>
> infrastructure maintenance expenses$_{\text{KPI}}$ – 3.2_{CY}, Palantír channelling costs$_{\text{KPI}}$ – 1.8_{CY},

we found that predictions often omit a part of non-numerical entities like the KPI entity. In this example, predicting *channelling costs* instead of *Palantír channelling costs* yields a strict, conventional F_1 score of zero for both relations, despite capturing the main concept accurately.

To address this, we first define $\mathfrak{o}_i$ as the overlap or intersection between the predicted tokens of entity i and their ground truth

$$\mathfrak{o}_i := \mathfrak{e}_{i,\text{pred}} \cap \mathfrak{e}_{i,\text{gt}}, \tag{6.15}$$

where $\mathfrak{e}_{i,\text{pred}}$ and $\mathfrak{e}_{i,\text{gt}}$ are sets of token identifiers for the predicted and ground-truth entity spans, respectively.

Next, the *true positives* ($\text{tp}_{\text{relation}}$), *false negatives* ($\text{fn}_{\text{relation}}$), and *false positives* ($\text{fp}_{\text{relation}}$) for a relation between entities i and j are calculated by:

$$\text{tp}_{\text{relation}} = \frac{1}{2}\left(\frac{|\mathfrak{o}_i|}{|\mathfrak{e}_{i,\text{gt}}|} + \frac{|\mathfrak{o}_j|}{|\mathfrak{e}_{j,\text{gt}}|}\right) \tag{6.16}$$

$$\text{fn}_{\text{relation}} = 1 - \text{tp}_{\text{relation}} \tag{6.17}$$

$$\text{fp}_{\text{relation}} = \frac{1}{2}\left(\frac{|\mathfrak{e}_{i,\text{pred}}| - |\mathfrak{o}_i|}{|\mathfrak{e}_{i,\text{pred}}|} + \frac{|\mathfrak{e}_{j,\text{pred}}| - |\mathfrak{o}_j|}{|\mathfrak{e}_{j,\text{pred}}|}\right), \tag{6.18}$$

where the operation $|\cdot|$ denotes set cardinality.

With these, we can calculate precision, recall, and the F_1 score in the conventional way:

$$\text{precision}_{\text{relation}} = \frac{\text{tp}_{\text{relation}}}{\text{tp}_{\text{relation}} + \text{fp}_{\text{relation}}} \tag{6.19}$$

$$\text{recall}_{\text{relation}} = \frac{\text{tp}_{\text{relation}}}{\text{tp}_{\text{relation}} + \text{fn}_{\text{relation}}} \tag{6.20}$$

$$\text{F}_{1,\text{relation}} = 2 \cdot \frac{\text{precision}_{\text{relation}} \cdot \text{recall}_{\text{relation}}}{\text{precision}_{\text{relation}} + \text{recall}_{\text{relation}}}. \tag{6.21}$$

This adjusted metric is applied solely to evaluate approaches on our KPI-EDGAR dataset, as the dataset was developed specifically to align with this metric. For the proprietary dataset introduced alongside KPI-BERT [11], the standard F_1 metric is used, because the original publication and its associated evaluation predated the introduction of this adjusted metric.

6.4 Experiments

In this section, we introduce our two datasets, describe how we trained and evaluated our relation extraction models, and present their respective performance outcomes.

6.4.1 Proprietary Dataset

This section details our proprietary dataset and examines the results obtained by KPI-BERT, followed by an exploration of various ablation studies. We begin with a description of the dataset creation, then outline our training configuration, and finally discuss model performance.

Dataset

Our dataset comprises 500 manually annotated financial documents, spanning a total of 15,394 sentences. These were collected from the *Bundesanzeiger*[1], an online platform maintained by the German Ministry of Justice, where businesses are legally required to publish financial statements and related documents (see § 325 HGB [366]).

The pre-processing phase involves:

1. **Tokenisation:** We first split each report into sentences and then into individual words using the `syntok` Python library.

[1] bundesanzeiger.de

Table 6.2: Description and support of all entity types in the proprietary KPI-BERT dataset, excluding the *none* type. The table is adapted from our paper [11].

Entity	Support	Description
KPI	16849	Key Performance Indicators expressible in numerical and monetary value, e.g. revenue or net sales.
CY	11498	Current Year monetary value of a KPI
PY	5057	Prior Year monetary value of a KPI.
INCREASE	356	Increase of a KPI from the previous year to the current year.
DECREASE	230	Decrease of a KPI from the previous year to the current year.
DAVON	8827	Davon, German for thereof, represents a *subordinate* KPI, i.e. if a KPI is part of another, broader KPI.
DAVON-CY	8443	Current Year value of a thereof KPI.
DAVON-PY	4382	Prior Year value of a thereof KPI.

2. **Monetary Tagging:** We apply rule-based heuristics to identify monetary values and capture their scale (e.g., "million") and currency unit (e.g., "$").

3. **Sentence Filtering:** We retain only those sentences that contain a recognised monetary value, as our primary interest is matching KPIs with their respective financial amounts.

4. **Manual Annotation:** We then generate token- and span-level annotations, specifying entity types and their relations.

A team of six qualified auditors, overseen by a senior auditing expert, carried out the manual annotation. Table 6.2 lists the entity classes and their respective frequencies. During annotation, the definitions of entity classes underwent several iterations to account for variations and edge cases in the data. In particular, distinguishing between *kpi* and *davon* posed challenges depending on sentence context. After completing the annotation, the senior auditor performed a quality check on 50 randomly selected documents, confirming that the overall annotations were of high quality. Due to time and budget constraints, each document was annotated only once, preventing us from reporting an inter-annotator agreement metric. Nevertheless, we remain confident in the dataset's overall accuracy.

Finally, the pre-processed dataset was split at the document level into training, validation, and test sets, containing 13,835, 821, and 738 sentences, respectively.

Baselines

We compare our proposed model to three alternative architectures, each using the same BERT-based sentence encoder from Section 6.3.1 to ensure fair comparisons.

Table 6.3: Hyperparameter configurations evaluated by grid search. The best configuration on the validation set is highlighted in boldface. The table is adapted from our paper [11].

Hyperparameter	Configurations
Word-, entity- and context pooling	**Bi-GRU**, Min, Max
NER decoding	**GRU**, CRF, Span, Linear
Conditional label masking	**True**, False
Dropout	0.0, **0.1**, 0.2, 0.3
Confidence threshold (τ_{RE})	0.4, **0.5**, 0.6
Filtering impossible relations	**True**, False
Removing overlapping relations	**True**, False
Batch size	**2**, 4, 8
Learning rate	$5e^{-5}$, $\mathbf{1e^{-5}}$, $5e^{-6}$
Weight decay	None, **0.01**, 0.1
Gradient normalization	None, **1.0**

First, we replace the GRU-based NER decoder with a simple linear layer that classifies named entities in parallel, based on the BERT-encoded word embeddings. This approach, described in [330], serves as a straightforward baseline because it disregards inter-label dependencies when predicting entity tags.

Second, we use SpERT [328] as introduced in Section 6.3.4. Our implementation closely follows the original implementation, but with two extensions: we include our novel Bi-GRU pooling method in the hyperparameter search and apply additional filtering for overlapping or invalid relations.

Third, we implement a CRF using Viterbi decoding [361] as a NER decoder. This is a popular setup for NER due to its ability to model label dependencies [363–365, 367]. To enable a fair comparison, we integrate the IOBES label constraints (see Table 6.1) by masking any invalid class transitions in the trainable transition matrix.

Training Setup and Hyperparameter Selection

To determine the optimal hyperparameter configuration for each model, we conduct an extensive grid search by evaluating different parameter combinations on the validation set's relation classification F_1-score. A relation is counted as correct if the spans and types of both related entities are accurately predicted. Table 6.3 lists all fine-tuned model parameters with their respective value ranges, highlighting in bold the best-performing setup on the validation set. Note that the "NER decoding" row distinguishes KPI-BERT (GRU_{LM}, i.e., GRU with conditional label masking) from the baseline models.

All approaches employ the cased `bert-base` encoder, pre-trained on a large German text corpus of over two billion tokens drawn from Wikipedia dumps,

Table 6.4: Ablation study of our tuned KPI-BERT model, applying different pooling functions and removing filtering heuristics and conditional label masking. F_1-scores are reported on the validation set. The table is adapted from our paper [11].

Configuration/Ablations	Relation F_1 in %
KPI-BERT	70.32
No conditional label masking	69.25
No filtering overlapping relations	69.73
No filtering impossible relations	69.47
No filtering impossible & overlapping relations	69.04
KPI-BERT$_{\text{max pooling}}$	69.16
KPI-BERT$_{\text{mean pooling}}$	69.34

news crawls, and other sources. Released by the MDZ Digital Library team (dbmdz),[2] it has the same architectural specifications as the English `bert-base` model: 12 multi-head attention layers, each with 12 attention heads, and 768-dimensional output embeddings. We initialise all trainable parameters from a normal distribution $\mathcal{N}(0, 0.02)$, fix the random seed at 42 for every training run, and adopt the AdamW optimiser [174] with a linear warm-up (10% of the total steps) followed by a linear decay learning rate schedule.

Furthermore, we set the width embedding dimension d_{width} to 25, the label embedding dimension d_{IOBES} to 128 (where applicable), and sample at most 100 negative relation instances per sentence. We additionally experiment with different dropout rates before the entity and relation classifiers, as well as varying levels of weight decay and gradient normalisation. The peak learning rate, batch size, and threshold τ_{RE} for relation prediction are also varied and evaluated.

Each model variation is trained for 20 epochs, and the best checkpoint is identified through early stopping.[3]

Ablation Study

During hyperparameter tuning, we paid particular attention to certain parameter ablations of KPI-BERT, with the corresponding results summarised in Table 6.4.

First, conditional label masking increases the model's relation F_1-score on the validation set by 1.07 percentage points, indicating the value of incorporating prior knowledge in the form of label dependencies. Second, we examined the effect of different pooling mechanisms: results show that trainable bidirectional GRU-based pooling layers outperform simple mean and max pooling by 1.16

[2] huggingface.co/dbmdz/bert-base-german-cased

[3] The best relation F_1-score on the validation set for KPI-BERT is achieved by epoch 18.

Table 6.5: Test set evaluation of the joint named entity and relation classification task, reporting mean (standard deviation) Precision-, Recall- and F_1-scores of 10 identical training runs with varying seeds. Our model, KPI-BERT, outperforms the competing state-of-the-art architectures in both entity extraction and relation linking. A relation is only considered correct, if both the spans and the types of the two related entities are classified correctly. The table is adapted from our paper [11].

(a) Entity scores in %

Name	Configuration	Precision*		Recall*		F_1^*	
–	BERT + Linear + RE	76.81	(1.00)	81.57	(0.59)	79.12	(0.72)
SpERT	BERT + Span + RE	75.67	(0.63)	**83.45**	(**0.46**)	79.37	(0.47)
–	BERT + CRF + RE	79.80	(0.63)	82.35	(0.51)	81.05	(0.51)
KPI-BERT	BERT + GRU + RE	**79.87**	(**0.55**)	82.31	(0.55)	**81.08**	(**0.53**)

* micro average

(b) Relation scores in %

Name	Configuration	Precision		Recall		F_1	
–	BERT + Linear + RE	66.95	(1.51)	69.34	(1.13)	68.12	(1.26)
SpERT	BERT + Span + RE	67.00	(0.84)	69.48	(0.63)	68.22	(0.61)
–	BERT + CRF + RE	**70.68**	(**0.81**)	70.62	(0.93)	70.65	(0.83)
KPI-BERT	BERT + GRU + RE	70.33	(0.55)	**71.43**	(**0.60**)	**70.88**	(**0.55**)

percentage points. Third, we evaluated how filtering overlapping and impossible relations influences overall performance. Although both heuristics improve the relation extraction F_1-score, the exclusion of impossible relations yields a larger gain, which aligns with the task's simplified structure due to many relations being impossible, as seen in Table 6.1.

Results

After selecting the best hyperparameters, we train KPI-BERT and each baseline on the union of the original training and validation sets from scratch. To account for variability arising from random weight initialisation, every model is trained ten times with different seeds. We then evaluate the resulting checkpoints on the held-out test set for the joint named entity recognition and relation extraction task. Table 6.5 reports the mean and standard deviation of each metric across the ten runs.

KPI-BERT scores the highest for both entity and relation classification, achieving F_1-scores of 81.08% and 70.88%, respectively. By contrast, SpERT and the linear NER decoder perform remarkably worse on both tasks, most likely because they ignore label dependencies when predicting entity tags. The CRF-based model with conditional label masking does consider such dependencies, yet

Table 6.6: Description and support of all entity types in the KPI-EDGAR dataset, excluding the *none* type. The table is adapted from our paper [12].

Entity	Support	Description / Annotation Guideline
KPI	1341	Key Performance Indicators expressible in numerical and monetary value, e.g. revenue or net sales.
CY	1211	Current Year monetary value of a KPI.
PY	619	Prior Year monetary value of a KPI.
PY1	307	2 Year Past Value of a KPI.
INCREASE	35	Increase of a KPI from the previous year to the current year.
INCREASE-PY	15	Analogous to increase, but from py1 to py.
DECREASE	23	Decrease of a KPI from the previous year to the current year.
DECREASE-PY	11	Analogous to decrease, but from py1 to py.
THEREOF	507	Represents a *subordinate* KPI, i.e. if a KPI is part of another, broader KPI.
ATTR	272	Attribute that further describes a KPI.
KPI-COREF	11	A co-reference to a KPI mentioned in a previous sentence.
FALSE-POSITIVE	170	Captures tokens that are similar to other entities, but are explicitly not one of them, e.g. when the writer of the report forecasts next year's revenue.

it still lags behind KPI-BERT and exhibits a larger standard deviation, indicating lower robustness to random initialisation.

6.4.2 KPI-EDGAR

In this section, we describe our published open-source dataset titled "KPI-EDGAR" and discuss the results obtained by various fine-tuned and zero-shot approaches. As before, we begin this section with a description of the dataset and its creation, then we outline which baselines we use for benchmarking, and we close this section with the performances of various model configurations.

Dataset

Our corpus comprises 81 manually annotated *10-K* reports containing 1,355 sentences, 4,522 entities, and 3,841 relations. A *10-K* report is a comprehensive financial annual filing in which a publicly listed company discloses its financial performance. All documents were scraped from the EDGAR (Electronic Data Gathering, Analysis, and Retrieval) system, a database maintained by the U.S. Securities and Exchange Commission that hosts statutory filings for every company traded on U.S. exchanges.

For pre-processing, we use the same methodology as before. To reiterate this process: First, each report is tokenised into sentences and subsequently into words. Next, monetary values are detected, along with their scale (e.g.,

Table 6.7: Comprehensive overview of all allowed relations and their uniqueness. "1:1": One entity of type 1 can only be linked to one entity of type 2, "1:n": One entity of type 1 can be linked to many entities of type 2. "-": No relation possible. The table is adapted from our paper [12].

	KPI	CY	PY	PY1	INCREASE	INCREASE-PY
KPI	-	1:1	1:1	1:1	1:1	1:1
CY	1:1	-	-	-	-	-
PY	1:1	-	-	-	-	-
PY1	1:1	-	-	-	-	-
INCREASE	1:1	-	-	-	-	-
INCREASE-PY	1:1	-	-	-	-	-
DECREASE	1:1	-	-	-	-	-
DECREASE-PY	1:1	-	-	-	-	-
THEREOF	n:1	1:1	1:1	1:1	1:1	1:1
ATTR	n:1	-	-	-	-	-
KPI-COREF	-	1:1	1:1	1:1	1:1	1:1
FALSE-POSITIVE	-	-	-	-	-	-

(a)

	DECREASE	DECREASE-PY	THEREOF	ATTR	KPI-COREF	FALSE-POSITIVE
KPI	1:1	1:1	1:n	1:n	-	-
CY	-	-	1:1	-	1:1	-
PY	-	-	1:1	-	1:1	-
PY1	-	-	1:1	-	1:1	-
INCREASE	-	-	1:1	-	1:1	-
INCREASE-PY	-	-	1:1	-	1:1	-
DECREASE	-	-	1:1	-	1:1	-
DECREASE-PY	-	-	1:1	-	1:1	-
THEREOF	1:1	1:1	-	-	n:1	-
ATTR	-	-	-	-	n:1	-
KPI-COREF	1:1	1:1	1:n	1:n	-	-
FALSE-POSITIVE	-	-	-	-	-	-

(b)

billion) and currency unit (mostly U.S. dollars), using rule-based string-matching heuristics. This enables us to retain only those sentences containing a monetary expression, as the dataset's primary aim is to link numerical amounts to the corresponding KPIs. The selected sentences are then annotated for word-level entities and their relations.

Four annotators, overseen by a senior auditing specialist, produced the annotations. The set of entity classes (Table 6.6) and the relation matrix that restricts permissible links between entities (Table 6.7) were defined in consultation with a wider group of auditors, whose feedback proved invaluable during the project's early stages. To assess annotation quality, the lead auditor re-annotated 41 documents originally labelled by the other three annotators. The resulting

Table 6.8: Model performances, measured in the conventional as well as adjusted (see subsection 6.3.5) F_1 score, of the models described in sections 6.3.1 and 6.3.4 on KPI-EDGAR. The table is adapted from our paper [12].

Model	Relation F_1 in %	Adjusted Relation F_1 in %
KPI-BERT	**22.68**	**43.76**
SpERT	20.95	40.04
EDGAR–W2V	6.13	19.71
GloVe	5.11	17.18
tf-idf	0	0.25

inter-annotator agreement, measured as Cohen's κ [4] [7] at the word level, is 0.7037. During the analysis of the results, we provide further discussion, and Table 6.10 shows additional inter-annotator statistics.

Baselines

To benchmark our primary KPI-BERT model (Section 6.3.1), we evaluate several baseline configurations. These alternatives replace KPI-BERT's sentence encoder with EDGAR–W2V [360], GloVe [71], or TF-IDF vectorisation [67], as detailed in Section 6.3.4. The performance of these baselines, alongside KPI-BERT, in their standard noise-free state provides comparative results.

We further investigate the impact of noise injection as a regularisation technique on KPI-BERT, following the methodology in Section 6.3.2. First, noise is applied globally to all model weights, as shown in [358] and Equation 6.13. Second, more targeted noise injection strategies (detailed in Section 6.3.2 and Equation 6.14) are employed. For the KPI-BERT models, this involves adding targeted noise to residual connections, layer normalisation parameters, and distinct BERT layer zones.

Results

First, we evaluate several approaches on our KPI-EDGAR dataset: namely the ones introduced in Section 6.3.1 and 6.3.4, KPI-BERT, SpERT, EDGAR–W2V, GloVe, and tf-idf. Table 6.8 presents both the conventional and the adjusted F_1 scores for these configurations. An initial and obvious observation is that approaches without a transformer-based encoder perform substantially worse compared to those that utilise such an encoder. Given the sequential and context-based nature of detecting and extracting key performance indicators and their values,

[4] Cohen's κ ranges from 1 (perfect agreement) to -1 (complete disagreement), with 0 indicating chance-level agreement; see [368] for interpretation guidelines.

Table 6.9: Several example sentences from the test set of KPI-EDGAR with joint named entity recognition and relation extraction results. Green, Blue and Yellow represent "true positive", "false positive", and "false negative" entity and relation classifications, respectively. The predictions are generated by KPI-BERT [11]. The table is adapted from our paper [12].

	Sentence with predicted Entities	Relations
(a)	Entity boundaries are debatable, leading to arguably correct predictions but a conventional F_1 score of zero	
1	(a) [[Unrealized gains]$_{\text{kpi}}$ totaled]$_{\text{kpi}}$ $[96]$_{cy} million in 2020, $[88]$_{py} million in 2019 and $[73]$_{py1} million in 2018 [...].	kpi – cy kpi – cy kpi – py kpi – py kpi – py1 kpi – py1
2	As of December 31, 2020 and 2019, the Company's [Medicare Part D [receivables]$_{\text{kpi}}$]$_{\text{kpi}}$ amounted to $[2.9]$_{cy} billion and $[2.3]$_{py} billion, respectively.	kpi – cy kpi – cy kpi – py kpi – py
(b)	Difference in prediction and ground truth, however both are viable options. Nevertheless, the conventional and adjusted F_1 score will assign 0 here.	
3	As a result, we recognized $[50]$_{cy} million of [costs]$_{\text{kpi}}$ primarily related to [employee termination expenses and losses]$_{\text{kpi}}$ from closing certain stores impacting both segments.	kpi – cy kpi – cy

this outcome aligns with our expectations for attention-based models, given their inherent capacity to model such relationships. Furthermore, the tf-idf vectorisation approach proved entirely unable to capture these relationships.

Considering the adjusted F_1 score, it is, as expected, notably higher than the conventional score. This begs the question: does it do a better job in actually measuring the success of the prediction?

To address this question, Table 6.9 presents several examples taken directly from the test set of KPI-EDGAR. Subtable (a) highlights instances where our adjusted F_1 score allows the metric to reflect more accurately the model's true performance. These examples also reveal a critical challenge that both annotators and, consequently, machine learning models encounter when extracting KPIs from financial documents: Where are the exact boundaries of an entity? What are relevant pieces of information that still belong to the entity in question, and which information is less important? Often, two expert auditors will differ in their opinions regarding the precise placement of these boundaries.

This inherently ambiguous and noisy process can also be observed from Table 6.10, which reports various Cohen's Kappa scores. The table indicates that

Table 6.10: Cohen's Kappa [7] scores of various input series. *All words* includes all word tokens. *Only entities* only includes word tokens that belong to an entity annotation. Types starting with *Entity:* are calculated by only considering word tokens that are of such an entity. The table is adapted from our paper [12].

Type	Cohen's Kappa
All words	0.7037
Only entities	0.4885
Entity: KPI	0.0822
Entity: CY	0.5972
Entity: PY	0.6657
Entity: PY1	0.6095
Entity: INCREASE	0.7419
Entity: INCREASE-PY	0.5556
Entity: DECREASE	0.7713
Entity: DECREASE-PY	-0.1538
Entity: THEREOF	0.3053
Entity: ATTR	-0.1076
Entity: KPI-COREF	-0.2592
Entity: FALSE-POSITIVE	-0.7587

annotators exhibit strong agreement on the location of numeric entities such as *CY* or *PY*; however, the location, and particularly the boundaries, of non-numeric entities like *KPI* or *THEREOF* are subject to greater debate.

Therefore, we suggest that our proposed adjusted F_1 score, by design, accommodates some imprecision in the predicted boundaries. Consequently, it should be better able to capture these fuzzy borders more accurately, particularly when compared with the conventional strict F_1 score.

Nevertheless, the adjusted F_1 score cannot fully account for all variations in annotation and prediction differences, as illustrated in Subtable (b) of Table 6.10. Detecting that both options are viable presents a significantly greater challenge. This level of discrimination would require in-depth auditing knowledge, a capability that our proposed weighting scheme inherently cannot possess.

Thereafter, we evaluate the noise injection approach introduced in [322] and detailed in Section 6.3.2. As shown in Table 6.11, the application of noise generally enhances the models' ability to generalise. This targeted noise injection strategy resulted in a remarkable increase of 2.85% in F_1 score compared to the non-perturbed model, and an improvement of 1.09% over the global NoisyTune approach [358]. These results, therefore, confirm the findings of [358] regarding the benefits of noise injection. Furthermore, our findings demonstrate that performance can be further enhanced beyond global noise application by selectively applying noise to specific components of the model architecture.

Table 6.11: Results of adding noise to certain parts of KPI-BERT, evaluated on KPI-EDGAR. Adding noise to all parameters is equivalent to the approach from [358]. Add&Norm refers to the process of adding noise to residual connections and layer normalisation. The table is adapted from our paper [322] and the models were retrained.

Noise added to	λ_{noise}	Adjusted F_1 in %
Nothing	0.00	43.76
All	0.81	45.52
Bias	**0.41**	**46.61**
Weights	0.50	45.05
Add&Norm	0.20	46.27
Layer zones	0.90	44.81

Finally, to provide a broader perspective, we also evaluated a diverse set of nine LLMs on the KPI-EDGAR dataset utilising the prompt presented in Appendix B.1. The results of this are presented in Table 6.12. In these few-shot evaluations, proprietary models generally outperformed their open-source counterparts. Mistral-Large [270] achieved the highest F_1 score among the LLMs, at 30.75, followed by GPT-4 [41] and GPT-3.5-turbo [40] with scores of 26.91 and 23.89, respectively. Open-source models such as Llama-3 [98] and Mixtral [224] yielded lower F_1 scores of 21.30 and 14.48. Interestingly, these performance rankings do not entirely align with general LLM leaderboards (e.g., the LMSYS Chatbot Arena Leaderboard [186]), suggesting that conversational prowess does not directly translate to capability in specialised extraction tasks like KPI identification.

However, even the leading F_1 score of 30.75 achieved by Mistral-Large in a few-shot setting falls short of the performance levels demonstrated by the more specialised, fine-tuned models discussed previously in this work, i.e., KPI-BERT and its optimised variants, as shown in Table 6.8 and 6.11. While the LLM evaluation was conducted using a few-shot paradigm, contrasting with the full fine-tuning employed for the bespoke models, this significant performance difference underscores a key finding: for high-precision, domain-specific tasks such as KPI extraction from financial documents, the current generation of general-purpose LLMs cannot yet compete with smaller, dedicated models that have been thoroughly fine-tuned on relevant training data. This highlights the continued value of tailored architectures and focused training for achieving state-of-the-art results in complex information extraction scenarios.

Table 6.12: Evaluation of the few-shot performance of various LLMs on the test set of KPI-EDGAR. The table is adapted from our paper [323].

Model	Size	Open-Source	Adjusted F_1
GPT-3.5-turbo [40]	N.A.	No	23.89
GPT-4 [41]	N.A.	No	26.91
Llama-2 [96]	70B	Yes	6.31
Llama-2 [96]	13B	Yes	14.49
Llama-2 [96]	7B	Yes	10.07
Llama-3 [98]	70B	Yes	21.30
Mistral-Large [270]	N.A.	No	**30.75**
Mixtral [224]	8x7B	Yes	14.48
Zephyr-β [369]	7B	Yes	11.40

6.5 Conclusion

This chapter has charted a comprehensive investigation into the automated extraction of KPIs and their associated values from financial documents. We started with the development of a fine-tuned neural transformer architecture and ended with the evaluation of cutting-edge LLMs, consistently aiming to enhance the accuracy, robustness, and practical applicability of information extraction techniques within the complex financial domain.

Our initial contribution, KPI-BERT, introduced a novel end-to-end system specifically designed for joint NER and RE in German financial reports. By integrating a BERT-based encoder with an RNN employing conditional label masking, KPI-BERT successfully modelled crucial label dependencies and accounted for the sequential nature of entity tagging. This hybrid approach, further enhanced by a trainable RNN-based pooling mechanism for word representations, outperformed span-based methods like SpERT [328] and other baselines in both entity recognition and relation extraction.

Building upon this, we addressed the need for public benchmarks and more nuanced evaluation by introducing the KPI-EDGAR dataset, a manually annotated corpus of U.S. American *10-K* reports. Alongside this dataset, we proposed an adjusted F_1 score. This metric, incorporating a word-level weighting scheme, offers a more granular assessment of model performance by awarding partial credit, thereby better reflecting the practical success of extraction when faced with the inherently fuzzy entity boundaries prevalent in financial texts. Our findings confirmed its superiority over the conventional strict F_1 score for this domain and especially our dataset, and we recommend its adoption when evaluating models on KPI-EDGAR.

Subsequently, we explored hybrid methods to further refine the training of such specialised models. We investigated the targeted injection of noise into various components of pre-trained transformer models during fine-tuning. This study demonstrated that such controlled perturbations can act as an important regularisation tool, significantly enhancing performance on joint NER and RE tasks with KPI-EDGAR. Notably, selectively applying noise to specific model parts, such as biases, weights, or residual connections and layer normalisation parameters, yielded improvements beyond those achieved by global noise application, like the one proposed in [358].

Finally, we situated our fine-tuned approaches within the context of the rapidly evolving LLM landscape. Our evaluation of nine diverse LLMs on the KPI-EDGAR dataset in a few-shot setting revealed that whilst proprietary models like Mistral-Large [270] and GPT-4 [41] surpassed their open-source counterparts, even the best-performing LLM fell substantially short of the accuracy achieved by our fine-tuned KPI-BERT system, especially when enhanced with controlled noise. These differences, even considering the few-shot versus full fine-tuning paradigms, underscore a critical insight: for high-precision, domain-specific tasks such as KPI extraction, current general-purpose LLMs, when applied with limited task-specific examples, are not yet a substitute for smaller, dedicated models meticulously fine-tuned on relevant data. This also highlighted that strong conversational abilities, as often reflected in general LLM leaderboards, do not directly translate to specialised information extraction prowess.

Our contributions open several promising avenues for future research. This includes further developing language models specifically tailored to the financial domain, either by continued pre-training of existing large architectures on extensive financial corpora like EDGAR, or by designing novel architectures with enhanced numerical and tabular reasoning capabilities that move beyond the limitations of plain text. Investigating cross-attention-based transformers, potentially coupled with conditional label masking, could also lead to more sophisticated sequential tagging and relation classification. Furthermore, experimentation with alternative noise distributions for regularisation, and extending the application of targeted noise injection techniques to the pre-training or fine-tuning of LLMs, could improve their adaptability and potentially reduce the need for vast fine-tuning datasets.

Different paradigms for leveraging LLMs also warrant exploration, such as fine-tuning them specifically for NER as a precursor to relation extraction, or for direct structured output generation using more sophisticated prompting or

in-context learning strategies. We have undertaken such an approach in our paper [183] and in the previous chapter with our iNERD architecture. Linking such an approach with an RE strategy might yield promising results.

The quality and scope of training data and evaluation metrics are equally paramount for continued progress. Future efforts might aim to enrich available datasets by continuing to improve the availability, quality, and quantity of financial corpora. For instance, creating a comprehensive NER dataset based on GAAP entities from EDGAR or expanding KPI-EDGAR to include a wider variety of documents and KPI types would likely be a worthwhile undertaking. Concurrently, investigating dynamic weighting schemes for the adjusted F_1 score, perhaps by incorporating the semantic importance of individual words within an entity or relation, could provide an even more nuanced performance measure. There is also significant potential in evaluating the performance of these models and regularisation techniques on datasets in other languages or in low-resource settings, where the benefits of robust regularisation might be particularly pronounced due to data scarcity and the inherent imbalances in multilingual model training.

Finally, the principles of controlled noise injection, in particular, merit investigation across a wider array of NLP tasks beyond joint NER and RE. In [322], we have already investigated a summarisation task and found the noise injection regularisation to be beneficial. Extending this research to areas such as natural language inference or sentiment analysis would help to establish the generalisability of its benefits across diverse NLP applications.

Since I doubt, I think; since I think, I exist.

— Dennis Taylor in *We are Legion (We are Bob)* [370]

7 Dementia Detection

Contents

Accurate detection of dementia is crucial for timely intervention and care, and leveraging multimodal data holds significant potential for improving diagnostic accuracy. In this chapter, we explore deep learning approaches for dementia classification using the Pitt corpus, which includes brief descriptions of the "Cookie Theft" picture description task from participants. We analyse 242 control and 307 dementia audio clips to investigate various representation learning techniques. Our best-performing approach fuses audio spectrograms with advanced language models, including Whisper model transcriptions and transformer-based feature extraction. We rigorously evaluate these models and find that our multimodal approach, with an F_1-score of 86.42%, surpasses single-modality approaches by a considerable margin. Our findings underscore the promise of multimodal deep learning in advancing the reliability of dementia detection through audio analysis, paving the way for more robust and accessible diagnostic tools.

7.1 Introduction

This chapter is based on our publication **"Fusing Speech and Language Models for Dementia Detection"** (co-authored with Abdul Mohsin Siddiqi, Lorenz Sparrenberg, Tobias Adams, Christian Bauckhage, and Rafet Sifa) published in the proceedings of the *12th IEEE International Conference on Big Data* [371].

This final empirical chapter extends the hybrid paradigm into the multimodal domain, illustrating its potential beyond text-based information extraction. Dementia, particularly Alzheimer's disease (AD), represents a growing global health challenge, as the number of dementia cases is projected to surge from 57.4 million in 2019 to 152.8 million by 2050, driven largely by the ageing global population [372]. Dementia is not a single disease but a collection of symptoms marked by a decline in cognitive abilities relative to an individual's previous level of functioning [373]. It includes several distinct types, such as Alzheimer's disease (AD), Vascular Dementia (VaD), Lewy Body Dementia (LBD), Frontotemporal Dementia (FTD), and Mixed Dementia (MD) [374], with AD being the most prevalent, accounting for 60–70% of all dementia cases [375]. AD is marked by the build-up of intracellular neurofibrillary tangles and extracellular β-amyloid plaques, along with widespread synaptic loss and neuronal atrophy in the brain [376]. These neuropathological changes can begin years before clinical symptoms manifest [377]. However, a definitive diagnosis of AD can only be established through microscopic examination of brain tissue, typically during an autopsy [378, 379]. Consequently, the term Dementia of the Alzheimer's Type (DAT) is used for suspected cases of AD that have not been clinically confirmed [380]. Early and accurate diagnosis of DAT is crucial for improving patient outcomes, emphasising the need for accessible diagnostic methods capable of detecting cognitive impairments in the earliest stages. Traditional diagnostic approaches, based on clinical assessments and neuropsychological testing, are often time-consuming and subject to variability in interpretation. This has driven a growing demand for automated, scalable diagnostic solutions that offer greater consistency, efficiency, and earlier detection [17, 381–384].

While memory loss is often regarded as the primary symptom of DAT [378], language also provides a valuable source of clinical information. Speech analysis, in particular, presents a promising non-invasive and cost-effective method for detecting cognitive decline. Linguistic and paralinguistic features of speech, such as fluency, articulation, and prosody, are well-established markers of dementia, offering insights into the early signs of cognitive impairment [385–387]. Recent

advancements in machine learning and multimodal representation learning, which combine acoustic and linguistic features, have demonstrated significant potential for improving the accuracy of dementia detection.

In this paper, we utilise the Pitt dataset [388], which contains speech samples from both diagnosed dementia patients and a control group, to evaluate multimodal approaches. For transcription, we employ Whisper [389], while BERT [32] and Stella [390] are used for language representation. Additionally, spectrograms are used to extract audio features. Our goal is to improve dementia classification performance by comparing a variety of machine learning models, with a particular focus on integrating both audio and text features for a more comprehensive analysis.

Our multimodal dementia detection system improves upon existing single-modality systems like [391] or [392] by a considerable margin. We achieve a Classification Accuracy of 86.59% and F_1-score of 86.42% on our hold-out test set, demonstrating the feasibility of our approach.

In the following sections, we first discuss related studies. Section 7.3 describes our methodology. Thereafter, we highlight our experiments and results in Section 7.4. We close this chapter with a conclusion and an outlook on future work.

7.2 Related Work

Several studies have explored the use of machine learning to automatically detect dementia and cognitive decline. Early approaches employed classical machine learning techniques, such as decision trees or Naive Bayes classifiers [393–395].

Modern dementia detection systems predominantly rely on deep neural networks and can be broadly categorised into three primary modalities: imaging data, clinical variables, and voice/language data [374].

Imaging data, particularly magnetic resonance imaging (MRI), has been extensively studied in dementia detection [396–398]. Image-based methods often outperform other modalities in terms of accuracy. For instance, one study reported 97% accuracy in multi-class Alzheimer's disease (AD) stage classification using the ADNI dataset and a pre-trained VGG19 model [399]. Despite their high performance, imaging-based techniques require expensive hardware and the involvement of skilled professionals for data acquisition and interpretation, which limits their practical application.

Clinical variables constitute the second major modality in dementia detection. These approaches typically involve cognitive assessments such as the Mini-Mental State Examination (MMSE) [400], as well as the analysis of biomarkers like amyloid, p-tau, and t-tau, often integrated with demographic factors such as age and gender [401]. Some studies also incorporate mRNA-based biosignatures [402] or leverage socio-demographic, basic health, and cognitive reserve proxy data [403]. While these methods have demonstrated promising results, they require specialised testing and expert interpretation, limiting their feasibility for routine, large-scale implementation.

The third modality focuses on voice and language data, which stands out due to its simplicity, non-invasive nature, and minimal hardware requirements. Initial efforts to utilise speech and language features for dementia detection were pioneered by studies such as [404], [405], and [406], which laid the groundwork for future research in this area. More recent studies have shown that both linguistic and paralinguistic features of speech can be powerful indicators of dementia [407–413].

To establish standardised benchmarks for voice/language data, the ADReSS Challenge was introduced by [414], aiming to standardise Alzheimer's dementia detection through spontaneous speech analysis. This challenge provided a benchmark dataset and tasks for dementia classification and MMSE score regression, reinforcing the need for more robust models to improve upon existing techniques. Moreover, [391] demonstrated that purely acoustic features, particularly paralinguistic ones, could yield competitive accuracy in Alzheimer's detection using the Pitt [388] corpus. Their novel Active Data Representation (ADR) method achieved 78.70% classification accuracy, highlighting the potential of non-verbal speech cues for dementia diagnosis.

Our work builds on these studies by integrating acoustic and textual features more effectively. We combine audio spectrograms with the Whisper transcription model [389] and use advanced language models to capture rich linguistic representations, aiming to improve dementia classification performance through a comprehensive multimodal approach.

For a more thorough review of recent advances in neurodegenerative disease detection using machine learning, we refer readers to the work of [415], [374], and [381].

7.3 Methodology

We split this section into three parts: The first describes how we extract features from the audio signal, the second how we can leverage transcription models and language models to find a linguistic representation, and the third sheds light on how we combine these two methods to arrive at our final model architecture.

7.3.1 Audio

To convert a raw audio signal $\mathbf{x}$ into a feature representation suitable for dementia detection, we compute a Mel-spectrogram [416]. Given this raw audio signal in which i_{raw} represents the index of said audio sample and with n_{samples} total number of samples,

$$x_{i_{\text{raw}}}, \quad i_{\text{raw}} = 0, 1, 2, ..., n_{\text{samples}} - 1, \tag{7.1}$$

we first divide the audio sample into overlapping frames $\mathbf{X}^w$. Each such frame $\mathbf{x}^w_{i_{\text{frame}}}$ is windowed using a Hann window [8], defined as:

$$x^w_{i_{\text{frame}}, i_{\text{window}}} = x_{i_{\text{frame}} \cdot \lambda_{\text{hop}} + i_{\text{window}}} \cdot \mathcal{W}(i_{\text{window}}), \tag{7.2}$$

where i_{frame} is the frame counter, i_{window} is the index of the time sample within the current windowed frame $\mathbf{x}^w_{i_{\text{frame}}}$, ranging from 0 to $\ell - 1$, λ_{hop} is the distance between frames, and $\mathcal{W}(\cdot)$ is the Hann window function:

$$\mathcal{W}(i_{\text{window}}) = 0.5 \left(1 - \cos\left(\frac{2\pi i_{\text{window}}}{\ell - 1}\right)\right) \tag{7.3}$$

and ℓ is the window length.

For each windowed frame, we compute the Short-Time Fourier Transform (STFT) [417] to obtain the frequency domain representation:

$$x^s_{i_{\text{freq}}, i_{\text{frame}}} = \sum_{i_{\text{window}}=0}^{\ell-1} x^w_{i_{\text{frame}}, i_{\text{window}}} \cdot e^{-\text{i}2\pi \frac{i_{\text{freq}} i_{\text{window}}}{\ell}}, \quad i_{\text{freq}} = 0, 1, \ldots, n_{\text{freq}} - 1 \tag{7.4}$$

where i_{freq} corresponds to the current frequency bin, $n_{\text{freq}} = \left\lfloor \frac{\ell}{2} \right\rfloor + 1$ represents the number of frequency bins, and $x^s_{i_{\text{freq}}, i_{\text{frame}}} \in \mathbb{C}$. The magnitude spectrogram is obtained by taking the absolute value of the Fourier coefficients $x^s_{i_{\text{freq}}, i_{\text{frame}}}$:

$$x^m_{i_{\text{freq}}, i_{\text{frame}}} = |x^s_{i_{\text{freq}}, i_{\text{frame}}}| \tag{7.5}$$

where $x^m_{i_{\text{freq}},i_{\text{frame}}}$ represents the magnitude of the frequency component at bin i_{freq} for the current windowed frame and $x^m_{i_{\text{freq}},i_{\text{frame}}} \in \mathbb{R}_{\geq 0}$.

Next, we apply a set of n_{filter} triangular filters to the magnitude spectrogram. These filters are spaced according to the Mel scale [418], which warps the linear frequency axis (measured in Hertz) to better align with human auditory perception. This transformation allows us to create a feature representation that is more perceptually meaningful and closer to how human perceive audio signals. The formula to convert a linear frequency f^{lin} to the Mel scale is defined as [419]:

$$\mathcal{F}_M(f^{\text{lin}}) = 2595 \cdot \log_{10}\left(1 + \frac{f^{\text{lin}}}{700}\right). \tag{7.6}$$

This serves as a blueprint for constructing the set of n_{filter} triangular filters. The goal is to define the filters' centre-points, denoted by the frequency bin indices $b_c[j]$, such that they are equally spaced in the Mel domain rather than the linear frequency domain.

The desired linear frequency range, from $f^{\text{lin}}_{\text{min}}$ to $f^{\text{lin}}_{\text{max}}$, is converted to the Mel scale using the definition from above:

$$f^{\text{mel}}_{\text{min}} = \mathcal{F}_M(f^{\text{lin}}_{\text{min}}) \tag{7.7}$$

$$f^{\text{mel}}_{\text{max}} = \mathcal{F}_M(f^{\text{lin}}_{\text{max}}) \tag{7.8}$$

A set of $n_{\text{filter}} + 2$ points, denoted by the vector $\mathbf{f}^{\text{mel}}$, is then created with linear spacing between these two Mel boundaries.

$$\mathbf{f}^{\text{mel}} = \left[f^{\text{mel}}_{i_{\text{filter}}} \mid f^{\text{mel}}_{i_{\text{filter}}} = f^{\text{mel}}_{\text{min}} + i_{\text{filter}} \cdot \frac{f^{\text{mel}}_{\text{max}} - f^{\text{mel}}_{\text{min}}}{n_{\text{filter}} + 1} \right], \quad i_{\text{filter}} = 0, 1, \ldots, n_{\text{filter}} + 1 \tag{7.9}$$

Each point in this Mel-space vector $\mathbf{f}^{\text{mel}}$ is subsequently converted back to the linear frequency scale (Hertz) using the inverse Mel transformation:

$$\mathcal{F}^{-1}_M(f^{\text{mel}}) = 700 \cdot \left(10^{f^{\text{mel}}/2595} - 1\right) \tag{7.10}$$

This yields a vector of frequencies $\mathbf{f}^{\text{lin}}$ where $\mathbf{f}^{\text{lin}}_{i_{\text{filter}}} = \mathcal{F}^{-1}_M(f^{\text{mel}}_{i_{\text{filter}}})$, which are now spaced according to the Mel scale. In the final step, each of these frequencies f^{lin}_j are transformed into the discrete frequency bin indices $b_c[j]$ required for the spectrogram. This conversion depends on the sampling rate f_s and the window length ℓ:

$$b_c[j] = \left\lfloor \frac{\ell}{f_s} f^{\text{lin}}_j \right\rfloor \tag{7.11}$$

This procedure results in filter centre-points that are close together at low frequencies and more spread out at high frequencies, mimicking human auditory perception. With the set of bin indices $b_c[j]$ now fully defined, the triangular filters can be applied to the magnitude spectrogram. Each filter sums the magnitude in its corresponding frequency band:

$$x^f_{j,i_{\text{frame}}} = \sum_{i_{\text{freq}}=0}^{n_{\text{freq}}-1} x^m_{i_{\text{freq}},i_{\text{frame}}} \cdot \mathcal{F}^{\text{tri}}_j[i_{\text{freq}}] \tag{7.12}$$

where $\mathcal{F}^{\text{tri}}_j[i_{\text{freq}}]$ is the j-th triangular Mel filter defined as:

$$\mathcal{F}^{\text{tri}}_j[i_{\text{freq}}] = \begin{cases} 0, & \text{if } i_{\text{freq}} < b_c[j-1] \\ \frac{i_{\text{freq}}-b_c[j-1]}{b_c[j]-b_c[j-1]}, & \text{if } b_c[j-1] \leq i_{\text{freq}} \leq b_c[j] \\ \frac{b_c[j+1]-i_{\text{freq}}}{b_c[j+1]-b_c[j]}, & \text{if } b_c[j] < i_{\text{freq}} \leq b_c[j+1] \\ 0, & \text{if } i_{\text{freq}} > b_c[j+1] \end{cases}, \tag{7.13}$$

in which we define $b_c[-1]$ and $b_c[n_{\text{filter}}]$ as boundary bins outside the valid filter range, set to 0 and $n_{\text{freq}} - 1$ respectively.

This process is repeated for all $j = 0, ..., n_{\text{filter}} - 1$ filters, producing a vector of n_{filter} values for each time frame. The final Mel-spectrogram is a matrix formed by stacking the resulting vector from each time frame. It is a representation where each row corresponds to a Mel frequency band and each column corresponds to a time frame.

To compress the dynamic range of the values, we apply logarithmic scaling:

$$x^{\log}_{j,i_{\text{frame}}} = \log_{10}(x^f_{j,i_{\text{frame}}} + \epsilon) \tag{7.14}$$

where $x^f_{j,i_{\text{frame}}}$ is the value of the Mel-spectrogram for the j-th filter at time frame i_{frame}, and ϵ is a small constant to avoid taking the logarithm of zero. This log-Mel-spectrogram is then used as an input feature for our classification models.

7.3.2 Text

Similar to the previous section, we start with a raw audio signal. To compute a linguistic representation, we first transcribe the audio to text using a transcription model, namely Whisper [389]. This results in a string representation s, which is tokenised into sub-word units w, totalling n_s tokens. We then generate an embedding matrix $\mathbf{R}_{\text{text}}$ for the sentence s by passing the tokenised sequence,

represented as a vector of size n_s (with input IDs), through an "encoder-only" transformer model [6] $\mathcal{E}(\cdot)$, which has an embedding size of d_e.

$$\mathbf{R}_{\text{text}} = \mathcal{E}(s), \tag{7.15}$$

where $\mathbf{R}_{\text{text}}$ is the resulting matrix of size $n_s \times d_e$, with n_s corresponding to the number of tokens and d_e to the embedding size of the text encoder $\mathcal{E}$. This matrix is then the input to a pooling function $\mathcal{P}(\cdot)$, such as *max* pooling, *mean* pooling, or *CLS* pooling, which aggregates the matrix along the first dimension. The resulting vector of dimension d_e is the second component of the feature vector used for our classification model.

7.3.3 Complete Model Architecture

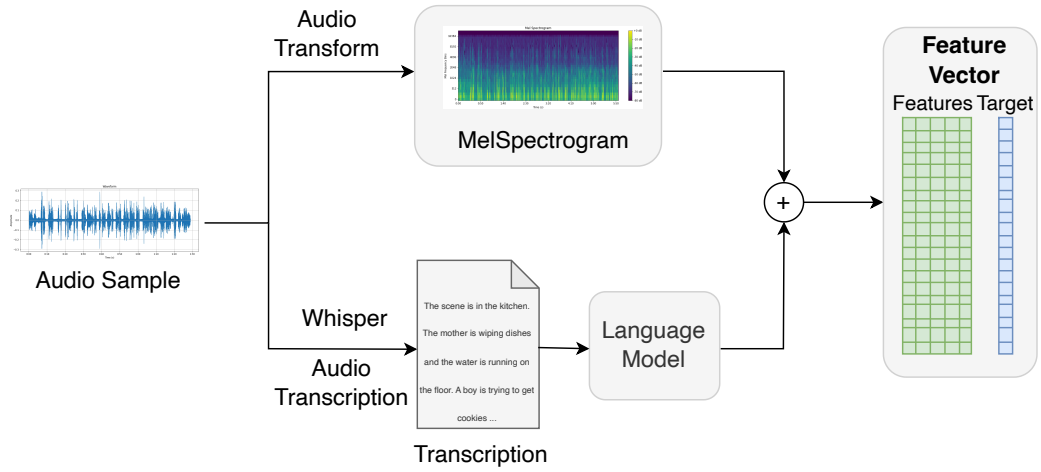
We combine the models described in the previous sections by concatenating both representations and adding a classification head on top of these. We test a random forest and a multilayer perceptron as classifiers. The model architecture and training approach are shown in Figure 7.1.

7.4 Experiments

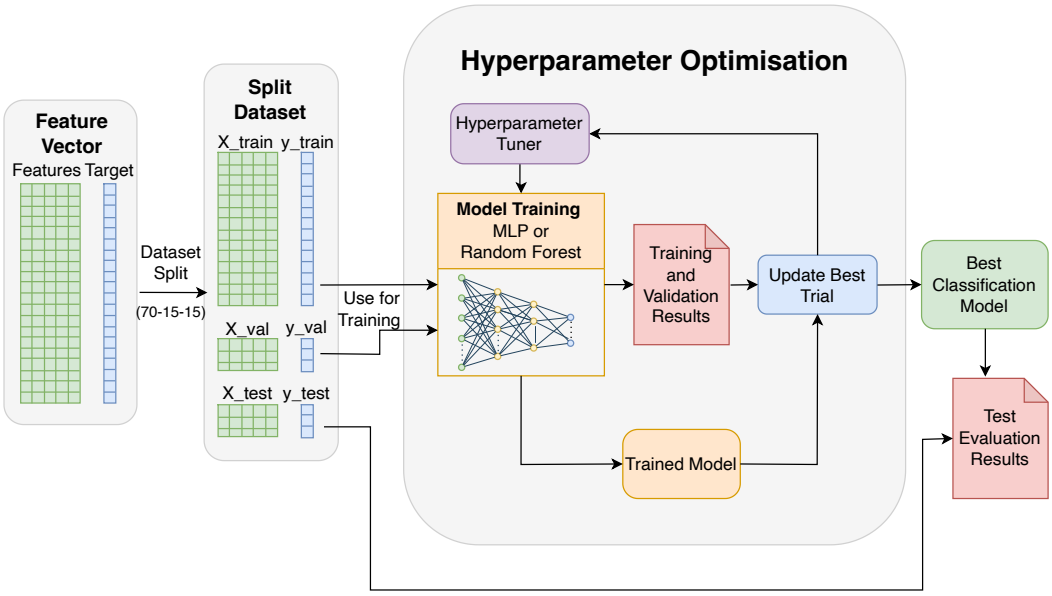
In this section, we outline the experiments conducted, the datasets utilised, and the results obtained. The model architecture employed is described in detail in the previous section. For hyperparameter optimisation, we use Optuna [420], and for the classification task, we apply cross-entropy loss. All experiments were executed on a shared computing cluster equipped with four Nvidia A40 GPUs (48 GB VRAM each), two Intel Xeon 4309Y CPUs, and 400 GB of RAM.

7.4.1 Data

The dataset used in this study is sourced from the Pitt corpus [388], a widely recognised resource from the University of Pittsburgh's Alzheimer's Research Program. It consists of audio recordings of spontaneous speech from participants categorised into two groups: Dementia and Control (healthy). The participants in the study were 44 years or older, had at least seven years of formal education, no history of nervous system disorders, and were not taking neuroleptic medications that could affect cognition. Additionally, all participants had an initial Mini-Mental State Examination (MMSE) score of 10 or higher [400], ensuring that they could provide reliable and meaningful speech data.



(a) Creating a feature vector from an audio sample.



(b) Using the created feature vector to train a model, optimise hyperparameters, and determine the best classification model.

Figure 7.1: Feature vector creation and model training and optimisation. We use Optuna [420] as the framework for hyperparameter optimisation. The figure is adapted from our paper [371].

The dataset includes 99 control participants and 194 dementia patients, each contributing varying numbers of speech recordings across multiple visits. Control participants provided a total of 242 recordings, while dementia patients contributed 307. The demographic distribution of the dataset reflects a balanced representation of males and females, with most participants falling within the 65–70 and 70–75 age ranges. Both groups participated in a variety of tasks, including the Cookie Theft picture description [388], fluency exercises, story recall, and sentence construction. These tasks varied in duration, ranging from brief 1–2 minute activities to extended story recall sessions lasting up to an hour. The diversity of verbal tasks in this dataset enables a comprehensive analysis of cognitive and linguistic functions in both healthy individuals and those with dementia.

7.4.2 Results

To evaluate the various configurations and combinations outlined in Section 7.3 and depicted in Figure 7.1, we conducted a total of 42 experiments. The corresponding results are presented in Table 7.1 and Table 7.2. We began by testing a configuration that utilises only the representation generated by the spectrogram, as described in Equations (7.1) to (7.14). Next, we evaluated five different Whisper models ranging from tiny to large [389] across two transformer architectures, Stella [390] and BERT Base [32], yielding 10 configurations in total. For this, the transcriptions generated by the Whisper models are passed through the corresponding transformer model (see Equation (7.15)) to generate the representations for classification. Finally, we combined the spectrogram-based and language model-based approaches into a multimodal system that integrates both audio and text features. This full model, as illustrated in Figure 7.1, combines the spectrogram with a transcribed text input processed by a language model.

We repeated the entire process using two different classifiers: a random forest (Table 7.1) and a multilayer perceptron (Table 7.2), allowing us to compare performance across these classifier architectures.

As shown in Tables 7.1 and 7.2, our system's performance improves significantly when multimodality is introduced by combining the Mel-spectrogram with a language model using transcribed text. The best-performing configuration, which integrates Stella, Whisper, and the Mel-spectrogram, achieves an impressive F_1-score of 86.42% and an accuracy of 86.59%, substantially outperforming single-modality approaches such as [391], which reported a

Table 7.1: Results in % with a **random forest** as classifier. The table is adapted from our paper [371].

Configuration	Accuracy	Precision	Recall	F_1-Score	AUROC
Mel-Spectrogram	54.88	55.54	57.77	56.63	46.43
BERTBase + ...					
WhisperTiny	68.29	68.81	72.70	70.70	76.67
WhisperBase	78.05	78.39	80.58	79.47	77.86
WhisperSmall	69.51	70.06	74.83	72.37	76.67
WhisperMedium	75.61	75.95	77.95	76.75	80.66
WhisperLarge	74.39	74.76	77.07	75.74	79.82
Stella + ...					
WhisperTiny	76.83	77.14	78.84	77.48	85.00
WhisperBase	78.05	78.33	79.75	79.04	**89.29**
WhisperSmall	75.61	75.95	77.95	76.75	86.55
WhisperMedium	76.83	77.14	78.84	77.48	85.95
WhisperLarge	76.83	77.14	78.84	77.48	87.14
Mel-Spectrogram + BERTBase + ...					
WhisperTiny	69.51	72.56	69.94	71.23	72.92
WhisperBase	78.05	80.58	78.39	79.47	78.45
WhisperSmall	70.73	75.65	71.25	73.38	77.98
WhisperMedium	71.95	74.40	72.32	73.34	78.63
WhisperLarge	73.17	76.19	73.57	74.86	78.45
Mel-Spectrogram + Stella + ...					
WhisperTiny	76.83	79.73	77.20	78.45	84.82
WhisperBase	73.17	76.19	73.57	74.86	**89.29**
WhisperSmall	75.61	77.19	75.89	76.53	86.31
WhisperMedium	73.17	74.08	73.39	73.73	87.32
WhisperLarge	**79.27**	**80.67**	**79.52**	**80.09**	88.93

classification accuracy of 78.70%. Additionally, as expected, configurations using a Multilayer Perceptron proved to be considerably more powerful than those utilising a Random Forest classifier.

7.5 Conclusion

This study introduced a multimodal deep learning approach for dementia detection using the Pitt [388] dataset, which is a collection of audio samples from dementia patients and a control group. By combining audio spectrograms with language models, specifically Whisper [389] for transcription and transformer-based models like BERT [32] and Stella [390] for linguistic feature extraction, we developed a comprehensive framework that captures both acoustic and linguistic markers indicative of dementia.

Our experiments demonstrated that the multimodal approach significantly

Table 7.2: Results in % with a **multilayer perceptron** as classifier. The table is adapted from our paper [371].

Configuration	Accuracy	Precision	Recall	F_1-Score	AUROC
Mel-Spectrogram	60.98	60.30	65.02	62.57	64.35
BERTBase + ...					
WhisperTiny	67.07	67.38	68.47	67.92	74.35
WhisperBase	76.83	77.08	78.14	77.61	76.67
WhisperSmall	73.17	73.45	74.63	74.04	77.68
WhisperMedium	76.83	76.73	76.97	76.85	82.98
WhisperLarge	78.05	78.16	78.29	78.22	80.00
Stella + ...					
WhisperTiny	81.71	81.91	82.58	82.24	86.37
WhisperBase	81.71	81.73	81.71	81.72	**89.94**
WhisperSmall	80.49	80.66	81.10	80.88	86.85
WhisperMedium	81.71	81.85	82.13	81.99	88.93
WhisperLarge	82.93	82.98	82.98	82.98	89.23
Mel-Spectrogram + BERTBase + ...					
WhisperTiny	73.17	73.21	73.21	73.21	79.64
WhisperBase	81.71	81.71	81.73	81.72	79.94
WhisperSmall	79.27	83.60	79.70	81.61	84.76
WhisperMedium	81.71	81.71	81.73	81.72	83.60
WhisperLarge	82.93	82.98	82.98	82.98	83.81
Mel-Spectrogram + Stella + ...					
WhisperTiny	81.71	82.13	81.85	81.99	85.83
WhisperBase	**86.59**	**87.79**	**86.37**	**86.42**	87.98
WhisperSmall	82.93	82.98	82.98	82.98	88.33
WhisperMedium	84.15	84.60	84.29	84.13	89.82
WhisperLarge	85.37	85.42	85.42	85.37	89.35

outperforms single-modality methods. The integration of Whisper transcriptions with a Stella model consistently yielded higher Classification Accuracy and F_1-scores compared to models utilising only audio spectrograms or text features. The best-performing configuration achieved a classification accuracy of 86.59% and an F_1-score of 86.42% on the hold-out test set, marking a substantial improvement over previous studies that relied solely on acoustic or linguistic features.

These results highlight the importance of capturing the multifaceted nature of dementia, which affects both speech patterns and the use of language. By effectively combining acoustic and linguistic features, our approach provides a more complete understanding of the patient's condition, which is crucial for early detection and intervention.

Future work could incorporate another crucial modality into our multimodal system: neuroimaging. A plethora of studies have found it effective in predicting dementia [421–423], which leads us to believe that adding neuroimaging data

to our classification system would improve its predictive power even further. Additionally, evaluating the system's performance in real-world clinical settings will be essential to determine its practical applicability and scalability.

Furthermore, we plan to test various additional configurations, such as adding a Tempogram [424] to include rhythmic analysis or augmenting the audio data, e.g., by time stretching, pitch shifting, and adding background noise, to improve the stability of our models and the size of the dataset. In a similar vein, we plan to utilise and test our system on other corpora available at the DementiaBank [425], possibly in other languages, such as the Spanish or Mandarin corpora introduced in [426] and [427], respectively.

In conclusion, this research demonstrates the efficacy of multimodal deep learning techniques in enhancing the reliability of dementia detection through audio and linguistic analysis. The proposed approach holds promise for developing more robust and accessible diagnostic tools, contributing to earlier interventions and improved patient outcomes.

I am glad you are here with me. Here at the end of all things, Sam.

— J.R.R. Tolkien in *The Return of the King* [3]

8 Conclusion

This thesis, titled *Hybrid Representation Learning for Information Extraction*, has investigated how hybrid approaches, combining data-driven neural architectures with structural, rule-based, and multimodal components, can enhance the robustness, performance, and efficiency of machine learning systems for diverse information extraction tasks like named entity recognition or relation extraction. This final chapter summarises the key findings from the preceding chapters, provides an overall reflection on the contributions of this work, and outlines potential avenues for future research.

8.1 Summary of Findings

To start with, we provided a comprehensive overview of representation learning in Chapter 2. This theoretical foundation introduced the core architectures of neural networks, from multilayer perceptrons and recurrent neural networks to transformer models, and traced the evolution of textual representations from sparse encodings such as Bag-of-Words and TF–IDF to dense, contextualised embeddings derived from transformer-based models.

Chapter 3 addressed the task of contradiction detection in financial reports, a challenge of particular importance in auditing and financial consistency checking, because if these are left uncorrected, they can lead to "bad operational decisions, reputational damage, economic loss, penalties, fines, legal action and even bankruptcy" [131]. Two hybrid methodologies were proposed: one that integrated

linguistic knowledge via part-of-speech-informed pre-training into a transformer-based classifier, and another that combined large language models (LLMs) with embedding-based clustering for document-level inconsistency detection. The results reflect a broader shift in information extraction: before the emergence of LLMs, integrating structured linguistic features and task-specific fine-tuning offered the best performance gains, whereas today, hybrid systems built around LLMs achieve superior flexibility and accuracy.

Chapter 4 incorporated a hybrid approach into named entity recognition, reformulating it as a constrained generative task. Our proposed *Informed Named Entity Recognition Decoding* (iNERD) framework fused the expressive generative capacity of decoder-only LLMs with rule-based decoding constraints to ensure syntactic and factual correctness. Experiments on legal-domain data confirmed that such symbolic-informed constraints markedly improve reliability and domain transferability, highlighting the limitations of purely unconstrained LLM prompting.

Chapter 5 turned to the practical and ethical problem of text anonymisation, a further downstream task of named entity recognition. To mitigate privacy risks associated with cloud-based LLMs, this chapter presented a lightweight anonymisation framework built on knowledge distillation. A large "teacher" model transferred its representational understanding to a compact "student" transformer, which was then integrated with rule-based post-processing. The resulting hybrid system achieved superior recall and overall F_1 scores compared to established baselines, while remaining efficient enough for on-device deployment, thereby aligning performance with data protection requirements like the European General Data Protection Regulation (GDPR).

Building on insights of the named entity recognition chapter, Chapter 6 explored a plethora of relation extraction methods, focusing on the automatic identification of key performance indicators (KPIs) in financial documents. The proposed KPI-BERT model combined BERT-based contextual encoding with recurrent components for joint NER and RE, while a new open-source dataset, KPI-EDGAR, was also introduced to make this task accessible to the whole research community. Regularisation via controlled noise injection improved generalisation, and a novel adjusted F_1 metric better captured the inherent ambiguity of financial text boundaries. We also studied the performance of general-purpose LLMs on this task and found them to be inferior to specialised methods like KPI-BERT. Together, these contributions demonstrated how hybrid and regularised architectures can outperform monolithic neural systems in complex, domain-specific tasks.

Finally, Chapter 7 applied hybrid representation learning to healthcare by investigating multimodal dementia detection. The proposed model fused acoustic representations derived from speech spectrograms with textual representations via models like BERT [32] or Stella [390] obtained from automatic transcriptions. This multimodal fusion outperformed single-modality approaches and established a new state-of-the-art, illustrating that the integration of heterogeneous data sources can yield richer, more discriminative representations for early diagnostic applications.

8.2 General Conclusions

Across all studies presented in this thesis, hybrid representation learning has emerged as a powerful and promising paradigm for information extraction. By uniting data-driven machine learning with structural, symbolic, and multimodal approaches, the proposed methods consistently achieved greater robustness, performance, and practical applicability across multiple domains, including finance, law, privacy, and healthcare. Each empirical contribution demonstrated that the combination of learned representations with additional sources of structure, whether linguistic, logical, or modal, enables models to generalise more effectively and to operate under the complex constraints of real-world data.

A central conclusion of this thesis is that progress in information extraction does not solely depend on ever more scaling model size or data volume, but rather on the intelligent integration of complementary forms of knowledge. While large language models have significantly advanced language understanding [48], they also expose limitations in factual consistency, computational efficiency, and explainability [428]. Hybrid systems, by contrast, mitigate these weaknesses through targeted incorporation of expert knowledge, structured guidance, and additional modalities. In the context of this work, such hybrid approaches were shown to improve both the stability of smaller models and the controllability of larger generative ones, offering a balanced path between performance and reliability.

In this sense, the hybridisation of representation learning bridges the gap between theoretical advances in deep neural architectures and their operational deployment in practice. It provides a promising pathway to balance flexibility with structure, automation with human knowledge, and empirical strength with smaller resource constraints. The results presented throughout this thesis thus support the view that the future of reliable and efficient information extraction likely lies not in ever larger transformer models, but in the fusion of diverse

representations and knowledge, each contributing towards an advancement of artificial intelligence as a whole.

8.3 Outlook and Future Work

Several promising research directions arise from this thesis. Each builds upon the hybrid methodologies introduced in this thesis and extends them into new conceptual or application domains.

Unified hybrid frameworks and multimodal extensions

Future research could investigate the integration of multiple hybridisation strategies within a single cohesive framework. While this thesis explored combinations of symbolic constraints, multimodal fusion, and knowledge distillation in isolation, their unification could yield even more powerful architectures. For instance, a model might incorporate symbolic constraints as well as multimodal reasoning during training and inference, and distillation to ensure efficiency and privacy. Such unified systems might be capable of balancing interpretability, performance, and resource efficiency, leading to privacy-preserving but still high-performing information extraction pipelines suitable for deployment in sensitive domains such as healthcare, law, and finance.

The results in Chapter 7 demonstrated that combining acoustic and linguistic features improved dementia detection performance. A natural extension of this work is to include additional modalities, such as neuroimaging data, physiological signals, or different forms of structured or unstructured text, e.g., the patient's history. Integrating these complementary sources could enable more complete and improved clinical assessments in medical machine learning. Beyond healthcare, similar multimodal fusion principles could benefit applications in robotics, law enforcement, and environmental monitoring, where contextual information from multiple sensors must be synthesised into coherent and task-specific representations.

As digital documents usually combine text, images, tables, and logos, the anonymisation of multimodal content becomes an open challenge within these unified frameworks. Future research could explore how large multimodal models might detect and redact sensitive visual elements, such as faces, signatures, or corporate logos, alongside textual information. A further step would be to distil such multimodal anonymisation capabilities into smaller, locally deployable models, ensuring that privacy preservation remains computationally feasible

without reliance on cloud-based services. This research direction also invites exploration into the joint detection of linked entities across modalities, for example, ensuring that a company name redacted in text is also removed from an accompanying chart or image. Such comprehensive multimodal approaches would exemplify the unified hybrid frameworks envisioned at the outset, combining symbolic constraints, multimodal reasoning, and distillation into integrated systems that address both performance and privacy requirements.

Dynamic constraint learning

Building upon the *iNERD* framework introduced in Chapter 4, an important next step is to move from static rule-based constraints to dynamically learned or adaptive constraints. These could be derived automatically from data with an additional "constraint-learner" model, allowing the system to infer structural rules during training rather than relying solely on human-crafted patterns. Moreover, this concept might extend beyond text: in multimodal or visual tasks, dynamic constraints could guide attention mechanisms or enforce consistency across modalities, for instance, ensuring alignment between image regions and textual descriptions. Exploring such adaptive constraint mechanisms would deepen the understanding of how modern neural systems can internalise and operationalise structure across domains.

Cultural and contextual dimensions in pseudoanonymisation

Another important avenue concerns the inclusion of culture, gender, and other latent sociolinguistic attributes in pseudoanonymisation systems. Future tools might replace named entities not only with neutral placeholders but with culturally and contextually appropriate analogues, for example, substituting a female Elvish name such as *Galadriel* with another female Elvish name like *Arwen*, or a male Hobbit name like *Bilbo* with *Frodo*. Such contextual consistency could preserve narrative structure and semantics in natural language while maintaining anonymity. However, this direction also raises ethical questions about whether such enrichment is desirable or risks incorporating unintended bias. Addressing and exploring these trade-offs will be crucial for the design of socially responsible anonymisation systems.

Hybrid information extraction for retrieval-augmented generation

Retrieval-augmented generation (RAG) systems have become a cornerstone of knowledge-intensive NLP [429, 430]. We see a promising line of research involving enhancing these systems with hybrid information extraction techniques. By extracting structured metadata and performing entity clustering with *iNERD*, finding contradictory statements prior to retrieval, or anonymising data before sending it to remotely hosted LLMs, hybrid models could provide RAG pipelines with more precise, trustworthy, and context-aware retrieval contexts. Such pre-processing may reduce hallucinations, improve factual accuracy, and make retrieval and generation more interpretable and data protection compliant. We theorise that integrating hybrid representations into the retrieval loop therefore represents a promising direction for improving the reliability and explainability of generative systems.

8.4 Closing Words

In summary, this thesis has shown that the future of reliable and responsible information extraction likely lies not in any single methodological paradigm but in the deliberate fusion of multiple forms of intelligence: statistical, symbolic, and human. Hybrid representation learning provides a pathway towards this synthesis of intelligence, enabling systems that are not only more capable but also more trustworthy and aligned with ethical, social, and practical considerations.

Appendices

A
Individual Contributions

In this part of the Appendix, we will be detailing the individual contributions of each author of each paper on which this thesis is based upon. The papers are ordered chronologically by their order of appearance and, if published in the same venue, alphabetically.

KPI-BERT: A Joint Named Entity Recognition and Relation Extraction Model for Financial Reports

For the paper "KPI-BERT: A Joint Named Entity Recognition and Relation Extraction Model for Financial Reports" [11], authored by Lars Hillebrand, Tobias Deußer, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 26th International Conference on Pattern Recognition* in 2022, Tobias Deußer, together with Lars Hillebrand, was responsible for the development and implementation of the KPI-BERT model. The evaluation of experiments and the writing of the paper was also a collaborative effort undertaken by Lars Hillebrand and Tobias Deußer. Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

KPI-EDGAR: A Novel Dataset and Accompanying Metric for Relation Extraction from Financial Documents

For the paper "KPI-EDGAR: A Novel Dataset and Accompanying Metric for Relation Extraction from Financial Documents" [12], authored by Tobias Deußer, Syed Musharraf Ali, Lars Hillebrand, Desiana Nurchalifah, Basil Jacob, Christian

Bauckhage, and Rafet Sifa, published in the *Proceedings of the 21st IEEE International Conference on Machine Learning and Applications* in 2022, Tobias Deußer, was responsible for the concept of the adjusted F_1 metric and led the development of the model architecture. He implemented the majority of the codebase and oversaw the evaluation of experiments as well as the annotation of the dataset. He also wrote most of the paper. Syed Musharraf Ali and Desiana Nurchalifah supported the development of the codebase and the annotation of the dataset. Tobias Deußer, Syed Musharraf Ali, Desiana Nurchalifah, and Basil Jacob annotated the dataset. Lars Hillebrand, Christian Bauckhage, and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

Contradiction Detection in Financial Reports

For the paper "Contradiction Detection in Financial Reports" [128], authored by Tobias Deußer, Maren Pielka, Lisa Pucknat, Basil Jacob, Tim Dilmaghani, Mahdis Nourimand, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 4th Northern Lights Deep Learning Conference* in 2023, Tobias Deußer, in collaboration with Maren Pielka, was responsible for the research of how one can detect contradiction in financial reports with transformer models. The evaluation of experiments and the writing of the paper was also a joint effort undertaken by Tobias Deußer and Maren Pielka. The pre-training and fine-tuning pipeline was originally developed by Lisa Pucknat for a different project and was reused in this work. The data collection and annotation was done by Basil Jacob, Tim Dilmaghani, and Mahdis Nourimand and overseen by Tobias Deußer and Maren Pielka. Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

Controlled Randomness Improves the Performance of Transformer Models

For the paper "Controlled Randomness Improves the Performance of Transformer Models" [322], authored by Tobias Deußer, Cong Zhao, Wolfgang Krämer, David Leonhard, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 22nd IEEE International Conference on Machine Learning and Applications* in 2023, Tobias Deußer was responsible for the main methodology of the work. The implementation and evaluation of the experiments was done by Cong Zhao. Tobias Deußer wrote the manuscript with a few additions from David Leonhard.

Wolfgang Krämer, Christian Bauckhage, and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

Uncovering Inconsistencies and Contradictions in Financial Reports using Large Language Models

For the paper "Uncovering Inconsistencies and Contradictions in Financial Reports using Large Language Models" [129], authored by Tobias Deußer, David Leonhard, Lars Hillebrand, Armin Berger, Mohamed Khaled, Sarah Heiden, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 11th IEEE International Conference on Big Data* in 2023, Tobias Deußer's main responsibilities were the development and research of the contradiction detection and clustering algorithm. The implementation of the codebase and evaluation of experiments was a joint effort undertaken by Tobias Deußer and David Leonhard. Tobias Deußer mainly wrote the manuscript, with contributions from David Leonhard. The data annotation process was done by Mohamed Khaled, Sarah Heiden, and Tim Dilmaghani and overseen by Tobias Deußer. Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

A Comparative Study of Large Language Models for Named Entity Recognition in the Legal Domain

For the paper "A Comparative Study of Large Language Models for Named Entity Recognition in the Legal Domain" [14], authored by Tobias Deußer, Cong Zhao, Lorenz Sparrenberg, Daniel Uedelhoven, Armin Berger, Maren Pielka, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 12th IEEE International Conference on Big Data* in 2024, Tobias Deußer was responsible for the ideation of the named entity recognition process and the prompt design. Cong Zhao implemented the inference pipeline and ran the experiments. The manuscript was written by Tobias Deußer with contributions from Lorenz Sparrenberg. Daniel Uedelhoven took care of the large language model hosting process. Armin Berger, Maren Pielka, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

Fusing speech and language models for dementia detection

For the paper "Fusing speech and language models for dementia detection" [371], authored by Tobias Deußer, Abdul Mohsin Siddiqi, Lorenz Sparrenberg, Tobias Adams, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 12th IEEE International Conference on Big Data* in 2024, Tobias Deußer's main responsibility was the primary research guidance of Abdul Mohsin Siddiqi, who's Master Thesis the paper is based upon. Abdul Mohsin Siddiqi implemented the codebase and ran the experiments. Together with Lorenz Sparrenberg, Tobias Deußer advised on where the research project should head, which direction to avoid, and which machine learning models and architectures should be chosen. Tobias Deußer, with support from Lorenz Sparrenberg and Tobias Adams, wrote the manuscript. Christian Bauckhage and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

Informed Named Entity Recognition Decoding for Generative Language Models

For the paper "Informed Named Entity Recognition Decoding for Generative Language Models" [183], authored by Tobias Deußer, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 12th IEEE International Conference on Big Data* in 2024, Tobias Deußer came up with the *iNERD* architecture, the training and inference pipeline, and implemented the codebase. He also wrote the manuscript. Lars Hillebrand, Christian Bauckhage, and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

Leveraging Large Language Models for Few-Shot KPI Extraction from Financial Reports

For the paper "Leveraging Large Language Models for Few-Shot KPI Extraction from Financial Reports" [323], authored by Tobias Deußer, Cong Zhao, Daniel Uedelhoven, Lorenz Sparrenberg, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 12th IEEE International Conference on Big Data* in 2024, Tobias Deußer was responsible for the ideation of the relation extraction process and the prompt design. Cong Zhao implemented the inference pipeline and ran the experiments. The manuscript was written by Tobias Deußer, with contributions from Lorenz Sparrenberg. Daniel Uedelhoven took care of the large language model hosting process. Lars Hillebrand, Christian Bauckhage, and

Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

Resource-Efficient Anonymization of Textual Data via Knowledge Distillation from Large Language Models

For the paper "Resource-Efficient Anonymization of Textual Data via Knowledge Distillation from Large Language Models" [279], authored by Tobias Deußer, Max Hahnbück, Tobias Uelwer, Cong Zhao, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 31st International Conference on Computational Linguistics* in 2025, Tobias Deußer was the team lead of the "Anonymiser" team and was responsible for the ideation of the anonymisation and knowledge distillation process. He was also the lead developer integrating the codebase, with support from Max Hahnbück, Tobias Uelwer, and Cong Zhao. The manuscript was written by Tobias Deußer, with contributions from Max Hahnbück and Tobias Uelwer. Christian Bauckhage and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

A Survey on Current Trends and Recent Advances in Text Anonymization

For the paper "A Survey on Current Trends and Recent Advances in Text Anonymization" [280], authored by Tobias Deußer, Lorenz Sparrenberg, Armin Berger, Max Hahnbück, Christian Bauckhage, and Rafet Sifa, published in the *Proceedings of the 12th IEEE International Conference on Data Science and Advanced Analytics* in 2025, Tobias Deußer researched potential candidates for the inclusion in the survey paper. He wrote the manuscript, with contributions from Lorenz Sparrenberg, Armin Berger, and Max Hahnbück. Christian Bauckhage and Rafet Sifa provided guidance during the research process, read the manuscript, and gave constructive feedback on it.

B
Tables

B.1 KPI Extraction Prompt

Formatted KPI Extraction Prompt

Extract Key Performance Indicators and their corresponding numerical value from the following sentence in JSON format. The format is: `{"relation_id": {"named_entity_class": "named_entity_value", "named_entity_class": "named_entity_value"}, ...}`. Here `relation_id` can be multiple, such as 0, 1, 2, 3, 4, 5, etc. Ensure that there are exactly two key-value pairs in `{"named_entity_class": "named_entity_value", "named_entity_class": "named_entity_value"}` The value of a key-value pair can be primarily numbers, and if that's not possible, then strings.
Examples are:
1. Example
"Includes $ 6.7 billion of revenue recognized in 2021 that was included in deferred revenue as of September 26, 2020, $ 5.0 billion of revenue recognized in 2020 that was included in deferred revenue as of September 28, 2019, and $ 5.9 billion of revenue recognized in 2019 that was included in deferred revenue as of September 29, 2018." from which you should extract: {"0": {"KPI": "revenue", "value current year": 6.7}, "1": {"KPI": "revenue", "value two years ago": 5.9}, "2": {"KPI": "revenue", "value previous year": 5.0}}.
... (additional examples continue in the same format) ...
Named entity classes are only allowed from the following range: "KPI", "value current year", "value previous year", "value two years ago", "current

year value increase", "previous year value increase", "current year value decrease", "previous year value decrease", "thereof", "attribute", "KPI-Coreference".
Here are the descriptions for the named entity classes:

- "KPI": Key Performance Indicators expressible in numerical and monetary value, e.g. revenue or net sales
- "value current year": Current Year monetary value of a KPI
- "value previous year": Prior Year monetary value of a KPI
- "value two years ago": 2 Year Past Value of a KPI
- "current year value increase": Increase of a KPI from the previous year to the current year
- "previous year value increase": Analogous to increase, but from value two years ago to value previous year
- "current year value decrease": Decrease of a KPI from the previous year to the current year
- "previous year value decrease": Analogous to decrease, but from value two years ago to value previous year
- "thereof": Represents a subordinate KPI, i.e., if a KPI is part of another, broader KPI
- "attribute": Attribute that further describes a KPI
- "KPI-Coreference": A co-reference to a KPI mentioned in a previous sentence

Please follow the examples and description to extract Key Performance Indicators and generate a consistent JSON format.

References

[1] John Ronald Reuel Tolkien. *The Fellowship of the Ring*. Vol. 1. The Lord of the Rings. London: George Allen & Unwin, 1954.

[2] John Ronald Reuel Tolkien. *The Two Towers*. Vol. 2. The Lord of the Rings. London: George Allen & Unwin, 1954.

[3] John Ronald Reuel Tolkien. *The Return of the King*. Vol. 3. The Lord of the Rings. London: George Allen & Unwin, 1955.

[4] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal Loss for Dense Object Detection". In: *Proc. ICCV*. 2017.

[5] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proc. EMNLP*. 2014.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Proc. NeurIPS*. 2017.

[7] Jacob Cohen. "A coefficient of agreement for nominal scales". In: *Educational and psychological measurement* (1960).

[8] F.J. Harris. "On the use of windows for harmonic analysis with the discrete Fourier transform". In: *Proceedings of the IEEE* (1978).

[9] Pierre-Carl Langlais, Carlos Rosas Hinostroza, Mattia Nee, Catherine Arnett, Pavel Chizhov, Eliot Krzystof Jones, Irène Girard, David Mach, Anastasia Stasenko, and Ivan P. Yamshchikov. *Common Corpus: The Largest Collection of Ethical Data for LLM Pre-Training*. 2025. arXiv: 2506.01732 [cs.CL].

[10] Rafet Sifa, Anna Ladi, Maren Pielka, Rajkumar Ramamurthy, Lars Hillebrand, Birgit Kirsch, David Biesner, Robin Stenzel, Thiago Bell, Max Lübbering, et al. "Towards automated auditing with machine learning". In: *Proc. DocEng*. 2019.

[11] Lars Hillebrand, Tobias Deußer, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. "KPI-BERT: A Joint Named Entity Recognition and Relation Extraction Model for Financial Reports". In: *Proc. ICPR*. 2022.

[12] Tobias Deußer, Syed Musharraf Ali, Lars Hillebrand, Desiana Nurchalifah, Basil Jacob, Christian Bauckhage, and Rafet Sifa. "KPI-EDGAR: A Novel Dataset and Accompanying Metric for Relation Extraction from Financial Documents". In: *Proc. ICMLA*. 2022.

[13] Hang Jiang, Xiajie Zhang, Robert Mahari, Daniel Kessler, Eric Ma, Tal August, Irene Li, Alex Pentland, Yoon Kim, Deb Roy, and Jad Kabbara. "Leveraging Large Language Models for Learning Complex Legal Concepts through Storytelling". In: *Proc. ACL*. 2024.

[14] Tobias Deußer, Cong Zhao, Lorenz Sparrenberg, Daniel Uedelhoven, Armin Berger, Maren Pielka, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa. "A Comparative Study of Large Language Models for Named Entity Recognition in the Legal Domain". In: *Proc. BigData*. 2024.

[15] Alan Khoja, Martin Kölbl, Stefan Leue, and Rüdiger Wilhelmi. "Automated consistency analysis for legal contracts". In: *Artificial Intelligence and Law* (2025).

[16] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. "MIMIC-IV, a freely accessible electronic health record dataset". In: *Scientific data* (2023).

[17] Elena Doering, Merle C. Hönig, Tobias Deußer, Gérard N. Bischof, Thilo van Eimeren, Alexander Drzezga, and Lotta M. Ellingsen. "Translating the future: image-to-image translation for the prediction of future brain metabolism". In: *Medical Imaging 2024: Clinical and Biomedical Imaging*. 2024.

[18] Haowei Yang, Ziyu Shen, Junli Shao, Luyao Men, Xinyue Han, and Jing Dong. *LLM-Augmented Symptom Analysis for Cardiovascular Disease Risk Prediction: A Clinical NLP*. 2025. arXiv: 2507.11052 [cs.CL].

[19] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. "The pushshift reddit dataset". In: *Proc. ICWSM*. 2020.

[20] Xiang Deng, Vasilisa Bashlovkina, Feng Han, Simon Baumgartner, and Michael Bendersky. "LLMs to the Moon? Reddit Market Sentiment Analysis with Large Language Models". In: *Proc. WWW*. 2023.

[21] Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Surabhi Adhikari, Hariram Veeramani, Mehwish Nasim, and Usman Naseem. "Large language models (LLM) in computational social science: prospects, current state, and challenges". In: *Social Network Analysis and Mining* (2025).

[22] Jana Sedlakova, Paola Daniore, Andrea Horn Wintsch, Markus Wolf, Mina Stanikic, Christina Haag, Chloé Sieber, Gerold Schneider, Kaspar Staub, Dominik Alois Ettlin, et al. "Challenges and best practices for digital unstructured data enrichment in health research: A systematic narrative review". In: *PLOS Digital Health* (2023).

[23] Evert de Haan, Manjunath Padigar, Siham El Kihal, Raoul Kübler, and Jaap E Wieringa. "Unstructured data research in business: Toward a structured approach". In: *J. of Business Research* (2024).

[24] Hans Eguia, Carlos Luis Sánchez-Bocanegra, Franco Vinciarelli, Fernando Alvarez-Lopez, and Francesc Saigí-Rubió. "Clinical decision support and natural language processing in medicine: systematic literature review". In: *J. of Medical Internet Research* (2024).

[25] Martin Spring, James Faulconbridge, and Atif Sarwar. "How information technology automates and augments processes: Insights from Artificial-Intelligence-based systems in professional service operations". In: *J. of Operations Management* (2022).

[26] Sadaf Zubair. "AI-Driven Automation: Transforming Workplaces and Labor Markets". In: *Frontiers in Artificial Intelligence Research* (2024).

[27] Jerry Hobbs. "FASTUS: A finite-state processor for information extraction from real-world text". In: *Proc. IJCAI*. 1993.

[28] Sunita Sarawagi. "Information extraction". In: *Foundations and Trends in Databases* (2008).

[29] Ralph Grishman. "Twenty-five years of information extraction". In: *Natural Language Engineering* (2019).

[30] Liu Pai, Wenyang Gao, Wenjie Dong, Lin Ai, Ziwei Gong, Songfang Huang, Li Zongsheng, Ehsan Hoque, Julia Hirschberg, and Yue Zhang. "A Survey on Open Information Extraction from Rule-based Model to Large Language Model". In: *Findings of the ACL: EMNLP 2024*. 2024.

[31] Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. "A Survey on Neural Open Information Extraction: Current Status and Future Directions". In: *Proc. IJCAI*. 2022.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proc. NAACL-HLT*. 2019.

[33] John Hewitt and Christopher D. Manning. "A Structural Probe for Finding Syntax in Word Representations". In: *Proc. NAACL*. 2019.

[34] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. "A primer in BERTology: What we know about how BERT works". In: *TACL* (2021).

[35] Sreejan Kumar, Theodore R Sumers, Takateru Yamakoshi, Ariel Goldstein, Uri Hasson, Kenneth A Norman, Thomas L Griffiths, Robert D Hawkins, and Samuel A Nastase. "Shared functional specialization in transformer-based language models and the human brain". In: *Nature communications* (2024).

[36] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* (1989).

[37] Microsoft, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, et al. *Phi-4-Mini Technical Report: Compact yet Powerful Multimodal Language Models via Mixture-of-LoRAs*. 2025. arXiv: 2503.01743 [cs.CL].

[38] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. "Improving Language Understanding by Generative Pre-Training". In: (2018).

[39] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners". In: (2019).

[40] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language Models are Few-Shot Learners". In: *Proc. NeurIPS*. 2020.

[41] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL].

[42] OpenAI. *GPT-5 System Card*. Accessed: 15/09/2025. 2025. URL: https://cdn.openai.com/gpt-5-system-card.pdf.

[43] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2025. arXiv: 2312.11805 [cs.CL].

[44] Armin Berger, Lars Hillebrand, David Leonhard, Tobias Deußer, Thiago Bell Felix De Oliveira, Tim Dilmaghani, Mohamed Khaled, Bernd Kliem, Rudiger Loitz, Christian Bauckhage, et al. "Towards automated regulatory compliance verification in financial auditing with large language models". In: *Proc. BigData*. 2023.

[45] Kaige Xie and Mark Riedl. "Creating Suspenseful Stories: Iterative Planning with Large Language Models". In: *Proc. EACL*. 2024.

[46] Lorenz Sparrenberg, Tobias Schneider, Tobias Deußer, Markus Koppenborg, and Rafet Sifa. "Correcting Systematic Bias in LLM-Generated Dialogues Using Big Five Personality Traits". In: *Proc. BigData*. 2024.

[47] Saptarshi Sengupta, Connor Heaton, Shreya Ghosh, Wenpeng Yin, Preslav Nakov, and Suhang Wang. "TOP-Training: Target-Oriented Pretraining for Medical Extractive Question Answering". In: *Proc. COLING*. 2025.

[48] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. *Large Language Models: A Survey*. 2025. arXiv: 2402.06196 [cs.CL].

[49] Somshubra Majumdar, Vahid Noroozi, Mehrzad Samadi, Sean Narenthiran, Aleksander Ficek, Wasi Uddin Ahmad, Jocelyn Huang, Jagadeesh Balam, and Boris Ginsburg. "Genetic Instruct: Scaling up Synthetic Generation of Coding Instructions for Large Language Models". In: *Proc. ACL*. 2025.

[50] Sotaro Takeshita, Tornike Tsereteli, and Simone Paolo Ponzetto. "GenGO Ultra: an LLM-powered ACL Paper Explorer". In: *Proc. ACL*. 2025.

[51] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. "Detecting hallucinations in large language models using semantic entropy". In: *Nature* (2024).

[52] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions". In: *ACM Transactions on Information Systems* (2025).

[53] Jared Fernandez, Clara Na, Vashisth Tiwari, Yonatan Bisk, Sasha Luccioni, and Emma Strubell. "Energy Considerations of Large Language Model Inference and Efficiency Optimizations". In: *Proc. ACL*. 2025.

[54] Erik Johannes Husom, Arda Goknil, Merve Astekin, Lwin Khin Shar, Andre Kåsen, Sagar Sen, Benedikt Andreas Mithassel, and Ahmet Soylu. "Sustainable llm inference for edge ai: Evaluating quantized llms for energy efficiency, output accuracy, and inference latency". In: *ACM Transactions on Internet of Things* (2025).

[55] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: `1907.11692 [cs.CL]`.

[56] Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L Opdahl, and Csaba Veres. "Named entity extraction for knowledge graphs: A literature overview". In: *IEEE access* (2020).

[57] Isaac Asimov. *I, Robot*. New York: Gnome Press, 1950.

[58] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* (1991).

[59] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. "A neural probabilistic language model". In: *J. of Machine Learning Research* (2003).

[60] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. "Recurrent neural network based language model." In: *Proc. Interspeech*. 2010.

[61] Jeffrey L Elman. "Finding structure in time". In: *Cognitive science* (1990).

[62] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *Proc. ICML*. 2013.

[63] Eric Martin and Chris Cundy. "Parallelizing Linear Recurrent Neural Nets Over Sequence Length". In: *Proc. ICLR*. 2018.

[64] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. "Reformer: The Efficient Transformer". In: *Proc. ICLR*. 2020.

[65] Bohan Zhuang, Jing Liu, Zizheng Pan, Haoyu He, Yuetian Weng, and Chunhua Shen. "A survey on efficient training of transformers". In: *Proc. IJCAI*. 2023.

[66] Zellig S Harris. "Distributional structure". In: *Word* (1954).

[67] Karen Spärck Jones. "A statistical interpretation of term specificity and its application in retrieval". In: *J. of documentation* (1972).

[68] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[69] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: `1301.3781 [cs.CL]`.

[70] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed Representations of Words and Phrases and their Compositionality". In: *Proc. NeurIPS*. 2013.

[71] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: *Proc. EMNLP)*. 2014.

[72] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*. 3rd. 2025.

[73] Michael Hahn. "Theoretical Limitations of Self-Attention in Neural Sequence Models". In: *TACL* (2020).

[74] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[75] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *Nature* (1986).

[76] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proc. ICML*. 2010.

[77] Yann LeCun, Leon Bottou, Genevieve B. Orr, and Klaus -Robert Müller. "Efficient BackProp". In: *Neural Networks: Tricks of the Trade*. 1998.

[78] John S Bridle. "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition". In: *Neurocomputing: Algorithms, architectures and applications*. 1990.

[79] Yoshua Bengio. "Learning deep architectures for AI". In: *Foundations and trends in Machine Learning* (2009).

[80] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2013).

[81] W Jeffrey Johnston and Stefano Fusi. "Abstract representations emerge naturally in neural networks trained to perform multiple tasks". In: *Nature Communications* (2023).

[82] Michael I Jordan. "Attractor dynamics and parallelism in a connectionist sequential machine". In: *Proc. CogSci*. 1986.

[83] Fraser Love. *NNTikZ - TikZ Diagrams for Deep Learning and Neural Networks*. 2024. URL: https://github.com/fraserlove/nntikz.

[84] Sepp Hochreiter. "Untersuchungen zu dynamischen neuronalen Netzen". Diploma. Technische Universität München, 1991.

[85] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* (1994).

[86] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* (1997).

[87] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE].

[88] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies". In: *A Field Guide to Dynamical Recurrent Neural Networks* (2001).

[89] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. "Self-Attention with Relative Position Representations". In: *Proc. NAACL*.

[90] Pu-Chin Chen, Henry Tsai, Srinadh Bhojanapalli, Hyung Won Chung, Yin-Wen Chang, and Chun-Sung Ferng. "A Simple and Effective Positional Encoding for Transformers". In: *Proc. EMNLP*. 2021.

[91] Ofir Press, Noah Smith, and Mike Lewis. "Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation". In: *Proc. ICLR*. 2022.

[92] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. "Roformer: Enhanced transformer with rotary position embedding". In: *Neurocomputing* (2024).

[93] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *Proc. CVPR*. 2016.

[94] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML].

[95] Abdul Wahab and Rafet Sifa. "DIBERT: Dependency injected bidirectional encoder representations from transformers". In: *Proc. SSCI*. 2021.

[96] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL].

[97] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL].

[98] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI].

[99] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *J. of Machine Learning Research* (2020).

[100] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proc. ACL*. 2020.

[101] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2017. arXiv: 1609.04747 [cs.LG].

[102] Paul J Werbos. "Generalization of backpropagation with application to a recurrent gas market model". In: *Neural Networks* (1988).

[103] Danny Kopec, Leonid Shamkovich, and Gabriel Schwartzman. "Deeper Blue Beats Kasparov". In: *Chess Life* (July 1997). URL: https://uscf1-nyc1.aodhosting.com/CL-AND-CR-ALL/CL-ALL/1997/1997_07_5.pdf.

[104] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. "Deep blue". In: *Artificial intelligence* (2002).

[105] Charles Poynton. *Digital Video and HD: Algorithms and Interfaces*. 2nd ed. Morgan Kaufmann/Elsevier, 2012.

[106] American National Standards Institute (ANSI). *American National Standard for Information Systems — Coded Character Sets — 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)*. 1986.

[107] Jindřich Libovický, Helmut Schmid, and Alexander Fraser. "Why don't people use character-level machine translation?" In: *Findings of the ACL: ACL 2022*. 2022.

[108] Ali Araabi, Christof Monz, and Vlad Niculae. "How Effective is Byte Pair Encoding for Out-Of-Vocabulary Words in Neural Machine Translation?" In: *Proc. AMTA*. 2022.

[109] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural Machine Translation of Rare Words with Subword Units". In: *Proc. ACL*. Ed. by Katrin Erk and Noah A. Smith. 2016.

[110] Mike Schuster and Kaisuke Nakajima. "Japanese and korean voice search". In: *Proc. ICASSP*. 2012.

[111] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. "Indexing by latent semantic analysis". In: *J. of the American society for information science* (1990).

[112] Yu-Chen Lin, Si-An Chen, Jie-Jyun Liu, and Chih-Jen Lin. "Linear Classifier: An Often-Forgotten Baseline for Text Classification". In: *Proc. ACL*. 2023.

[113] Geoffrey E Hinton. "Learning distributed representations of concepts". In: *Proc. CogSci*. Vol. 8. 1986.

[114] Frederic Morin and Yoshua Bengio. "Hierarchical probabilistic neural network language model". In: *Proc. AISTATS*. 2005.

[115] Michael U Gutmann and Aapo Hyvärinen. "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics". In: *J of Machine Learning Research* (2012).

[116] Andriy Mnih and Yee Whye Teh. "A fast and simple algorithm for training neural probabilistic language models". In: *Proc. ICML*. 2012.

[117] Ian Tenney, Dipanjan Das, and Ellie Pavlick. "BERT Rediscovers the Classical NLP Pipeline". In: *Proc. ACL*. 2019.

[118] Tiwalayo Eisape, Vineet Gangireddy, Roger Levy, and Yoon Kim. "Probing for Incremental Parse States in Autoregressive Language Models". In: *Findings of the ACL: EMNLP 2022*. 2022.

[119] Dmitry Nikolaev and Sebastian Padó. "Investigating Semantic Subspaces of Transformer Sentence Embeddings through Linear Structural Probing". In: *Proc. Workshop BlackboxNLP*. 2023.

[120] Linyang He, Peili Chen, Ercong Nie, Yuanning Li, and Jonathan R. Brennan. "Decoding Probing: Revealing Internal Linguistic Structures in Neural Language Models Using Minimal Pairs". In: *Proc. LREC-COLING*. 2024.

[121] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. *In-context Learning and Induction Heads*. 2022. arXiv: 2209.11895 [cs.LG].

[122] Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. "Knowledge circuits in pretrained transformers". In: *Proc. NeurIPS* (2024).

[123] Chanyeol Choi, Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, and Jy-yong Sohn. *Linq-Embed-Mistral Technical Report*. 2024. arXiv: 2412.03223 [cs.CL].

[124] Niklas Muennighoff, Hongjin SU, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. "Generative Representational Instruction Tuning". In: *Proc. ICLR*. 2025.

[125] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. *Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models*. 2025. arXiv: 2506.05176 [cs.CL].

[126] George Orwell. *Nineteen Eighty-Four*. London: Secker & Warburg, 1949.

[127] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. "A large annotated corpus for learning natural language inference". In: *Proc. EMNLP*. 2015.

[128] Tobias Deußer, Maren Pielka, Lisa Pucknat, Basil Jacob, Tim Dilmaghani, Mahdis Nourimand, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. "Contradiction detection in financial reports". In: *Proc. NLDL*. 2023.

[129] Tobias Deußer, David Leonhard, Lars Hillebrand, Armin Berger, Mohamed Khaled, Sarah Heiden, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. "Uncovering Inconsistencies and Contradictions in Financial Reports using Large Language Models". In: *Proc. BigData*. 2023.

[130] Henrike Biehl, Christopher Bleibtreu, and Ulrike Stefani. "The real effects of financial reporting: Evidence and suggestions for future research". In: *J. of international accounting, auditing and taxation* (2024).

[131] Kristina Russo. *What Are the Risks of Inaccurate Financial Reporting?* [Online; posted 21/03/2022; retrieved 15/01/2025]. 2022. URL: https://www.netsuite.com/portal/resource/articles/accounting/inaccurate-financial-reporting.shtml.

[132] Lars Hillebrand, Tobias Deußer, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. "Towards automating Numerical Consistency Checks in Financial Reports". In: *Proc. BigData*. 2022.

[133] Yixuan Cao, Hongwei Li, Ping Luo, and Jiaquan Yao. "Towards Automatic Numerical Cross-Checking: Extracting Formulas from Text". In: *Proc. WWW*. 2018.

[134] Steven F Cahan, Chen Chen, and Li Chen. "In financial statements we trust: institutional investors' stockholdings after restatements". In: *The Accounting Review* (2024).

[135] Lars Hillebrand, Armin Berger, Tobias Deußer, Tim Dilmaghani, Mohamed Khaled, Bernd Kliem, Rüdiger Loitz, Maren Pielka, David Leonhard, Christian Bauckhage, and Rafet Sifa. "Improving Zero-Shot Text Matching for Financial Auditing with Large Language Models". In: *Proc. DocEng*. 2023.

[136] Hanchi Gu, Marco Schreyer, Kevin C. Moffitt, and Miklos A. Vasarhelyi. "Artificial Intelligence Co-Piloted Auditing". In: *SSRN Electronic Journal* (2023).

[137] Maren Pielka, Svetlana Schmidt, Lisa Pucknat, and Rafet Sifa. "Towards Linguistically Informed Multi-objective Transformer Pre-training for Natural Language Inference". In: *Proc. ECIR*. 2023.

[138] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. "Adversarial NLI: A New Benchmark for Natural Language Understanding". In: *Proc. ACL*. 2020.

[139] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. "Unsupervised cross-lingual representation learning at scale". In: *Proc. ACL*. 2020.

[140] Ido Dagan, Oren Glickman, and Bernardo Magnini. "The PASCAL Recognising Textual Entailment Challenge". In: *Proc. MLWC*. 2006.

[141] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. "StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding". In: *Proc. ICLR*. 2020.

[142] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. "SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization". In: *Proc. ACL*. 2020.

[143] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *Proc. ICLR*. 2020.

[144] Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. "Muppet: Massive Multi-task Representations with Pre-Finetuning". In: *Proc. EMNLP*. 2021.

[145] Abdul Wahab and Rafet Sifa. "DIBERT: Dependency Injected Bidirectional Encoder Representations from Transformers". In: *Proc. SSCI*. 2021.

[146] Jonathan Pilault, Amine El hattami, and Christopher Pal. "Conditionally Adaptive Multi-Task Learning: Improving Transfer Learning in NLP Using Fewer Parameters & Less Data". In: *Proc. ICLR*. 2021.

[147] Reto Gubelmann, Aikaterini-lida Kalouli, Christina Niklaus, and Siegfried Handschuh. "When Truth Matters - Addressing Pragmatic Categories in Natural Language Inference (NLI) by Large Language Models (LLMs)". In: *Proc. *SEM*. 2023.

[148] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. "Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond". In: *ACM Trans. Knowl. Discov. Data* (2024).

[149] Seo Yeon Park and Cornelia Caragea. "VerifyMatch: A Semi-Supervised Learning Paradigm for Natural Language Inference with Confidence-Aware MixUp". In: *Proc. EMNLP*. 2024.

[150] Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. "Negation, contrast and contradiction in text processing". In: *Proc. AAAI*. Vol. 6. 2006.

[151] Marie-Catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. "Finding contradictions in text". In: *Proc. ACL-HLT*. 2008.

[152] Minh Quang Nhat Pham, Minh Le Nguyen, and Akira Shimazu. "Using Shallow Semantic Parsing and Relation Extraction for Finding Contradiction in Text". In: *Proc. IJCNLP*. 2013.

[153] Noha S Tawfik and Marco R Spruit. "Automated contradiction detection in biomedical literature". In: *Proc. MLDM*. 2018.

[154] Cheng Hsu, Cheng-Te Li, Diego Saez-Trumper, and Yi-Zhan Hsu. "WikiContradiction: Detecting Self-Contradiction Articles on Wikipedia". In: *Proc. Big Data*. 2021.

[155] Di Jin, Sijia Liu, Yang Liu, and Dilek Hakkani-Tur. "Improving Bot Response Contradiction Detection via Utterance Rewriting". In: (2022).

[156] Ala Eddine Kharrat, Lobna Hlaoua, and Lotfi Ben Romdhane. "Contradiction Detection Approach Based on Semantic Relations and Evidence of Uncertainty". In: *Proc. ICCCI*. 2022.

[157] Shivam Sharma, Mirdul Swarup, Tanushri Mahajan, and Zeel Dilipkumar Patel. "Detecting anomalies, contradictions, and contextual analysis through NLP in text". In: *Proc. ICICT*. 2022.

[158] Robiert Sepúlveda-Torres, Alba Bonet-Jover, and Estela Saquete. ""Here Are the Rules: Ignore All Rules": Automatic Contradiction Detection in Spanish". In: *Applied Sciences* 11.7 (2021).

[159] Robiert Sepúlveda-Torres, Alba Bonet-Jover, and Estela Saquete. "Detecting Misleading Headlines Through the Automatic Recognition of Contradiction in Spanish". In: *IEEE Access* 11 (2023).

[160] Yu Takabatake, Hajime Morita, Daisuke Kawahara, Sadao Kurohashi, Ryuichiro Higashinaka, and Yoshihiro Matsuo. "Classification and acquisition of contradictory event pairs using crowdsourcing". In: *Proc. Workshop on EVENTS at NAACL-HLT*. 2015.

[161] Zeinab Rahimi and Mehrnoush ShamsFard. *Contradiction Detection in Persian Text*. 2021. arXiv: 2107.01987 [cs.CL].

[162] Khloud Al Jallad and Nada Ghneim. "ArNLI: Arabic Natural Language Inference for Entailment and Contradiction Detection". In: *Computer Science* 24.2 (2023).

[163] Chujie Zheng, Jinfeng Zhou, Yinhe Zheng, Libiao Peng, Zhen Guo, Wenquan Wu, Zheng-Yu Niu, Hua Wu, and Minlie Huang. "CDConv: A Benchmark for Contradiction Detection in Chinese Conversations". In: *Proc. EMNLP*. 2022.

[164] Rafet Sifa, Maren Pielka, Rajkumar Ramamurthy, Anna Ladi, Lars Hillebrand, and Christian Bauckhage. "Towards contradiction detection in German: a translation-driven approach". In: *Proc. SSCI*. 2019.

[165] Maren Pielka, Rafet Sifa, Lars Patrick Hillebrand, David Biesner, Rajkumar Ramamurthy, Anna Ladi, and Christian Bauckhage. "Tackling contradiction detection in german using machine translation and end-to-end recurrent neural networks". In: *Proc. ICPR*. 2021.

[166] Lisa Pucknat, Maren Pielka, and Rafet Sifa. "Detecting Contradictions in German Text: A Comparative Study". In: *Proc. SSCI*. 2021.

[167] Adina Williams, Nikita Nangia, and Samuel Bowman. "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proc. NAACL*. 2018.

[168] Ahmed Hazourli. "FinancialBERT - A Pretrained Language Model for Financial Text Mining". In: (2022).

[169] Dogu Araci. *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. 2019. arXiv: `1908.10063 [cs.CL]`.

[170] Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. "Good debt or bad debt: Detecting semantic orientations in economic texts". In: *J. of the Association for Information Science and Technology* 65.4 (2014).

[171] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. "spaCy: Industrial-strength natural language processing in python". In: (2020).

[172] Ryan Greene, Ted Sanders, Lilian Weng, and Arvind Neelakantan. *New and improved embedding model*. Accessed 18/01/2025. Dec. 2022. URL: `https://openai.com/blog/new-and-improved-embedding-model`.

[173] T. Cover and P. Hart. "Nearest neighbor pattern classification". In: *IEEE Transactions on Information Theory* (1967).

[174] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *Proc. ICLR*. 2019.

[175] Lisa Milici Gaynor, Andrea Seaton Kelton, Molly Mercer, and Teri Lombardi Yohn. "Understanding the Relation between Financial Reporting Quality and Audit Quality". In: *AUDITING: A Journal of practice & Theory* (2016).

[176] Ricardo Campello, Davoud Moulavi, and Joerg Sander. "Density-Based Clustering Based on Hierarchical Density Estimates". In: *Proc. PAKDD*. 2013.

[177] Cinzia Viroli and Geoffrey J. McLachlan. "Deep Gaussian mixture models". In: *Statistics and Computing* (2019).

[178] Randy D Gordon. "The Sources and Consequences of Disputes over Contractual Meaning". In: *University of Pennsylvania J. of Business Law* (2024).

[179] Patricia L. Cornish, Sandra R. Knowles, Romina Marchesano, Vincent Tam, Steven Shadowitz, David N. Juurlink, and Edward E. Etchells. "Unintended Medication Discrepancies at the Time of Hospital Admission". In: *Archives of Internal Medicine* (2005).

[180] Petra Filkuková, Peter Ayton, Kim Rand, and Johannes Langguth. "What Should I Trust? Individual Differences in Attitudes to Conflicting Information and Misinformation on COVID-19". In: *Frontiers in Psychology* (2021).

[181] Homer. Ὀδύσσεια. Ancient Greek original text of the Odyssey. circa 8th–7th century BCE.

[182] Alexander Pope, William Broome, and Elijah Fenton. *The Odyssey of Homer; Translated from the Greek*. Vol. 2. Bernard Lintot, 1725.

[183] Tobias Deußer, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa. "Informed Named Entity Recognition Decoding for Generative Language Models". In: *Proc. BigData*. 2024.

[184] Wahab Khan, Ali Daud, Khairullah Khan, Shakoor Muhammad, and Rafiul Haq. "Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends". In: *Natural Language Processing Journal* (2023).

[185] Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. "A Paradigm Shift in Machine Translation: Boosting Translation Performance of Large Language Models". In: *Proc. ICLR*. 2024.

[186] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. *Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference*. 2024. arXiv: 2403.04132 [cs.AI].

[187] Wenxuan Zhou and Muhao Chen. "Learning from Noisy Labels for Entity-Centric Information Extraction". In: *Proc. EMNLP*. 2021.

[188] Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. "Packed Levitated Marker for Entity and Relation Extraction". In: *Proc. ACL*. 2022.

[189] Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne P Bernard. "NuNER: Entity Recognition Encoder Pre-training via LLM-Annotated Data". In: *Proc. EMNLP*. 2024.

[190] Kohei Tsuji, Tatsuya Hiraoka, Yuchang Cheng, and Tomoya Iwakura. "SubRegWeigh: Effective and Efficient Annotation Weighing with Subword Regularization". In: *Proc. COLING*. 2025.

[191] Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. "LLMs Know More Than They Show: On the Intrinsic Representation of LLM Hallucinations". In: *Proc. ICLR*. 2025.

[192] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. "Emergent Abilities of Large Language Models". In: *Transactions on Machine Learning Research* (2022).

[193] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. "Generating Wikipedia by Summarizing Long Sequences". In: *Proc. ICLR*. 2018.

[194] Franz A. Heinsen. *An Algorithm for Routing Vectors in Sequences*. 2022. arXiv: 2211.11754 [cs.LG].

[195] Harsh Verma, Sabine Bergler, and Narjesossadat Tahaei. "Comparing and combining some popular NER approaches on Biomedical tasks". In: *Proc. BioNLP*. 2023.

[196] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. *A Survey of Large Language Models*. 2023. arXiv: 2303.18223 [cs.CL].

[197] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. *A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity*. 2023. arXiv: 2302.04023 [cs.CL].

[198] Ralph Grishman and Beth Sundheim. "Message Understanding Conference- 6: A Brief History". In: *Proc. COLING*. 1996.

[199] Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. "The Text Anonymization Benchmark (TAB): A Dedicated Corpus and Evaluation Framework for Text Anonymization". In: *Computational Linguistics* (2022).

[200] Erik F. Tjong Kim Sang and Fien De Meulder. "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". In: *Proc. Conf. on Natural Language Learning at HLT-NAACL*. 2003.

[201] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. "Unsupervised named-entity extraction from the Web: An experimental study". In: *Artificial Intelligence* (2005).

[202] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. "Nymble: a High-Performance Learning Name-finder". In: *Proc. Applied Natural Language Processing*. 1997.

[203] Michael Collins and Yoram Singer. "Unsupervised Models for Named Entity Classification". In: *Proc. EMNLP*. 1999.

[204] Paul McNamee and James Mayfield. "Entity Extraction without Language-Specific Resources". In: *Proc. COLING*. 2002.

[205] Shaodian Zhang and Noémie Elhadad. "Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts". In: *J. of Biomedical Informatics* (2013).

[206] Ying Luo, Fengshun Xiao, and Zhao Hai. "Hierarchical Contextualized Representation for Named Entity Recognition". In: *Proc. AAAI*. 2020.

[207] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* (2019).

[208] Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. "Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning". In: *Proc. ACL-IJCNLP*. 2021.

[209] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. "LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention". In: *Proc. EMNLP*. 2020.

[210] Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. "A Unified Generative Framework for Various NER Subtasks". In: *Proc. ACL-IJCNLP*. 2021.

[211] Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. "LasUIE: Unifying Information Extraction with Latent Adaptive Structure-aware Generative Language Model". In: *Proc. NeurIPS*. 2022.

[212] Tianyu Liu, Yuchen Eleanor Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. "Autoregressive Structured Prediction with Language Models". In: *Findings of the ACL: EMNLP 2022*. 2022.

[213] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. "An Autoregressive Text-to-Graph Framework for Joint Entity and Relation Extraction". In: *Proc. AAAI*. 2024.

[214] Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Ma Jie, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, Stefano Soatto, et al. "Structured prediction as translation between augmented natural languages". In: *Proc. ICLR*. 2021.

[215] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. "Template-Based Named Entity Recognition Using BART". In: *Findings of the ACL: ACL-IJCNLP 2021*. 2021.

[216] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. *GPT-NER: Named Entity Recognition via Large Language Models*. 2023. arXiv: 2304.10428 [cs.CL].

[217] Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. "GenIE: Generative Information Extraction". In: *Proc. NAACL-HLT*. 2022.

[218] Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. "Text2Event: Controllable Sequence-to-Structure Generation for End-to-end Event Extraction". In: *Proc. ACL-IJCNLP*. 2021.

[219] Timo Schick and Hinrich Schütze. "Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference". In: *Proc. EACL*. 2021.

[220] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. "A Survey on Evaluation of Large Language Models". In: *ACM Transactions on Intelligent Systems and Technology* (2024).

[221] Jasper Roe and Mike Perkins. "'What they're not telling you about ChatGPT': exploring the discourse of AI in UK news media headlines". In: *Humanities and Social Sciences Communications volume* (2023).

[222] Reuben Ng and Ting Yu Joanne Chow. "Powerful tool or too powerful? Early public discourse about ChatGPT across 4 million tweets". In: *PLoS ONE* (2024).

[223] Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. "RedPajama: an Open Dataset for Training Large Language Models". In: *Proc. NeurIPS*. 2024.

[224] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. *Mixtral of Experts*. 2024. arXiv: 2401.04088 [cs.LG].

[225] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. *Scaling Language Models: Methods, Analysis & Insights from Training Gopher*. 2022. arXiv: 2112.11446 [cs.CL].

[226] Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. *Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model*. 2022. arXiv: 2201.11990 [cs.CL].

[227] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. "PaLM: Scaling Language Modeling with Pathways". In: *J. of Machine Learning Research* (2023).

[228] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. "Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model". In: *J. of Machine Learning Research* (2023).

[229] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL].

[230] Mozhi Zhang, Hang Yan, Yaqian Zhou, and Xipeng Qiu. *PromptNER: A Prompting Method for Few-shot Named Entity Recognition via k Nearest Neighbor Search*. 2023. arXiv: 2305.12217 [cs.CL].

[231] Yan Hu, Qingyu Chen, Jingcheng Du, Xueqing Peng, Vipina Kuttichi Keloth, Xu Zuo, Yujia Zhou, Zehan Li, Xiaoqian Jiang, Zhiyong Lu, et al. "Improving large language models for clinical named entity recognition via prompt engineering". In: *J. of the American Medical Informatics Association* (2024).

[232] Chenxiao Dou, Xianghui Sun, Yaoshu Wang, Yunjie Ji, Baochang Ma, and Xiangang Li. "Domain-Adapted Dependency Parsing for Cross-Domain Named Entity Recognition". In: *Proc. AAAI*. 2023.

[233] Ngoc Dang Nguyen, Wei Tan, Wray L. Buntine, Richard Beare, Changyou Chen, and Lan Du. "AUC Maximization for Low-Resource Named Entity Recognition". In: *Proc. AAAI*. 2023.

[234] Lance Ramshaw and Mitch Marcus. "Text Chunking using Transformation-Based Learning". In: *Proc. Workshop on Very Large Corpora*. 1995.

[235] Ronald J. Williams and David Zipser. "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". In: *Neural Computation* (1989).

[236] Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, and Christopher D. Manning. *BioMedLM: A 2.7B Parameter Language Model Trained On Biomedical Text*. 2024. arXiv: 2403.18421 [cs.CL].

[237] Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. "CrossWeigh: Training Named Entity Tagger from Imperfect Annotations". In: *Proc. EMNLP-IJCNLP*. 2019.

[238] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. "Towards Robust Linguistic Analysis using OntoNotes". In: *Proc. CoNLL*. 2013.

[239] Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. "Few-NERD: A Few-shot Named Entity Recognition Dataset". In: *Proc. ACL-IJCNLP*. 2021.

[240] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. "Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition". In: *Proc. Workshop on Noisy User-generated Text*. 2017.

[241] Nigel Collier, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Jin-Dong Kim. "Introduction to the Bio-entity Recognition Task at JNLPBA". In: *Proc. NLPBA/BioNLP*. 2004.

[242] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. "NCBI Disease Corpus: A Resource for Disease Name Recognition and Concept Normalization". In: *J. of Biomedical Informatics* (2014).

[243] Agam Shah, Ruchit Vithani, Abhinav Gullapalli, and Sudheer Chava. *FiNER: Financial Named Entity Recognition Dataset and Weak-Supervision Model*. 2023. arXiv: 2302.11157 [cs.CL].

[244] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* (2020).

[245] Albert Gu and Tri Dao. "Mamba: Linear-Time Sequence Modeling with Selective State Spaces". In: *Proc. COLM*. 2024.

[246] Brandon T. Willard and Rémi Louf. *Efficient Guided Generation for Large Language Models*. 2023. arXiv: 2307.09702 [cs.CL].

[247] Dennis Kurzon. "Foreign and archaic phrases in legal texts". In: *The International J. of Speech, Language and the Law* (2013).

[248] Stefan Th Gries and Brian G Slocum. "Ordinary Meaning and Corpus Linguistics". In: *BYU Law Review* (2017).

[249] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. *GPT-NER: Named Entity Recognition via Large Language Models*. 2023. arXiv: 2304.10428 [cs.CL].

[250] Jinyuan Li, Han Li, Di Sun, Jiahao Wang, Wenkun Zhang, Zan Wang, and Gang Pan. "LLMs as Bridges: Reformulating Grounded Multimodal Named Entity Recognition". In: *Findings of the ACL: ACL*. 2024.

[251] Vipina K Keloth, Yan Hu, Qianqian Xie, Xueqing Peng, Yan Wang, Andrew Zheng, Melih Selek, Kalpana Raja, Chih Hsuan Wei, Qiao Jin, et al. "Advancing entity recognition in biomedicine via instruction tuning of large language models". In: *Bioinformatics* (2024).

[252] Yuzhao Heng, Chunyuan Deng, Yitong Li, Yue Yu, Yinghao Li, Rongzhi Zhang, and Chao Zhang. "ProgGen: Generating Named Entity Recognition Datasets Step-by-step with Self-Reflexive Large Language Models". In: *Findings of the ACL: ACL*. 2024.

[253] Bipesh Subedi, Sunil Regmi, Bal Krishna Bal, and Praveen Acharya. "Exploring the Potential of Large Language Models (LLMs) for Low-resource Languages: A Study on Named-Entity Recognition (NER) and Part-Of-Speech (POS) Tagging for Nepali Language". In: *Proc. LREC-COLING*. 2024.

[254] Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. "Applicability of large language models and generative models for legal case judgement summarization". In: *Artificial Intelligence and Law* (2024).

[255] Lauren Martin, Nick Whitehouse, Stephanie Yiu, Lizzie Catterson, and Rivindu Perera. *Better Call GPT, Comparing Large Language Models Against Lawyers*. 2024. arXiv: `2401.16212 [cs.CY]`.

[256] Lars Hillebrand, Maren Pielka, David Leonhard, Tobias Deußer, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Milad Morad, Christian Temath, Thiago Bell, et al. "sustain. AI: a Recommender System to analyze Sustainability Reports". In: *Proc. ICAIL*. 2023.

[257] Joanna Zhao and Xinruo Wang. "Unleashing efficiency and insights: Exploring the potential applications and challenges of ChatGPT in accounting". In: *J. of Corporate Accounting & Finance* (2024).

[258] Kwok-Yan Lam, Victor CW Cheng, and Zee Kin Yeong. "Applying Large Language Models for Enhancing Contract Drafting." In: *Proc. Workshop LegalAIIA at ICAIL*. 2023.

[259] Dor Bernsohn, Gil Semo, Yaron Vazana, Gila Hayat, Ben Hagag, Joel Niklaus, Rohit Saha, and Kyryl Truskovskyi. "LegalLens: Leveraging LLMs for Legal Violation Identification in Unstructured Text". In: *Proc. EACL*. 2024.

[260] Pedro H. Luz de Araujo, Teófilo E. de Campos, Renato R. R. de Oliveira, Matheus Stauffer, Samuel Couto, and Paulo Bermejo. "LeNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text". In: *Proc. PROPOR*. 2018.

[261] Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. "Named Entity Recognition in Indian court judgments". In: *Proc. Workshop NLLP*. 2022.

[262] Vasile Pais, Maria Mitrofan, Carol Luca Gasan, Vlad Coneschi, and Alexandru Ianov. "Named Entity Recognition in the Romanian Legal Domain". In: *Proc. Workshop NLLP*. 2021.

[263] Can Çetindağ, Berkay Yazıcıoğlu, and Aykut Koç. "Named-entity recognition in Turkish legal texts". In: *Natural Language Engineering* (2023).

[264] Ting Wai Terence Au, Vasileios Lampos, and Ingemar Cox. "E-NER — An Annotated Named Entity Recognition Corpus of Legal Text". In: *Proc. Workshop NLLP*. 2022.

[265] Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. "A Dataset of German Legal Documents for Named Entity Recognition". In: *Proc. LREC*. 2020.

[266] Iosif Angelidis, Ilias Chalkidis, and Manolis Koubarakis. "Named entity recognition, linking and generation for greek legislation". In: *Legal Knowledge and Information Systems*. 2018.

[267] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

[268] Thomas Proisl and Peter Uhrig. "SoMaJo: State-of-the-art tokenization for German web and social media texts". In: *Proc. WAC-X*. 2016.

[269] OpenAI. *GPT-4o mini: Advancing cost-efficient intelligence*. OpenAI Blog. Accessed: 25/06/2025. 2024. URL: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

[270] Mistral AI Team. *Mistral Large*. Accessed: 19/03/2025. 2024. URL: https://mistral.ai/news/mistral-large/.

[271] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. *Qwen2 Technical Report*. 2024. arXiv: 2407.10671 [cs.CL].

[272] Thomas Mesnard Gemma Team, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, and Léonard Hussenot and. "Gemma". In: (2024).

[273] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, et al. *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone*. 2024. arXiv: 2404.14219 [cs.CL].

[274] Yining Huang, Keke Tang, and Meilian Chen. *Leveraging Large Language Models for Enhanced NLP Task Performance through Knowledge Distillation and Optimized Training Strategies*. 2024. arXiv: 2402.09282 [cs.CL].

[275] Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. "UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition". In: *Proc. ICLR*. 2024.

[276] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, et al. *Phi-4 Technical Report*. 2024. arXiv: 2412.08905 [cs.CL].

[277] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. *Janus-Pro: Unified Multimodal Understanding and Generation with Data and Model Scaling*. 2025. arXiv: 2501.17811 [cs.AI].

[278] *Grundgesetz für die Bundesrepublik Deutschland*. Bonn, 1949.

[279] Tobias Deußer, Max Hahnbück, Tobias Uelwer, Cong Zhao, Christian Bauckhage, and Rafet Sifa. "Resource-Efficient Anonymization of Textual Data via Knowledge Distillation from Large Language Models". In: *Proc. COLING*. 2025.

[280] Tobias Deußer, Lorenz Sparrenberg, Armin Berger, Max Hahnbück, Christian Bauckhage, and Rafet Sifa. "A Survey on Current Trends and Recent Advances in Text Anonymization". In: *Proc. DSAA*. 2025.

[281] Zheming Zuo, Matthew Watson, David Budgen, Robert Hall, Chris Kennelly, and Noura Al Moubayed. "Data anonymization for pervasive health care: systematic literature mapping study". In: *JMIR medical informatics* (2021).

[282] Stella Dimopoulou, Chrysostomos Symvoulidis, Konstantinos Koutsoukos, Athanasios Kiourtis, Argyro Mavrogiorgou, and Dimosthenis Kyriazis. "Mobile Anonymization and Pseudonymization of Structured Health Data for Research". In: *Proc. MobiSecServ*. 2022.

[283] Gergely Márk Csányi, Dániel Nagy, Renátó Vági, János Pál Vadász, and Tamás Orosz. "Challenges and open problems of legal document anonymization". In: *Symmetry* (2021).

[284] Ingo Glaser, Tom Schamberger, and Florian Matthes. "Anonymization of german legal court rulings". In: *Proc. ICAIL*. 2021.

[285] Lelio Campanile, Maria Stella de Biase, Stefano Marrone, Fiammetta Marulli, Mariapia Raimondo, and Laura Verde. "Sensitive Information Detection Adopting Named Entity Recognition: A Proposed Methodology". In: *Proc. Workshop ICCSA*. 2022.

[286] David Biesner, Rajkumar Ramamurthy, Robin Stenzel, Max Lübbering, Lars Hillebrand, Anna Ladi, Maren Pielka, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. "Anonymization of German financial documents using neural network-based language models with contextual word representations". In: *International J. of Data Science and Analytics* (2022).

[287] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. "Privacy risks of general-purpose language models". In: *IEEE Symposium on Security and Privacy*. 2020.

[288] Xiaodong Wu, Ran Duan, and Jianbing Ni. "Unveiling security, privacy, and ethical concerns of ChatGPT". In: *J. of Information and Intelligence* (2024).

[289] OECD. *Productivity and unit labour cost by industry, ISIC Rev. 4*. 2014. URL: https://www.oecd-ilibrary.org/content/data/data-00687-en.

[290] Omri Mendels, Coby Peled, Nava Vaisman Levy, Sharon Hart, Tomer Rosenthal, Limor Lahiani, et al. *Microsoft Presidio: Context aware, pluggable and customizable PII anonymization service for text and images*. Microsoft, 2018. URL: https://microsoft.github.io/presidio.

[291] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. "GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer". In: *Proc. NAACL*. 2024.

[292] Latanya Sweeney. "Replacing personally-identifying information in medical records, the Scrub system." In: *Proc. AMIA*. 1996.

[293] Filip Graliński, Krzysztof Jassem, Michał Marcińczuk, and Paweł Wawrzyniak. "Named entity recognition in machine anonymization". In: *Recent Advances in Intelligent Information Systems* (2009).

[294] Özlem Uzuner, Yuan Luo, and Peter Szolovits. "Evaluating the state-of-the-art in automatic de-identification". In: *J. of the American Medical Informatics Association* (2007).

[295] Zengjian Liu, Yangxin Chen, Buzhou Tang, Xiaolong Wang, Qingcai Chen, Haodi Li, Jingfeng Wang, Qiwen Deng, and Suisong Zhu. "Automatic de-identification of electronic medical records using token-level and character-level conditional random fields". In: *J. of Biomedical Informatics* (2015).

[296] Jason PC Chiu and Eric Nichols. "Named entity recognition with bidirectional LSTM-CNNs". In: *TACL* (2016).

[297] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. "De-identification of patient notes with recurrent neural networks". In: *J. of the American Medical Informatics Association* (2017).

[298] Arttu Oksanen, Eero Hyvönen, Minna Tamper, Jouni Tuominen, Henna Ylimaa, Katja Löytynoja, Matti Kokkonen, and Aki Hietanen. "An anonymization tool for open data publication of legal documents". In: *Proc. Workshop AI4LEGAL*. 2022.

[299] Bennett Kleinberg, Toby Davies, and Maximilian Mozes. *Textwash – automated open-source text anonymisation*. 2022. arXiv: 2208.13081 [cs.CL].

[300] Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. "Language Models are Advanced Anonymizers". In: *Proc. ICLR*. 2025.

[301] Constantinos Patsakis and Nikolaos Lykousas. "Man vs the Machine in the Struggle for Effective Text Anonymisation in the Age of Large Language Models". In: *Scientific Reports* (2023).

[302] Aleksandar Kovačević, Bojana Bašaragin, Nikola Milošević, and Goran Nenadić. "De-identification of clinical free text using natural language processing: A systematic review of current approaches". In: *Artificial intelligence in medicine* (2024).

[303] Zhengliang Liu, Yue Huang, Xiaowei Yu, Lu Zhang, Zihao Wu, Chao Cao, Haixing Dai, Lin Zhao, Yiwei Li, Peng Shu, Fang Zeng, Lichao Sun, Wei Liu, Dinggang Shen, Quanzheng Li, Tianming Liu, Dajiang Zhu, and Xiang Li. *DeID-GPT: Zero-shot Medical Text De-Identification by GPT-4*. 2023. arXiv: 2303.11032 [cs.CL].

[304] Gergely M. Csányi, Dániel Nagy, Renátó Vági, J. Pál Vadász, and Tamás Orosz. "Challenges and Open Problems of Legal Document Anonymization". In: *Symmetry* (2021).

[305] Kalliopi Terzidou. "Automated Anonymization of Court Decisions: Facilitating the Publication of Court Decisions through Algorithmic Systems". In: *Proc. ICAIL*. 2023.

[306] Benet Manzanares-Salor and David Sánchez. "Enhancing Text Anonymization via Re-Identification Risk-Based Explainability". In: *Knowledge-Based Systems* (2025).

[307] Timour Igamberdiev and Ivan Habernal. "DP-BART for Privatized Text Rewriting under Local Differential Privacy". In: *Findings of the ACL: ACL 2023*. 2023.

[308] Stephen Meisenbacher, Maulik Chevli, Juraj Vladika, and Florian Matthes. "DP-MLM: Differentially Private Text Rewriting Using Masked Language Models". In: *Findings of the ACL: ACL 2024*. 2024.

[309] Vladimir Panov, Mikhail Kovalchuk, Anastasiia Filatova, and Sergey Teryoshkin. "MuCAAT: Multilingual Contextualized Authorship Anonymization of Texts from Social Networks". In: *Procedia Computer Science* (2022).

[310] Gabriel Loiseau, Damien Sileo, Damien Riquet, Maxime Meyer, and Marc Tommasi. "TAROT: Task-Oriented Authorship Obfuscation Using Policy Optimization Methods". In: *Proc. Workshop PrivateNLP*. 2025.

[311] Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. "The Text Anonymization Benchmark (TAB): A Dedicated Corpus and Evaluation Framework for Text Anonymization". In: *Computational Linguistics* (2022).

[312] Oleksandr Yermilov, Vipul Raheja, and Artem Chernodub. "Privacy- and Utility-Preserving NLP with Anonymized Data: A Case Study of Pseudonymization". In: *Proc. Workshop TrustNLP*. 2023.

[313] Michael Stonebraker and Lawrence A Rowe. "The design of Postgres". In: *ACM Sigmod Record* (1986).

[314] Lars Hillebrand, Prabhupad Pradhan, Christian Bauckhage, and Rafet Sifa. "Pointer-Guided Pre-training: Infusing Large Language Models with Paragraph-Level Contextual Awareness". In: *Proc. ECML-PKDD*. 2024.

[315] Dietmar Schabus, Marcin Skowron, and Martin Trapp. "One Million Posts: A Data Set of German Online Discussions". In: *Proc. SIGIR*. 2017.

[316] Elin Törnquist and Robert Alexander Caulk. *Curating Grounded Synthetic Data with Global Perspectives for Equitable AI*. 2024. arXiv: `2406.10258 [cs.CL]`.

[317] Asahi Ushio and Jose Camacho-Collados. "T-NER: An All-Round Python Library for Transformer-based Named Entity Recognition". In: *Proc. EACL*. 2021.

[318] Alex Watson, Yev Meyer, Maarten Van Segbroeck, Matthew Grossman, Sami Torbey, Piotr Mlocek, and Johnny Greco. *Synthetic-PII-Financial-Documents-North-America: A synthetic dataset for training language models to label and detect PII in domain specific formats*. 2024. URL: `https://huggingface.co/datasets/gretelai/synthetic_pii_finance_multilingual`.

[319] Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. "TrueTeacher: Learning Factual Consistency Evaluation with Large Language Models". In: *Proc. EMNLP*. 2023.

[320] Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. *Open LLM Leaderboard v2*. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`. 2024.

[321] Andy Weir. *Project Hail Mary*. New York: Ballantine Books, 2021.

[322] Tobias Deußer, Cong Zhao, Wolfgang Krämer, David Leonhard, Christian Bauckhage, and Rafet Sifa. "Controlled Randomness Improves the Performance of Transformer Models". In: *Proc. ICMLA*. 2023.

[323] Tobias Deußer, Cong Zhao, Daniel Uedelhoven, Lorenz Sparrenberg, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa. "Leveraging Large Language Models for Few-Shot KPI Extraction from Financial Reports". In: *Proc. BigData*. 2024.

[324] Bernard Marr. *Key Performance Indicators (KPI): The 75 measures every manager needs to know*. Pearson UK, 2012.

[325] Hans-Ulrich Krause and Dayanand Arora. *Controlling-Kennzahlen-key performance indicators*. Oldenbourg Wissenschaftsverlag, 2009.

[326] Eduardo Brito, Rafet Sifa, Christian Bauckhage, Rüdiger Loitz, Uwe Lohmeier, and Christin Pünt. "A Hybrid AI Tool to Extract Key Performance Indicators from Financial Reports for Benchmarking". In: *Proc. DocEng*. 2019.

[327] Dimitra Farmakiotou, Vangelis Karkaletsis, John Koutsias, George Sigletos, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. "Rule-based named entity recognition for Greek financial texts". In: *Proc. COMLEX*. 2000.

[328] Markus Eberts and A. Ulges. "Span-based Joint Entity and Relation Extraction with Transformer Pre-training". In: *Proc. ECAI*. 2020.

[329] Zhiheng Huang, Wei Xu, and Kai Yu. *Bidirectional LSTM-CRF Models for Sequence Tagging*. 2015. arXiv: `1508.01991 [cs.CL]`.

[330] Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. "Let's Stop Incorrect Comparisons in End-to-end Relation Extraction!" In: *Proc. EMNLP*. 2020.

[331] Katrin Fundel, Robert Küffner, and Ralf Zimmer. "RelEx—Relation extraction using dependency parse trees". In: *Bioinformatics* (2007).

[332] Harsha Gurulingappa, Abdul Mateen-Rajpu, and Luca Toldo. "Extraction of potential adverse drug events from medical case reports". In: *J. of Biomedical Semantics* 3.1 (2012).

[333] Qi Li and Heng Ji. "Incremental Joint Extraction of Entity Mentions and Relations". In: *Proc. ACL*. 2014.

[334] Aasish Pappu, Roi Blanco, Yashar Mehdad, Amanda Stent, and Kapil Thadani. "Lightweight Multilingual Entity Extraction and Linking". In: *Proc. WSDM*. 2017.

[335] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. "BOND: BERT-Assisted Open-Domain Named Entity Recognition with Distant Supervision". In: *Proc. KDD*. 2020.

[336] Jue Wang and Wei Lu. "Two Are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders". In: *Proc. EMNLP*. 2020.

[337] Yongliang Shen, Xinyin Ma, Yechun Tang, and Weiming Lu. "A Trigger-Sense Memory Flow Framework for Joint Entity and Relation Extraction". In: *Proc. Web Conference*. 2021.

[338] Van-Hien Tran, Van-Thuy Phi, Akihiko Kato, Hiroyuki Shindo, Taro Watanabe, and Yuji Matsumoto. "Improved Decomposition Strategy for Joint Entity and Relation Extraction". In: *J. of NLP* 28.4 (2021).

[339] Zexuan Zhong and Danqi Chen. "A Frustratingly Easy Approach for Entity and Relation Extraction". In: *Proc. NAACL*. 2021.

[340] Tingting Hang, Jun Feng, Le Yan, Yunfeng Wang, and Jiamin Lu. "Joint extraction of entities and relations using multi-label tagging and relational alignment". In: *Neural. Comput. Appl.* (2022).

[341] Agata Savary, Alena Silvanovich, Anne-Lyse Minard, Nicolas Hiot, and Mirian Halfeld Ferrari. "Relation Extraction from Clinical Cases for a Knowledge Graph". In: *Proc. ADBIS*. 2022.

[342] Chen Gao, Xuan Zhang, LinYu Li, JinHong Li, Rui Zhu, KunPeng Du, and QiuYing Ma. "ERGM: A multi-stage joint entity and relation extraction with global entity match". In: *Knowledge-Based Systems* 271 (2023).

[343] Jiayue Tian and Masaomi Kimura. "Multi-task Learning for Joint Entity and Relation Extraction on Open-domain". In: *Proc. ACMLC*. 2024.

[344] Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. "Graphrel: Modeling text as relational graphs for joint entity and relation extraction". In: *Proc. ACL*. 2019.

[345] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. "Joint entity recognition and relation extraction as a multi-head selection problem". In: *Expert Systems with Applications* (2018).

[346] Zhiqiang Geng, Yanhui Zhang, and Yongming Han. "Joint entity and relation extraction model based on rich semantics". In: *Neurocomputing* (2021).

[347] Yu-Ming Shang, Heyan Huang, and Xianling Mao. "Onerel: Joint entity and relation extraction with one module in one step". In: *Proc. AAAI*. 2022.

[348] Shiye Li and Li Yi. "A Few-Shot Entity Relation Extraction Method in the Legal Domain Based on Large Language Models". In: *Proc. DEAI*. 2024.

[349] John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. "Structured information extraction from scientific text with large language models". In: *Nature Communications* (2024).

[350] Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. "A Comprehensive Survey on Relation Extraction: Recent Advances and New Frontiers". In: *ACM Computing Surveys* 56 (2024).

[351] Zikang Zhang, Wangjie You, Tianci Wu, Xinrui Wang, Juntao Li, and Min Zhang. "A Survey of Generative Information Extraction". In: *Proc. COLING*. 2025.

[352] Duarte Treigueiros and Robert Berry. "The application of neural network based methods to the extraction of knowledge from accounting reports". In: *Proc. HICSS*. 1991.

[353] David Biesner, Rajkumar Ramamurthy, Robin Stenzel, Max Lübbering, Lars Hillebrand, Anna Ladi, Maren Pielka, Robin Stenzel, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. "Anonymization of German financial documents using neural network-based language models with contextual word representations". In: *International J. of Data Science and Analytics* (2021).

[354] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. *Improving neural networks by preventing co-adaptation of feature detectors*. 2012. arXiv: `1207.0580 [cs.NE]`.

[355] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. "How to fine-tune BERT for text classification?" In: *Proc. CCL*. 2019.

[356] Chris M Bishop. "Training with noise is equivalent to Tikhonov regularization". In: *Neural computation* (1995).

[357] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. "Regularizing and Optimizing LSTM Language Models". In: *Proc. ICLR*. 2018.

[358] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. "NoisyTune: A Little Noise Can Help You Finetune Pretrained Language Models Better". In: *Proc. ACL*. 2022.

[359] Augustin-Louis Cauchy. "Méthode générale pour la résolution des systemes d'équations simultanées". In: *Comptes rendus de l'Académie des sciences* (1847).

[360] Lefteris Loukas, Manos Fergadiotis, Ion Androutsopoulos, and Prodromos Malakasiotis. "EDGAR-CORPUS: Billions of Tokens Make The World Go Round". In: *Proc. Workshop ECONLP*. 2021.

[361] Andrew Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE Transactions on Information Theory* 13 (1967).

[362] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proc. ICML*. 2001.

[363] Pabitra Mitra and SK Ghosh. "Conditional random field based named entity recognition in geological text". In: *International Journal of Computer Applications* 1 (2010).

[364] Donghuo Zeng, Chengjie Sun, Lei Lin, and Bingquan Liu. "LSTM-CRF for Drug-Named Entity Recognition". In: *Entropy* 19 (2017).

[365] Ying An, Xianyun Xia, Xianlai Chen, Fang-Xiang Wu, and Jianxin Wang. "Chinese clinical named entity recognition via multi-head self-attention based BiLSTM-CRF". In: *Artificial intelligence in medicine* 127 (2022).

[366] Bundesrepublik Deutschland. *Handelsgesetzbuch*. Accessed 11/04/2025. 2024. URL: `https://www.gesetze-im-internet.de/hgb`.

[367] Richard AA Jonker, Tiago Almeida, Rui Antunes, João R Almeida, and Sérgio Matos. "Multi-head CRF classifier for biomedical multi-class named entity recognition on Spanish clinical notes." In: *Database: The Journal of Biological Databases & Curation* 2024 (2024).

[368] Julius Sim and Chris C Wright. "The kappa statistic in reliability studies: use, interpretation, and sample size requirements". In: *Physical therapy* (2005).

[369] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. *Zephyr: Direct Distillation of LM Alignment*. 2023. arXiv: `2310.16944 [cs.LG]`.

[370] Dennis Taylor. *We Are Legion (We Are Bob)*. New York, NY: Worldbuilders Press, 2016.

[371] Tobias Deußer, Abdul Mohsin Siddiqi, Lorenz Sparrenberg, Tobias Adams, Christian Bauckhage, and Rafet Sifa. "Fusing speech and language models for dementia detection". In: *Proc. BigData*. 2024.

[372] Emma Nichols, Jaimie D Steinmetz, Stein Emil Vollset, Kai Fukutaki, Julian Chalek, Foad Abd-Allah, Amir Abdoli, Ahmed Abualhasan, Eman Abu-Gharbieh, Tayyaba Tayyaba Akram, et al. "Estimation of the global prevalence of dementia in 2019 and forecasted prevalence in 2050: an analysis for the Global Burden of Disease Study 2019". In: *The Lancet Public Health* (2022).

[373] Alexander Kurz, Hans-Jürgen Freter, Susanna Saxl, and Ellen Nickel. *Demenz. Das Wichtigste*. 8th. Deutsche Alzheimer Gesellschaft e. V., Berlin, 2019.

[374] Ashir Javeed, Ana Luiza Dallora, Johan Sanmartin Berglund, Arif Ali, Liaqata Ali, and Peter Anderberg. "Machine learning for dementia prediction: a systematic review and future research directions". In: *J. of medical systems* (2023).

[375] World Health Organization. *Dementia Fact Sheet*. `https://www.who.int/news-room/fact-sheets/detail/dementia`. Accessed: 29/07/2025. Mar. 2025.

[376] Dennis J Selkoe and John Hardy. "The amyloid hypothesis of Alzheimer's disease at 25 years". In: *EMBO Molecular Medicine* (2016).

[377] Elizabeth W Twamley, Susan A Legendre Ropacki, and Mark W Bondi. "Neuropsychological and neuroimaging changes in preclinical Alzheimer's disease". In: *Journal of the International Neuropsychological Society* (2006).

[378] Guy M McKhann, David S Knopman, Howard Chertkow, Bradley T Hyman, Clifford R Jack Jr, Claudia H Kawas, William E Klunk, Walter J Koroshetz, Jennifer J Manly, Richard Mayeux, et al. "The diagnosis of dementia due to Alzheimer's disease: recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease". In: *Alzheimer's & Dementia* (2011).

[379] Karl Li, Tanweer Rashid, Jinqi Li, Nicolas Honnorat, Anoop Benet Nirmala, Elyas Fadaee, Di Wang, Sokratis Charisis, Hangfan Liu, Crystal Franklin, et al. "Postmortem brain imaging in Alzheimer's disease and related dementias: The south Texas Alzheimer's disease research center repository". In: *J. of Alzheimer's disease* (2023).

[380] Robert E Hales, Stuart C Yudofsky, and Glen O Gabbard. *American Psychiatric Publishing Textbook of Psychiatry*. American Psychiatric Publishing Inc, Arlington, VA, 2008.

[381] Nikhil Pateria and Dilip Kumar. "A comprehensive review on detection and classification of dementia using neuroimaging and machine learning". In: *Multimedia Tools and Applications* (2024).

[382] Pranav Mahajan and Veeky Baths. "Acoustic and language based deep learning approaches for Alzheimer's dementia detection from spontaneous speech". In: *Frontiers in Aging Neuroscience* (2021).

[383] Suriya Murugan, Chandran Venkatesan, MG Sumithra, Xiao-Zhi Gao, B Elakkiya, Muthuramalingam Akila, and S Manoharan. "DEMNET: A deep learning model for early diagnosis of Alzheimer diseases and dementia from MR images". In: *IEEE Access* (2021).

[384] Ploypaphat Saltz, Shih Yin Lin, Sunny Chieh Cheng, and Dong Si. "Dementia detection using transformer-based deep learning and natural language processing models". In: *Proc. ICHI*. 2021.

[385] Greta Szatloczki, Ildiko Hoffmann, Veronika Vincze, Janos Kalman, and Magdolna Pakaski. "Speaking in Alzheimer's disease, is that an early sign? Importance of changes in language abilities in Alzheimer's disease". In: *Frontiers in aging neuroscience* (2015).

[386] Ali Khodabakhsh, Fatih Yesil, Ekrem Guner, and Cenk Demiroglu. "Evaluation of linguistic and prosodic features for detection of Alzheimer's disease in Turkish conversational speech". In: *EURASIP J. on Audio, Speech, and Music Processing* (2015).

[387] Laura Calzà, Gloria Gagliardi, Rema Rossini Favretti, and Fabio Tamburini. "Linguistic features and automatic classifiers for identifying mild cognitive impairment and dementia". In: *Computer Speech & Language* (2021).

[388] James T Becker, François Boiler, Oscar L Lopez, Judith Saxton, and Karen L McGonigle. "The natural history of Alzheimer's disease: description of study cohort and accuracy of diagnosis". In: *Archives of neurology* (1994). Supported by Grants NIA AG03705 and AG05133.

[389] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. "Robust speech recognition via large-scale weak supervision". In: *Proc. ICML*. 2023.

[390] Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. *Jasper and Stella: distillation of SOTA embedding models*. 2025. arXiv: 2412.19048 [cs.IR]. URL: https://arxiv.org/abs/2412.19048.

[391] Fasih Haider, Sofia de la Fuente, and Saturnino Luz. "An Assessment of Paralinguistic Acoustic Features for Detection of Alzheimer's Dementia in Spontaneous Speech". In: *J. of Selected Topics in Signal Processing* (2020).

[392] Kathleen C Fraser, Jed A Meltzer, and Frank Rudzicz. "Linguistic features identify Alzheimer's disease in narrative speech". In: *J. of Alzheimer's disease* (2015).

[393] Piew Datta, WR Shankle, and Michael Pazzani. "Applying machine learning to an Alzheimer's database". In: *Proc. AAAI symposium*. 1996.

[394] William Rodman Shankle, Subramani Mani, Michael J Pazzani, and Padhraic Smyth. "Detecting very early stages of dementia from normal aging with machine learning methods". In: *Proc. AIME*. 1997.

[395] Subramani Mani, Malcolm B Dick, Michael J Pazzani, Evelyn L Teng, Daniel Kempler, and I Maribell Taussig. "Refinement of neuro-psychological tests for dementia screening in a cross cultural population using machine learning". In: *Proc. AIMDM*. 1999.

[396] EL-Geneedy Marwa, Hossam El-Din Moustafa, Fahmi Khalifa, Hatem Khater, and Eman AbdElhalim. "An MRI-based deep learning approach for accurate detection of Alzheimer's disease". In: *Alexandria Engineering Journal* (2023).

[397] Pierluigi Carcagni, Marco Leo, Marco Del Coco, Cosimo Distante, and Andrea De Salve. "Convolution neural networks and self-attention learners for Alzheimer dementia diagnosis from brain MRI". In: *Sensors* (2023).

[398] Sven Haller, Hans Rolf Jäger, Meike W Vernooij, and Frederik Barkhof. "Neuroimaging in dementia: more than typical Alzheimer disease". In: *Radiology* (2023).

[399] Hadeer A Helaly, Mahmoud Badawy, and Amira Y Haikal. "Deep Learning Approach for Early Detection of Alzheimer's Disease". In: *Cognitive Computation* (2022).

[400] Marshal F Folstein, Susan E Folstein, and Paul R McHugh. "Mini-mental state: a practical method for grading the cognitive state of patients for the clinician". In: *J. of psychiatric research* (1975).

[401] Daniel Stamate, Min Kim, Petroula Proitsi, Sarah Westwood, Alison Baird, Alejo Nevado-Holgado, Abdul Hye, Isabelle Bos, Stephanie J.B. Vos, Rik Vandenberghe, et al. "A metabolite-based machine learning approach to diagnose Alzheimer-type dementia in blood: Results from the European Medical Information Framework for Alzheimer disease biomarker discovery cohort". In: *Alzheimer's & Dementia: Translational Research & Clinical Interventions* (2019).

[402] Makrina Karaglani, Krystallia Gourlia, Ioannis Tsamardinos, and Ekaterini Chatzaki. "Accurate Blood-Based Diagnostic Biosignatures for Alzheimer's Disease via Automated Machine Learning". In: *J.of Clinical Medicine* (2020).

[403] David Facal, Sonia Valladares-Rodriguez, Cristina Lojo-Seoane, Arturo X. Pereiro, Luis Anido-Rifon, and Onésimo Juncos-Rabadán. "Machine learning approaches to studying the role of cognitive reserve in conversion from mild cognitive impairment to dementia". In: *International J. of Geriatric Psychiatry* (2019).

[404] C. Thomas, V. Keselj, N. Cercone, K. Rockwood, and E. Asp. "Automatic detection and rating of dementia of Alzheimer type through lexical analysis of spontaneous speech". In: *Proc. ICMA*. 2005.

[405] Bart Peintner, William Jarrold, Dimitra Vergyri, Colleen Richey, Maria Luisa Gorno Tempini, and Jennifer Ogar. "Learning diagnostic models using speech and language measures". In: *Proc. EMBC*. 2008.

[406] Brian Roark, Margaret Mitchell, John-Paul Hosom, Kristy Hollingshead, and Jeffrey Kaye. "Spoken language derived measures for detecting mild cognitive impairment". In: *IEEE transactions on audio, speech, and language processing* (2011).

[407] M. Rupesh Kumar, Susmitha Vekkot, S. Lalitha, Deepa Gupta, Varasiddhi Jayasuryaa Govindraj, Kamran Shaukat, Yousef Ajami Alotaibi, and Mohammed Zakariah. "Dementia Detection from Speech Using Machine Learning and Deep Learning Architectures". In: *Sensors* 22.23 (2022).

[408] Bahman Mirheidari, Daniel Blackburn, Traci Walker, Markus Reuber, and Heidi Christensen. "Dementia detection using automatic analysis of conversations". In: *Computer Speech & Language* (2019).

[409] Ayimnisagul Ablimit, Catarina Botelho, Alberto Abad, Tanja Schultz, and Isabel Trancoso. "Exploring dementia detection from speech: Cross corpus analysis". In: *Proc. ICASSP*. 2022.

[410] Weidong Chen, Xiaofen Xing, Xiangmin Xu, Jianxin Pang, and Lan Du. "Speechformer++: A hierarchical efficient framework for paralinguistic speech processing". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023).

[411] Samad Amini, Boran Hao, Lifu Zhang, Mengting Song, Aman Gupta, Cody Karjadi, Vijaya B Kolachalama, Rhoda Au, and Ioannis Ch Paschalidis. "Automated detection of mild cognitive impairment and dementia from voice recordings: a natural language processing approach". In: *Alzheimer's & Dementia* (2023).

[412] Rui He, Kayla Chapin, Jalal Al-Tamimi, Núria Bel, Marta Marquié, Maitee Rosende-Roca, Vanesa Pytel, Juan Pablo Tartari, Montse Alegret, Angela Sanabria, et al. "Automated classification of cognitive decline and probable alzheimer's dementia across multiple speech and language domains". In: *American J. of Speech-Language Pathology* (2023).

[413] Laura C Maclagan, Mohamed Abdalla, Daniel A Harris, Therese A Stukel, Branson Chen, Elisa Candido, Richard H Swartz, Andrea Iaboni, R Liisa Jaakkimainen, and Susan E Bronskill. "Can Patients with Dementia Be Identified in Primary Care Electronic Medical Records Using Natural Language Processing?" In: *J. of Healthcare Informatics Research* (2023).

[414] Saturnino Luz, Fasih Haider, Sofia de la Fuente Garcia, Davida Fromm, and Brian MacWhinney. "Editorial: Alzheimer's Dementia Recognition through Spontaneous Speech". In: *Frontiers in Computer Science* (2021).

[415] Ngumimi Karen Iyortsuun, Soo-Hyung Kim, Min Jhon, Hyung-Jeong Yang, and Sudarshan Pant. "A review of machine learning and deep learning approaches on mental health diagnosis". In: *Healthcare*. 2023.

[416] Paul Mermelstein. "Distance measures for speech recognition, psychological and instrumental". In: *Pattern recognition and artificial intelligence* (1976).

[417] Dennis Gabor. "Theory of communication". In: *J. of the Institution of Electrical Engineers* (1946).

[418] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. "A scale for the measurement of the psychological magnitude pitch". In: *J. of the Acoustical Society of America* (1937).

[419] Douglas O'Shaughnessy. *Speech Communication: Human and Machine*. Addison-Wesley Publishing Company, 1987.

[420] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proc. KDD*. 2019.

[421] Hongyoon Choi, Kyong Hwan Jin, Alzheimer's Disease Neuroimaging Initiative, et al. "Predicting cognitive decline with deep learning of brain metabolism and amyloid imaging". In: *Behavioural brain research* (2018).

[422] Abdul Rehman, Myung-Kyu Yi, Abdul Majeed, and Seong Oun Hwang. "Early Diagnosis of Alzheimer's Disease using 18F-FDG PET with Soften Latent Representation". In: *IEEE Access* (2024).

[423] E Doering, T Deußer, M Hoenig, T van Eimeren, L Ellingsen, and A Drzezga. "How Will You Age? A glimpse into future brain aging on FDG-PET using deep learning". In: *Nuklearmedizin* (2023).

[424] Peter Grosche, Meinard Müller, and Frank Kurth. "Cyclic tempogram—a mid-level tempo representation for musicsignals". In: *Proc. ICASSP*. 2010.

[425] Alyssa M Lanzi, Anna K Saylor, Davida Fromm, Houjun Liu, Brian MacWhinney, and Matthew L Cohen. "DementiaBank: Theoretical rationale, protocol, and illustrative analyses". In: *American J. of Speech-Language Pathology* (2023).

[426] Olga Ivanova, Juan José G Meilán, Francisco Martínez-Sánchez, Israel Martínez-Nicolás, Thide E Llorente, and Nuria Carcavilla González. "Discriminating speech traits of Alzheimer's disease assessed through a corpus of reading task for Spanish language". In: *Computer Speech & Language* (2022).

[427] Guanyu Zhang, Jinghong Ma, Piu Chan, and Zheng Ye. "Graph theoretical analysis of semantic fluency in patients with Parkinson's disease". In: *Behavioural Neurology* (2022).

[428] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models". In: *Computational Linguistics* (2025).

[429] Lars Hillebrand, Armin Berger, Daniel Uedelhoven, David Berghaus, Ulrich Warning, Tim Dilmaghani, Bernd Kliem, Thomas Schmid, Rüdiger Loitz, and Rafet Sifa. "Advancing Risk and Quality Assurance: A RAG Chatbot for Improved Regulatory Compliance". In: *Proc. BigData*. 2024.

[430] Yucheng Hu and Yuxing Lu. *RAG and RAU: A Survey on Retrieval-Augmented Language Model in Natural Language Processing*. 2025. arXiv: 2404.19543 [cs.CL].