Domain-Agnostic Approaches to Competency Extraction via Large Language Models

Master Thesis

by

Bertil Braun

Degree Course: Informatik M.Sc.
Matriculation Number: 2360487

Institute of Applied Informatics and Formal Description Methods (AIFB)

KIT Department of Economics and Management

Advisor:	Prof. Dr. Andreas Oberweis
Second Advisor:	Prof. Dr. Ralf Reussner
Supervisor:	M.Sc. Martin Forell
Submitted:	October 2, 2024

Abstract

In contemporary academic and professional environments, there is an escalating need for precise competency profiling to enhance collaboration and strategic alignment. This thesis presents a domain-agnostic system utilizing Large Language Models (LLMs) to improve the extraction of competencies from various types of documents, addressing the limitations of current systems in handling unstructured data across diverse environments. The system integrates a comprehensive multi-phase approach that includes selecting and fine-tuning LLMs, exploring different extraction methods for competencies from documents—namely abstracts, full texts, and summaries—and the execution of extensive evaluations through both automatic and expert assessments.

Experimental results demonstrate that competency extraction from abstracts is most effective, aligning closely with expert evaluations and offering a quick and efficient profiling method. The adaptability of the system across various domains without loss of performance highlights its potential for widespread application in both academic and corporate contexts. This thesis emphasizes the significant role of advanced LLMs in developing dynamic, scalable competency extraction systems, which are crucial for fostering enhanced interdisciplinary collaboration and optimizing professional development.

Contents

List of Abbreviations

A100 NVIDIA A100 80GB Tensor Core GPU.

AIFB Institute of Applied Informatics and Formal Description Methods.

API Application Programming Interface.

BERT Bidirectional Encoder Representations from Transformers.

BW-Uni-Cluster Baden-Württemberg University Cluster 2.0.

CAS CAS Software AG.

DPO Direct Preference Optimization.

Elo Elo Rating System.

FcNN fully connected neural network.

Gemma2 A large language model family developed by Google.

GPT Generative Pre-trained Transformer, a large language model family developed by OpenAI.

GPU Graphics Processing Unit.

HR Human Resources.

KIT Karlsruhe Institute of Technology.

KL Kullback-Leibler Divergence.

KMS Knowledge Management System.

KV Key-Value.

Llama3 a large language model family developed by Meta.

LLM Large Language Model.

LMQL Language Model Query Language.

LoRA Low-Rank Adaptation.

LSTM Long Short-Term Memory.

Mixtral A large language model family developed by french mistral.ai.

NLP Natural Language Processing.

NN Neural Network.

OCR Optical Character Recognition.

PEFT Parameter Efficient Fine-Tuning.

Phi3 A large language model family developed by Microsoft.

PPO Proximal Policy Optimization.

QLoRA Quantized Low-Rank Adaptation.

RAG Retrieval Augmented Generation.

RL Reinforcement Learning.

RLHF Reinforcement Learning from Human Feedback.

RNN Recurrent Neural Network.

RoPE Rotary Position Embeddings.

SFT Supervised Fine-Tuning.

sota State-of-the-Art.

TRL Transformers Reinforcement Learning.

List of Figures

List of Tables

1 Introduction

This section introduces the importance of competence extraction, its challenges, and the potential of Large Language Models (LLMs) to address these issues. It explores the current systems and outlines objectives and methodologies for this thesis.

1.1 Motivation

The importance of competence extraction, defined as the identification and assessment of practical applications of knowledge and skills, has escalated as a focal area of inquiry. This is driven by the growing necessity for precise skill profiling within academic and business sectors [AS15; Sat+17; Mas+23]. With the complexity of industries and research areas expanding, organizations face challenges in identifying and utilizing specialized competencies. LLMs have proven to be effective in analyzing extensive volumes of unstructured text to extract information [Dun+22; Aro+23]. Recent developments in LLMs, showcasing their capability to distill complex data from elaborate texts, demonstrate their utility in refining knowledge extraction and structuring detailed competencies across various fields [Ngu+24]. This capability is especially pertinent in environments where precise delineation and visibility of skills are crucial for productive collaboration [Zha+22; Bö+18].

Problem Statement: The current systems for skill extraction, although operational in some structured settings such as job descriptions and job applications, do not consistently achieve the precision and adaptability needed in academic contexts [Sat+17]. Academic personnel at institutions like the Karlsruhe Institute of Technology (KIT) contribute a diverse range of competencies via their publications. Nonetheless, the variability in document formats and the intricate nature of academic outputs impede these systems' ability to compile skills into coherent profiles effectively. This issue originates from the limitations of conventional information extraction methods in handling complex, unstructured data [Sat+17]. Research indicates that LLMs, including GPT-3 and GPT-4, are adept at concept extraction [Dun+22; Aro+23]. Enhancing these models for domain-specific knowledge extraction could significantly elevate their precision and practicality [Ngu+24; Kum+22].

Relevance to Practice: An advanced competence extraction system is vital for fostering improved collaboration and knowledge dissemination in both academic and corporate settings [Sat+17; Bö+18]. In academic circles, an effective skill extraction system could facilitate the establishment of an extensive competence network, as worked on by the

Institute of Applied Informatics and Formal Description Methods (AIFB)[1], enabling researchers to engage with colleagues possessing complementary skills [Bö+18; Bay+18; Mas+23]. Such a system can significantly enhance interdisciplinary collaboration and foster more groundbreaking research. Likewise, in the corporate realm, a deeper insight into employee competencies enables better alignment of workforce skills with organizational strategic objectives [Vuk+21; AS15; Bay+18]. Systems adept at extracting and structuring competencies from varied document types, including scientific papers, job advertisements, and internal documents, are indispensable for creating strong expertise networks [Bay+18; Mas+23; GEA19a].

1.2 Aim of the Work

The principal objective of this thesis is to develop a system that consistently performs robust and accurate competency extraction across various document types for efficient and accurate competency extraction using LLMs. The system is designed with a domain-agnostic methodology to ensure scalability and adaptability across different domains and document formats.

Specific Objectives

- **Development of an Efficient Competency Extraction System:** This task involves constructing a system capable of processing extensive documents and extracting competencies with high precision. Utilizing advanced LLMs, the system addresses the complexity and diversity inherent in such documents. Various methods of extraction will be explored and assessed to establish the most effective technique for competency retrieval from different document types.
- **Expert Validation of Extracted Competencies:** The system's outputs will undergo an expert-based evaluation to validate the alignment of extracted competencies with established domain knowledge. This evaluation allows domain experts to review, compare different system outputs, and provide feedback on the relevance and precision of competencies identified by the system.
- **Automated Evaluation of Competency Extraction:** Alongside expert assessments, an automated evaluation framework will be implemented to gauge the accuracy and pertinence of competencies extracted, using comparisons facilitated by LLMs. This evaluation will quantify the system's performance and assist in pinpointing improvement needs during development. Correlations between expert as-

[1](visited on 10/01/2024) bis.aifb.kit.edu/317_389.php

sessments and automated results will be analyzed to affirm the system's reliability and efficacy.

- **Development and Fine-Tuning of LLMs:** A crucial component of this thesis is the fine-tuning of a pre-trained LLM specifically for competency extraction. This process involves generating a synthetic dataset for fine-tuning, selecting an optimal base model, and enhancing model performance through Direct Preference Optimization (DPO). The objective is to cultivate a model that surpasses existing base models in accuracy and efficiency for competency extraction.
- **Investigation of Domain Independence:** The ultimate objective is to assess the domain-independent capabilities of the developed system. This assessment will test the system's efficacy across diverse datasets, including academic publications and corporate documents, to confirm its ability to extract relevant competencies regardless of the domain. The aim is to establish a versatile, efficient, and effective general-purpose competency extraction tool.

1.3 Structure of the Thesis

Chapter 2: Theoretical Background This chapter delineates the foundational concepts pivotal to competency extraction and Natural Language Processing (NLP), emphasizing LLMs. It elucidates the core components of LLMs such as the Transformer architecture, embeddings, and the capacity of these models to manage extensive contexts. Additionally, this section delves into methodologies like fine-tuning, Reinforcement Learning from Human Feedback (RLHF), DPO, and Retrieval Augmented Generation (RAG) comprehensively.

Chapter 3: Current State of Research This chapter provides an overview of recent advancements in the field of competency extraction and associated technologies. It concentrates on techniques for deriving competencies from job postings and scholarly documents.

Chapter 4: Methodology This chapter delineates the approach for developing and assessing methods of competency extraction. It details techniques for deriving competencies from extensive scientific texts, ranging from abstracts to entire documents. The chapter introduces methods for LLMs fine-tuning with a focus on DPO and domain adaptation strategies such as RAG. It also covers data sources, preprocessing steps, and evaluation metrics utilized in determining the performance of the system.

Chapter 5: Development This chapter addresses the technical construction of the competency extraction system. It describes the architecture and components of the system, from backend infrastructure to LLMs integration. The processes for developing extraction methods—deriving competencies from abstracts, complete texts, and summaries—are elucidated. It further explores the challenges posed by diverse document types and formats and the iterative process of optimizing LLMs for superior performance.

Chapter 6: Evaluation The chapter on evaluation details findings from expert assessments, contrasting various competency extraction methods and their effectiveness. Quantitative results such as enhancements in preference scores, along with qualitative feedback from domain specialists, are discussed to evaluate the accuracy and dependability of the competency profiles. The benefits of DPO in refining the extraction process are examined. Additionally, the system's efficiency and scalability are assessed.

Chapter 7: Summary and Outlook The concluding chapter integrates research outcomes and contemplates advancements in developing an effective, domain-agnostic competency extraction system. It highlights future research possibilities, emphasizing the improvement of LLMs fine-tuning techniques and the integration of novel models into the extraction pipeline. Opportunities for augmenting system adaptability across diverse sectors, including corporate and academic environments, are discussed.

2 Theoretical Background

This chapter delineates the foundational concepts pivotal to competency extraction and NLP, emphasizing LLMs. It elucidates the core components of LLMs such as the Transformer architecture, embeddings, and the capacity of these models to manage extensive contexts. Additionally, this section delves into methodologies like fine-tuning, RLHF, DPO, and RAG comprehensively.

2.1 Introduction to Competence Extraction

This subsection explores the multifaceted nature of competence profiles within various contexts, emphasizing their pivotal role in organizational, scientific, and technological environments. It delves into how competencies are conceptualized and operationalized to reflect both technical abilities and knowledge application, highlighting their dynamic and evolving nature. The discussion extends to the systematic depiction of these competencies through competence profiles, which are crucial for strategic development, effective team formation, and optimized resource allocation across diverse sectors.

2.1.1 Definition of Competence Profiles

Competencies encompass a blend of abilities, knowledge, and skills that individuals exhibit in particular domains and apply effectively in real-world scenarios [Sch19]. These competencies are fundamental for assessing and comprehending the capabilities within organizations, teams, or scientific communities [RK92]. Various interpretations of competencies appear in scholarly works, with some frameworks emphasizing technical skills while others highlight personal and social capacities [Wes01; Mé05]. In this thesis, competencies are defined as encompassing both technical prowess and the capacity to generate and utilize knowledge in specific situations [Hon+21]. This conceptualization views competencies not merely as static qualities but as dynamic entities that develop through ongoing learning and experiential growth [Cam+10].

A competence profile provides a systematic depiction of these capabilities, enabling the transparent representation of individual or group competencies. Competence profiles play a pivotal role in team formation, the planning of professional development initiatives, and the pinpointing of specialized knowledge areas [GEA19b; Mar+22]. An illustrative competence profile is presented below:

Listing 1: Example Competence Profile for an Expert Specializing in Web Community Development through Competence Analysis

```
Domain: Expert in developing web communities through detailed competence
    analysis.

Competencies:
- Competence Identification: Expertly utilizes scientific publications to
    accurately delineate and identify individual competences, enhancing the
    precision of skill profiling.
- Community Building: Actively fosters the development and growth of
    scientific communities by harmonizing diverse expertises, thus
    facilitating collaborative innovation.
- Decision Support Systems: Enhances decision-making processes by
    integrating structured competences into advanced support systems,
    improving organizational outcomes.
- Team Formation: Promotes the efficient assembly of teams through precise
    competence identification, optimizing team synergy and performance.
- Knowledge Visualization: Deploys advanced evolutionary visual tools to
    effectively illustrate the progression and dynamics of virtual
    scientific communities, making complex data accessible.
- Expertise Analysis: Conducts thorough evaluations of published knowledge
    to recommend optimal roles and collaborations, enhancing the strategic
    utilization of expertise.
```

2.1.2 Differences Between Various Competence Definitions

The definition of competencies can significantly vary based on the application domain. In the context of corporate practice, the focus is often on technical and professional skills, while academic research prioritizes the capacity to generate new knowledge and innovate [Mé05]. A crucial distinction is also made between implicit and explicit knowledge [TWT23]. Implicit knowledge encompasses abilities that are challenging to formalize and express, in contrast to explicit knowledge, which can be clearly articulated and documented. This differentiation is essential for the process of automated competence extraction, as capturing implicit knowledge in textual data proves more difficult [TWT23].

2.1.3 Relevance of the Specific Definition for This Work

This thesis is dedicated to the automated extraction of competence profiles from scientific texts and other pertinent documents. It utilizes a definition of competencies that aligns with scientific and organizational standards, emphasizing both explicit and implicit competencies [Sat+17]. This thesis primarily targets competencies demonstrable through publications and other recognized knowledge sources. The objective is to extract these

competencies with precision and efficiency using NLP techniques and to depict them in a structured profile. Such profiles facilitate enhanced decision-making and more effective team formation in organizational settings [Sat+17; Hon+21].

2.1.4 Significance of Competence Profiles

Competence profiles play an integral role across organizations, scientific communities, and businesses by systematically showcasing the skills and expertise of individuals or teams [RK92]. These profiles provide a structured overview of available competencies, facilitating strategic personnel development, the promotion of expertise, and the optimized assembly of teams [KHB09]. By associating abilities clearly with individuals, competence profiles serve a significant purpose across various sectors.

Within organizations, competence profiles are instrumental in identifying strengths and areas for development among employees [Ver+08]. They enable the strategic assignment of projects tailored to individual capabilities and enhance internal communication and collaboration [Sat+17]. In the domain of personnel development, competence profiles are utilized to customize training and development programs, systematically supporting employee growth. Over time, this contributes to the organization's capacity for innovation and competitiveness [Sat+17].

In the scientific domain, competence profiles facilitate more effective networking among researchers [Mar+22]. They make expertise readily visible and aid in identifying appropriate collaborators, especially in interdisciplinary research fields [Mar+22]. This supports knowledge sharing and expedites scientific advancement. Moreover, competence profiles are employed to efficiently form research teams by uniting researchers with complementary skills [Mar+22].

The value of competence profiles is also acknowledged in the business sector. In an era emphasizing data-driven decision-making, competence profiles provide a robust foundation for strategic personnel decisions and project planning [SCF22; Mü+16]. For instance, businesses utilize competence profiles for more effective hiring processes, ensuring an ideal alignment between employees and roles [Mü+16]. They further support talent management by pinpointing and developing potential for upcoming challenges [SCF22].

2.1.5 Challenges and Benefits of Automated Competence Extraction

The automated extraction of competence profiles encounters several challenges. A primary issue is the lack of clear definitions or standardization of competencies [Wes01; Mé05]. Variations in the definitions and requirements for the same competencies across different industries or companies complicate the comparison and generalization of these

profiles. This problem is exacerbated when competencies are extracted from unstructured data sources, such as scientific publications or job descriptions, leading to potential misinterpretations or incorrect assignments that diminish the quality of the profiles [Sat+17; Mé05].

Additionally, the volume and complexity of the data utilized for competence extraction pose a challenge. Although manual validation is manageable with small datasets, the processing of large datasets necessitates robust and scalable algorithms that maintain high accuracy. Automated systems are required to efficiently filter relevant information and exclude irrelevant or redundant data from the competence profiles [Sat+17; SCF22].

However, the implementation of automated competence extraction presents considerable benefits. It significantly reduces the time needed to create competence profiles, especially when dealing with large volumes of documents or textual data [SCF22; Sat+17]. The efficiency gains are particularly notable in knowledge-intensive fields such as research or technology-driven sectors [Hon+21]. Furthermore, automation supports the continuous update of competence profiles, which is crucial in dynamic sectors where requirements and technologies frequently change [Hon+21].

Automation also aids in the standardization of competence profiles by employing uniform criteria and metrics for the evaluation and assignment of competencies [SCF22]. This enhancement in standardization aids in the comparison of profiles across various domains and supports decision-making in project or personnel planning. Additionally, automated competence extraction facilitates talent management and strategic personnel development by providing data-driven insights into current skills and potential areas for improvement [SCF22; Hon+21; Mar+22].

2.2 Foundations of NLP Research

This subsection explores the evolution and core principles that define contemporary Natural Language Processing (NLP). It begins with a historical overview, tracing the shift from rule-based systems to advanced machine learning frameworks that leverage large datasets to discern linguistic patterns. Subsequent sections delve into the foundational concepts crucial to NLP such as tokenization, embeddings, and the handling of variable sequence lengths. Special attention is given to the transformative impact of Transformer models and the pivotal role of embeddings in enhancing semantic analysis and supporting advanced applications in text processing.

2.2.1 Development of NLP

The field of NLP has witnessed substantial growth over recent decades, transitioning from rule-based to contemporary machine learning and deep learning methodologies [OMK21]. Initially, NLP was characterized by rule-based systems which utilized predefined rules and patterns to analyze linguistic structures [OMK21]. These systems, however, were inadequate for managing the complexity and variability of natural language, prompting a shift towards statistical models and data-oriented strategies. The adoption of machine learning, particularly on expansive datasets, was crucial in identifying and interpreting linguistic patterns [OMK21].

With the progression of machine learning, especially through the use of Neural Networks (NNs), the capability to apprehend more intricate linguistic structures improved significantly. Notable developments were seen with the introduction of Recurrent Neural Networks (RNNs) and Long Short-Term Memorys (LSTMs), which effectively modeled sequential dependencies in text [Elm20; HS97]. Subsequently, the emergence of Transformer models, particularly highlighted in [Vas+23], brought transformative changes to NLP by overcoming challenges associated with processing extensive dependencies in texts [Elm20; HS97; Vas+23]. These models have greatly enhanced the precision of language processing and have established a robust foundation for LLMs, which currently lead NLP research.

2.2.2 Fundamental Concepts in NLP

NLP is underpinned by several essential concepts:

- **Tokenization**: The process of breaking down text into smaller segments known as tokens, which include words, subwords, or characters, constitutes the initial phase in most NLP frameworks. This facilitates the structured text processing necessary for model operations [Mik+13].
- **Word Vectors (Embeddings)**: These vectors offer a continuous, high-dimensional representation of words, enabling models to detect semantic similarities. The models such as "Word2Vec" ([Mik+13]) and "Bidirectional Encoder Representations from Transformers (BERT)" ([Dev+19]) ascertain these relationships through context analysis.
- **Variable Sequence Lengths**: Conventional NN architectures necessitate uniform input dimensions and are thus inadequate for texts of variable lengths. Although padding and truncation methods help standardize these lengths, they may introduce inefficiencies [Elm20]. RNNs and LSTMs incorporate memory cells that preserve

information across time, facilitating the handling of sequential data irrespective of length variations [Elm20; HS97]. Nevertheless, these models encounter issues like catastrophic forgetting and vanishing gradients when managing extensive text sequences [HS97].

- **Transformer Models**: The advent of Transformer architectures (refer to section 2.3) mitigates issues related to sequence length and dependency processing. Through self-attention mechanisms, Transformers efficiently compute the relevance of words across various contexts [Vas+23].

2.2.3 Embeddings and Their Role in NLP

Embeddings serve as the cornerstone for contemporary NLP systems by providing dense, continuous vector space representations of words and sentences. In this space, words with similar meanings are positioned closely, enhancing semantic analysis [Mik+13]. This approach differs fundamentally from previous models such as the bag-of-words, as embeddings encapsulate inter-word dependencies. Further advancements include contextual embeddings, exemplified by "BERT" [Dev+19], which evaluate the full context surrounding a word, thereby effectively addressing ambiguities in word usage.

Additionally, the development of sentence and document embeddings has broadened these methodologies, enabling systems to semantically interpret entire texts [RG19]. Such advancements significantly benefit tasks like text classification, question answering, and machine translation. Consequently, embeddings are integral to the architecture of numerous advanced NLP models, especially LLMs, and they play a pivotal role in various downstream applications, as outlined in section 2.4.

2.3 Introduction to LLMs

This subsection explores the fundamental components and operational principles of LLMs, emphasizing the transformative impact of the Transformer architecture. It details the mechanisms of self-attention and positional embeddings that enable these models to process text in parallel, enhancing efficiency and scalability. Differences from previous architectures, such as RNNs and LSTMs, are examined, alongside a discussion on the implementation of models like GPT-4. The section also addresses enhancements in model training and deployment, focusing on the integration of advanced techniques to handle long sequences and complex reasoning tasks efficiently.

2.3.1 Fundamentals of Transformer Architecture

The Transformer architecture, as outlined in "Attention is All You Need" [Vas+23], underpins modern LLMs. Distinct from RNNs and LSTMs, the Transformer employs a self-attention mechanism to establish relationships among words within a text, facilitating the processing of global dependencies efficiently. The mechanism of self-attention assigns weights to each word relative to all other words based on significance, enabling the capture of contextual information throughout the sequence.

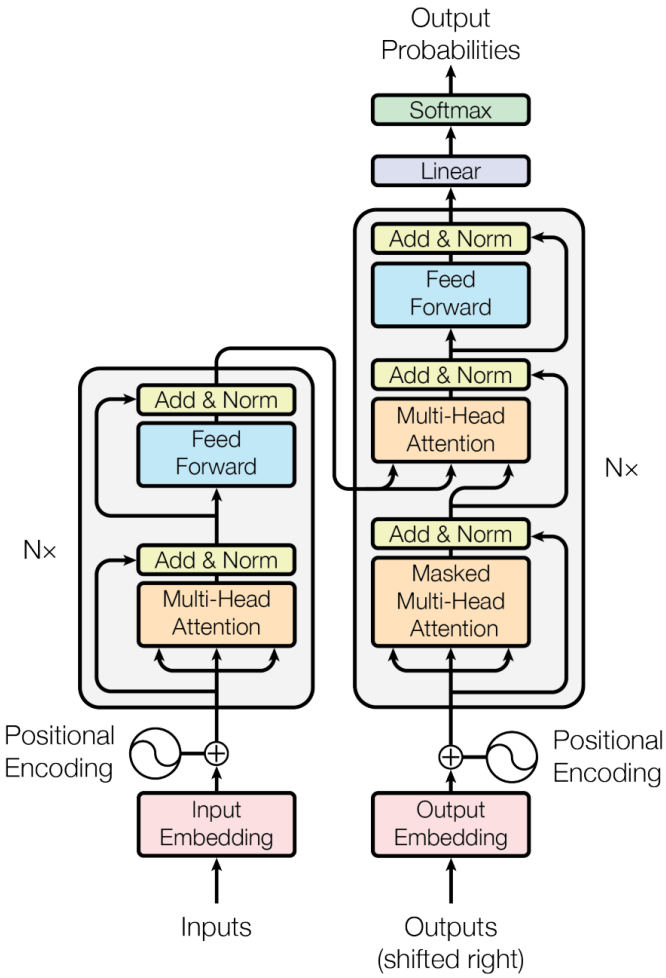

Figure 1: The Transformer - model architecture.

Figure 1: The Transformer architecture [Vas+23].

This architecture (see fig. 1) comprises two principal components: the encoder and the decoder. Both elements contain multiple layers of self-attention mechanisms and fully connected neural network (FcNN) layers. The self-attention mechanisms facilitate the model's ability to focus on different parts of the input sequence for contextual understanding, while the FcNN layers allow for the nonlinear transformation of data, crucial for integrating and refining the information across the network. In machine translation

contexts, the encoder converts an input sequence into a latent representation, which the decoder subsequently utilizes to produce an output. Numerous modern LLMs, including GPT-3([Bro+20]), a large language model family developed by Meta (Llama3)[2] and A large language model family developed by Microsoft (Phi3)[3], implement an autoregressive decoder-only architecture, where the generation of each token depends on the tokens generated previously (refer to fig. 2, section 2.3.4).

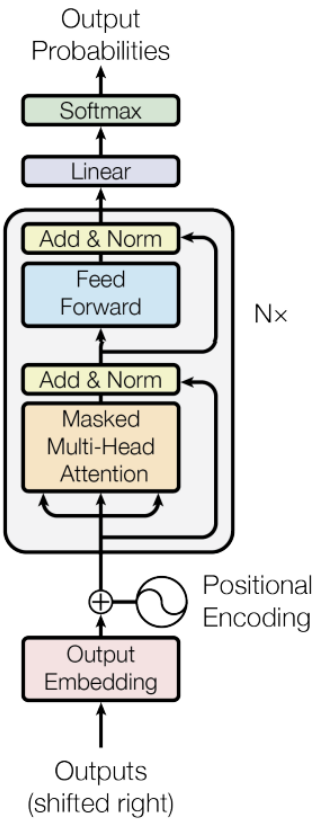

Figure 2: The autoregressive decoder-only architecture of the Transformer [Vas+23].

Differences from Previous Architectures The Transformer provides substantial improvements over RNNs and LSTMs regarding computational efficiency and model capability. Contrary to RNNs and LSTMs, which process data sequentially and are prone to the vanishing gradient problem, the Transformer processes input sequences in parallel, resulting in accelerated training and enhanced handling of long sequences. The self-attention mechanism enables flexible modulation of dependencies in texts, which enhances the system's proficiency in identifying and modeling intricate relationships [Hah20].

[2](visited on 10/01/2024) ai.meta.com/blog/meta-llama-3/

[3](visited on 10/01/2024) azure.microsoft.com/blog/introducing-phi-3-redefining-whats-possible-with-slms/

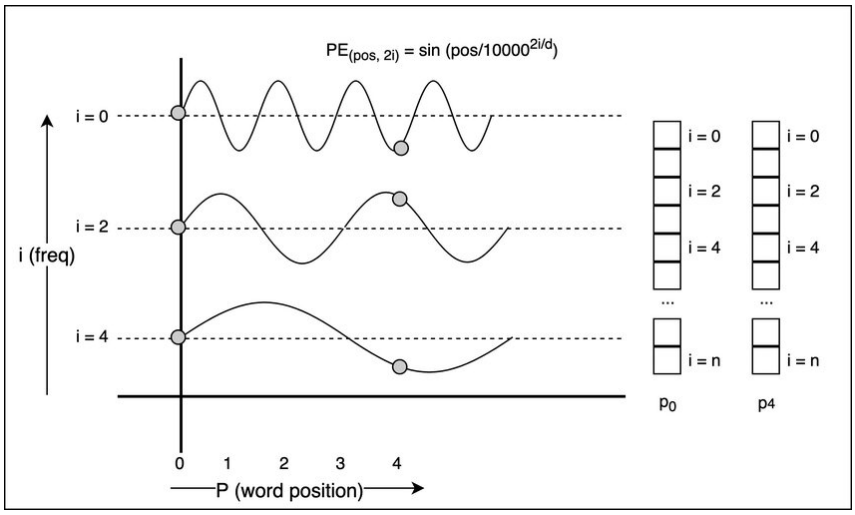

Figure 3: Sinusoidal Positional Encodings in Transformers from [HGS21]

Moreover, the Transformer employs positional embeddings (refer to section 2.3.2) to encode token order, unlike RNNs and LSTMs that model sequence order through their recurrent structures. These embeddings provide details about the relative positions of words within a sentence, allowing the system to acknowledge sequence order despite its non-sequential data processing approach [Vas+23].

The integration of self-attention and parallel processing in the Transformer marks a significant evolution, facilitating the development of models with hundreds of billions of parameters and the capability to manage extensive text volumes [Ope+23].

2.3.2 Uniform Implementation Aspects of LLMs

Positional Embeddings and Their Variations The predominant method for encoding positions utilizes sinusoidal Positional Encodings (see fig. 3) as introduced by [Vas+23]. This technique encodes each position in the sequence with sine and cosine functions across varying frequencies (i) where the position is used to sample the sin functions, enabling the model to discern both absolute and relative positions. At lower frequencies, the system captures global information, while at higher frequencies, it discerns local details. This method allows the model to differentiate words according to their position in the sequence and theoretically supports sequences of unlimited length.

Nevertheless, sinusoidal Positional Encodings exhibit challenges in delineating detailed positional information in extended sequences [Su+24]. To overcome these limitations, positional embedding methods such as Rotary Position Embeddings (RoPE) have been developed [Su+24]. Testing with models like RoBERTa and Performer using RoPE has demonstrated swifter training periods and enhanced robustness in handling positional information in extended sequences compared to sinusoidal embeddings [Su+24].

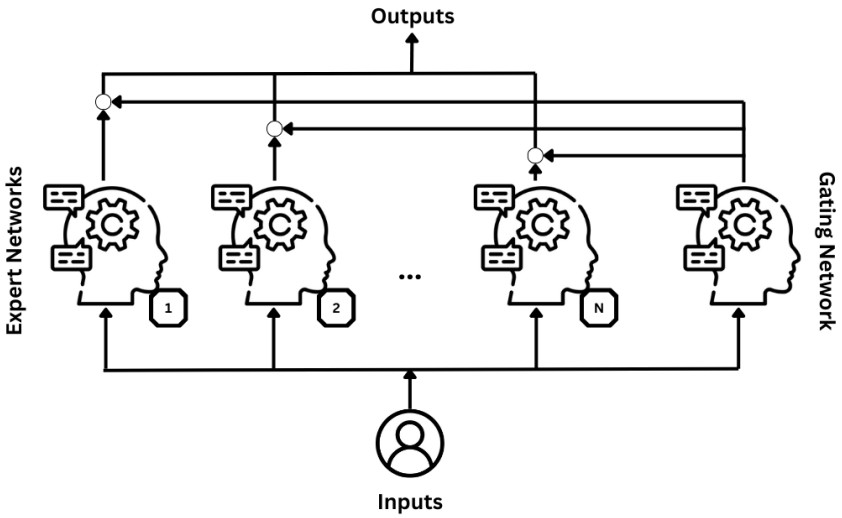

Figure 4: Mixture of Experts gating Network depiction

Mixture of Experts The Mixture of Experts (MoE) approach, outlined in [FZS22] and depicted in fig. 4, diverges from standard LLMs by activating only a subset of parameters or "experts" during each inference step. The MoE model consists of multiple specialized sub-networks tailored to distinct data types or tasks (expert networks $1 - N$ in fig. 4).

During inference, a "gate" mechanism selects the appropriate experts to activate based on the input features, indicated by the white activation gates, controlled by the gating network in fig. 4. This selection reduces computational demands and memory requirements by engaging only a portion of the available experts. The specialized nature of these experts allows the system to more effectively manage specific tasks.

However, challenges persist, including the development of an effective gating mechanism and the equitable distribution of tasks among experts. Techniques like load-balancing strategies are employed to ensure balanced distribution and sustained performance.

2.3.3 Training Methods for LLMs

Unsupervised Learning Unsupervised Learning eliminates the need for labeled data. LLMs detect patterns and structures in text by predicting parts of the text from its surrounding context [Dev+19]. Self-Supervised Learning techniques, such as predicting the next token or filling in masks, facilitate this process [Dev+19]. This method effectively processes large volumes of unlabeled text data, and is employed by leading models like GPT and BERT [Bro+20; Dev+19]. Nonetheless, its performance on domain-specific tasks may be restricted without targeted modifications [Dev+19].

Supervised Learning Supervised Fine-Tuning (SFT) consists of training a model using a dataset that includes inputs and their corresponding target outputs (labels). The model

is trained to predict the appropriate output for a given input, such as predicting the next word in a sentence [Dev+19]. This method proves beneficial when labeled data is accessible, as it allows for accurate predictions and customization to particular tasks [Dev+19]. However, it depends on the availability of extensive labeled datasets, which may be a constraint for tasks lacking such resources [Dev+19].

Reinforcement Learning Reinforcement Learning (RL) enhances LLMs by employing feedback and rewards for accurate predictions. A specific application in LLMs is RLHF, involving human evaluation of outputs to train a Reward Model, which then assists in further model optimization [Ouy+22]. RLHF is instrumental in aligning models with user-specific requirements, although it necessitates complex training mechanisms and high-quality feedback [Ouy+22].

Relevance of Pre-Training and Fine-Tuning Pre-Training involves training a model on extensive amounts of unlabeled text data to acquire a fundamental understanding of language patterns and structures [Min+24a]. This phase is not specific to any domain and provides a broad linguistic foundation, enabling versatile application across various tasks and domains [Min+24a]. Fine-Tuning customizes the model to specific tasks or domains using a more focused dataset. It can be conducted through SFT or RL, thereby boosting the model's effectiveness for specified needs [Min+24a]. Proper alignment during fine-tuning ensures the model meets user expectations and operates ethically, resulting in greater acceptance and effectiveness in practical applications [Min+24a].

In Context Learning In Context Learning is a method that allows LLMs to integrate context-specific information into their responses. By equipping the model with additional context, such as relevant documents or knowledge bases, before generating a response, the model can produce more precise and informative answers [Osw+23; Kir+24]. This approach proves particularly beneficial in question-answering tasks or dialogue systems where external information is necessary for accurate responses [Osw+23]. In Context Learning enables the model to utilize the external knowledge base for more informed responses, even post-training [Osw+23; Kir+24]. Furthermore, examples that demonstrate how to tackle similar tasks or problems significantly enhance the model's capability to generate superior responses [Dia+24; Ye+23; Kir+24].

Reasoning Before Answering Reasoning before answering enables LLMs to undertake reasoning tasks prior to generating a response. This method allows the model to evaluate multiple hypotheses, assess evidence, and make logical deductions before providing a response. Incorporating reasoning into the decision-making process enables the model

to deliver more accurate and contextually appropriate answers, especially in complex tasks requiring logical analysis [Koj+23; Dia+24]. This approach significantly enhances the model's capability in understanding and responding to intricate questions, thereby improving its effectiveness across various applications, such as question-answering and dialogue systems [Koj+23].

The sequence of reasoning and answering is critical. Reasoning before answering guarantees that the model bases its responses on logical deductions and evidence, resulting in more accurate and trustworthy answers. Conversely, answering prior to reasoning may cause the model to provide a response first and then seek to justify it, potentially leading to incorrect or misleading answers [Koj+23; Dia+24].

2.3.4 Sampling and Generation Processes

Autoregressive Generation and Logit Sampling Autoregressive generation serves as a fundamental technique for numerous LLMs [Vas+23]. In this method, the prediction of each subsequent token is sequentially determined based on tokens generated earlier. The system's output for each new token relies on the context provided by previous tokens [Vas+23]. The next token's selection stems from sampling a probability distribution vector, known as logits, which represent the raw model outputs prior to their conversion into probabilities through the softmax function.

The sampling process converts these logits into a probability distribution via a softmax function. Tokens are then chosen based on their likelihood of appearing next.

To maximize the likelihood of the entire sequence, methods consider not only the immediate token but also the broader context of the sequence. Hence, various sampling strategies such as greedy, beam search, top-k, top-p, and nucleus sampling are employed to determine the next token from the logits' probability distribution [Sha+17]. In greedy sampling, the system selects the token with the highest probability at each step. In contrast, methods like beam search and top-k sampling assess several likely tokens to enhance the diversity of the sequence outcomes [Sha+17]. Top-k sampling identifies the k most probable tokens and randomly selects one from this group, whereas top-p sampling continues to select tokens until the cumulative probability surpasses a predefined threshold p [FLD18; Hol+20].

Techniques like Beam Search and Top-k Sampling: Functionality and Use Cases Beam Search, alongside stochastic methods like Top-k ([FLD18]) and Top-p ([Hol+20]), uses a deterministic approach to discover the most probable sequence of tokens. Introduced in [FAO17], Beam Search generates several paths concurrently, each representing a potential sequence. These paths, or beams, are extended until the se-

quence concludes. The optimal sequence from these paths is then chosen, facilitating the generation of multiple, diverse outcomes that are all plausible.

Beam Search is preferred in contexts demanding consistency and coherence, such as machine translation [Sha+17]. Conversely, Top-k and Top-p Sampling are suitable for scenarios requiring creativity or diverse outputs, such as in creative writing applications [SL22].

Impact of Sampling on Generation The selection of a sampling strategy hinges on the specific application: Top-k and Top-p Sampling promote variability but might yield less coherent outcomes, whereas Beam Search prioritizes coherence and precision, albeit potentially resulting in overly rigid outputs [SL22; Sha+17].

Enforcement of Output Formats The challenge of ensuring that generated text adheres to predetermined formats is addressed through various techniques. These include employing logical grammar structures as per [Gen+23], utilized in tools like Language Model Query Language (LMQL)[4], Guidance[5], and Outlines[6]. These methods limit the output to specific rules, for instance, generating text solely in uppercase or restricting responses to "Yes" or "No". The concept involves setting all logits to $-\infty$ for non-conforming tokens, ensuring that after softmax application, such tokens will not be selected due to their zero probability, thus compelling the model to produce the specified format [Gen+23].

A grammar can be defined that enforces a specific JSON schema, ensuring that the output is not only valid JSON but also adheres to a defined schema (see [7]).

The Structured Output Enforcing feature introduced by OpenAI[8] in August 2024 enables LLMs to directly generate structured outputs. This innovation could enhance both the efficiency and accuracy of various applications by allowing only precisely structured and parsable outputs. Although introduced subsequent to the completion of this work, it exemplifies increased control over model outputs.

2.3.5 Handling Long Contexts in LLMs

Limitations of Positional Embeddings The processing of extensive text sequences constitutes a formidable challenge for LLMs. As sequence length increases, maintaining

[4](visited on 10/01/2024) lmql.ai
[5](visited on 10/01/2024) github.com/guidance-ai/guidance
[6](visited on 10/01/2024) github.com/dottxt-ai/outlines
[7](visited on 10/01/2024) guidance.readthedocs.io/en/latest/generated/guidance.json.html
[8](visited on 10/01/2024) platform.openai.com/docs/guides/structured-outputs/introduction

an accurate representation of relationships between distant tokens becomes progressively more difficult, as during training, the model repeatedly has to compute these long sequence lengths which scale quadratic in memory and computational costs due to the nature of the self-attention mechanisms in Transformers. The computation of attention across all token pairs leads to a quadratic rise in computational complexity. As the model has to have seen the positional embeddings during pre-training, the model has to be trained on sequences of the same length as the ones it will be used on, making it expensive to train models on very long sequences [Su+24; Liu+24]. Therefore either training on long sequences with specialized hardware, limiting the context length, or using techniques to extrapolate positional embeddings for longer sequences is necessary.

Extrapolation of Positional Embeddings Dynamic RoPE scaling ([Su+24]) represents a method to mitigate the challenges associated with processing long sequences. RoPE introduces a modification to sinusoidal Positional Embeddings, enhancing the encoding of positional information within very long sequences. Dynamic RoPE scaling ([Liu+24]) extrapolates positional data from the training set to lengths not previously encountered. Initially, the model is trained using a large corpus of short sequences. Subsequent scaling of positional embeddings accommodates longer sequence lengths, which would otherwise be prohibitively expensive to train directly. Consequently, only a minimal application of SFT is necessary for adaptation to these extended lengths. This approach enables models to manage longer sequences than those encountered during training, thus enhancing their capacity to process varied data formats and structures [Su+24; Liu+24].

2.4 Retrieval Augmented Generation (RAG)

Concept of Vector Databases Vector databases, designed for efficient storage and retrieval of high-dimensional vectors, facilitate the comparison and search for vectors similar to a given query [Dou+24]. These numerical representations are generated by embedding techniques applied to text, images, or other data types, playing a crucial role in machine learning and NLP [Dou+24].

The primary function of these databases is the "nearest neighbor search", aimed at finding the vector closest in similarity to a query vector. Cosine similarity is frequently employed to quantify the similarity between two vectors, expressed by the formula:

$$\text{cosine similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

In this equation, $A \cdot B$ denotes the dot product of vectors A and B, while $\|A\|$ and $\|B\|$ represent their respective magnitudes. A cosine similarity of 1 signifies perfect alignment,

and -1, complete divergence.

Efficient computation of these measures, even in large datasets, is enabled by algorithms such as Approximate Nearest Neighbors (ANN) [Li+20; Dou+24].

Applications of Vector Databases in NLP Within NLP, vector databases predominantly support information retrieval tasks through semantic search. This technique involves storing embeddings of documents or passages (refer to section 2.2.3) in the database and retrieving those that are most semantically relevant to an embedded query vector [Lew+21]. By capturing the semantic content of text, the system enables searches that are aware of contextual nuances. Documents that exhibit the highest semantic similarity to the query are identified as containing the most pertinent information [Lew+21].

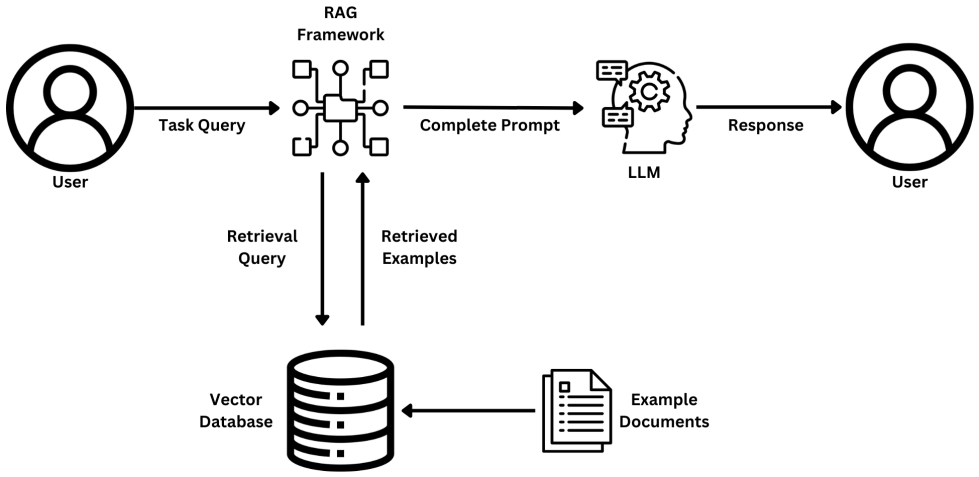

Figure 5: Illustration of the RAG with a vector database.

RAG incorporates this functionality into the inference process, initially enriching the prompts with context-specific information retrieved from the vector database [Lew+21] before generating responses (see section 2.3.3). By integrating knowledge from external sources, the model is capable of producing responses that are both relevant and richly informative [Lew+21]. Acting as a knowledge repository, the vector database equips the model with additional context, enhancing output quality [Lew+21].

This capability is especially vital in scenarios like question-answering or dialogue systems, where relevant external information may not be present in the training data. Through the use of RAG, the model gains the ability to "learn" from an external knowledge base, thus offering more informed responses even post-training [Lew+21].

2.5 Methods for evaluating Competence Rankings

This subsection explores various evaluation methods used for ranking competence profiles. It covers scoring methods, which offer straightforward numerical evaluations but are susceptible to biases and oversimplification. The discussion then transitions to pairwise preference assessments, which mitigate some of these biases by focusing on relative comparisons rather than absolute scores. Further, it examines tournament structures designed to enhance the efficiency and accuracy of these comparisons, and concludes with an overview of the Elo rating system, which provides dynamic, ongoing ranking updates.

2.5.1 Scoring Methods

Scoring methods are commonly utilized for assessing competence profiles, providing a direct means to attribute numerical values to each profile. These values facilitate the ranking of profiles in a straightforward and scalable fashion. Despite their widespread use, scoring methods exhibit numerous limitations, particularly in the context of complex, multi-dimensional tasks [Fre+21].

A fundamental issue with numerical scoring is its vulnerability to biases. Evaluators frequently employ arbitrary scales, and the absence of a robust framework for comparison can result in marked inconsistencies in outcomes. Moreover, scoring methods are profoundly affected by external variables such as the evaluator's background, expertise, and contextual knowledge, potentially introducing subjective biases [Fre+21]. The lack of a definitive reference or standard complicates the assurance of reliable and uniform evaluations across various evaluators [Fre+21].

Additionally, competence profiles inherently encompass multiple facets—including expertise, abilities, and contextual pertinence—that challenge condensation into a single score. This reductionism may yield rankings that do not accurately reflect the true complexity of the profiles, thereby obscuring significant distinctions among them [Fre+21]. Thus, while scoring methods offer simplicity, they frequently do not adequately represent the nuanced variances among competence profiles in intricate evaluation scenarios [Fre+21].

2.5.2 Pairwise Preference Assessment

Pairwise preference assessments provide a robust alternative to conventional scoring methods. Instead of assigning a definitive score to each profile, this approach involves evaluating two profiles simultaneously to determine the preferable one [NOS17]. This technique eliminates the need for a comparative external reference and promotes more consistent evaluations, which are solely based on relative comparisons.

A key benefit of using pairwise preference assessments is the reduction of subjectivity that is typically present in scoring methods. By concentrating on comparative judgments instead of fixed scores, evaluators are enabled to conduct more direct and meaningful comparisons between profiles [NOS17]. This approach is especially advantageous in scenarios where establishing a definitive baseline is challenging, or when the focus is on the comparative performance of profiles.

Efficiency Through Tournament Structures To enhance the efficiency of pairwise comparisons, tournament structures are commonly utilized. In these structures, profiles undergo comparisons in sequential rounds, each round decreasing the number of profiles under consideration [Gas98]. As profiles progress through the tournament, only the most informative comparisons persist, thus minimizing the total evaluations needed.

This organizational method greatly increases efficiency while ensuring a high level of accuracy in the resultant rankings. Adaptive algorithms may be employed to dynamically select pairs for comparison based on prevailing uncertainties, optimizing the evaluation process further [Gas98; NOS17].

2.5.3 Knockout Tournaments and Trade-offs

The knockout tournament represents a specific type of tournament applied in pairwise preference assessments. In this structure, profiles are evaluated in a head-to-head manner, resulting in the advancement of one profile and the elimination of the other. This procedure is repeated until a final winner emerges, ensuring that all profiles are evaluated [Gas98].

The primary advantage of the knockout tournament is its efficiency, characterized by the minimal comparisons required. For N profiles, merely $N-1$ comparisons are necessary to establish a conclusive ranking. Initially, $N/2$ comparisons occur in the first round, $N/4$ in the subsequent round, $N/8$ in the third, continuing in this pattern until only one profile remains. Such a method guarantees a highly efficient assessment process, utilizing the fewest possible comparisons to rank the profiles [Gas98].

Nonetheless, this method presumes the transitivity of preferences among profiles. Specifically, if Profile A is preferred to Profile B, and Profile B is preferred to Profile C, then it is expected that Profile A will be preferred over Profile C. This assumption might not consistently be valid due to the influence of particular contextual elements or biases of individual evaluators. Moreover, as the tournament does not facilitate every potential comparison, it is conceivable that some strong profiles might be prematurely eliminated, which could result in a less than optimal final ranking.

Despite these potential drawbacks, knockout tournaments are frequently selected for their

efficiency, especially in situations where computational resources are scarce or when a rapid assessment is essential, with the primary focus on identifying the overall winner. The balance between speed and precision is a crucial factor in choosing this methodology for competency profile evaluation.

2.5.4 Elo Rating System

In contexts requiring nuanced and dynamic rankings, the Elo rating system provides a viable alternative to knockout tournaments. Initially designed for evaluating chess players, the system updates the ratings of profiles following each pairwise comparison [Ing21; AV01]. A victory in a comparison results in an increase in the winning profile's Elo rating, while the rating of the losing profile declines. Such continuous updates facilitate the progressive refinement of profile rankings with each additional comparison [DAG15; AV01].

The Elo system proves highly effective for continuously ranking and updating profiles. Its adaptability allows it to consider various factors, including the margin of victory, and it supports enhancements through Bayesian models to integrate further context like task complexity or profile stability [Ing21; AV01]. Although it necessitates more comparisons than a knockout tournament, the Elo system yields a more detailed and precise approach for sustained ranking.

2.6 Automatic Evaluation vs. Expert Evaluation

The integration of LLMs such as GPT-4 has facilitated enhancements in automatic evaluation. This process involves the model emulating human-like judgment to assess task outputs. Automatic evaluation provides the benefits of rapid processing of large datasets, significant cost reductions, and the continuous availability that eliminates the need for human evaluators [Zhe+23; CL23]. The utilization of models such as GPT-4 to replicate expert judgment in tasks like skill assessment minimizes the manual labor involved in evaluation, presenting a scalable alternative [Zhe+23; CL23].

Nevertheless, several challenges persist due to the inherent limitations of these models. Automated systems may demonstrate biases that are introduced during their training phase, such as positional bias, leading to skewed results depending on the sequence in which data is introduced to the model [Wan+24; WHB14; Zhe+23]. In particular, GPT-4 is prone to verbosity bias, a tendency to prefer lengthier answers, thereby ranking longer responses more favorably, irrespective of their substantive quality [Hua+24; Anw+24; Zhe+23].

2.6.1 Comparison with Expert Evaluation

Expert Evaluation as the Gold Standard Expert evaluation remains the gold standard in evaluation methodologies, attributed to the unique qualities of human experts. Their capacity to apply in-depth, domain-specific knowledge, extensive experience, and subtle human intuition enables them to deliver assessments deemed highly reliable [Zhe+23; CL23]. Such evaluations provide insights that automated systems struggle to match, especially in context-specific decision-making and subjective evaluations, such as interpreting ambiguous or intricate competence profiles [Zhe+23; CL23].

Challenges of Expert Evaluation Nonetheless, expert evaluation encounters significant practical difficulties. The costs associated with engaging experienced professionals are typically substantial, and the time required to conduct thorough assessments can be considerable [Zhe+23; CL23]. Additionally, the inherent subjectivity of expert evaluations introduces the risk of inconsistency among different evaluators. Subjective biases, such as confirmation bias or an overestimation of personal judgments, can compromise the accuracy of these evaluations [CL23; Wan+24]. The mental or emotional state of experts, as well as potential fatigue from repeated evaluations, may also distort outcomes, contributing to their variability [RHU01].

Connection Between Automatic and Expert Evaluation The increasing deployment of automated systems for competency extraction offers opportunities to bridge the gap between automatic and expert evaluations. Although automatic systems deliver faster, more economical solutions, their results often require validation by human experts to ensure accuracy [CL23; Zhe+23]. By comparing the outcomes from both automatic and expert evaluations, a more comprehensive analysis of their correlation can be conducted, highlighting the strengths and limitations of each method. Studies have shown that when calibrated properly, automated systems can produce results that closely align with human judgments, though addressing subjective contexts remains a challenge [CL23; Zhe+23].

2.6.2 Biases in Evaluation and Their Handling

Overview of Biases in Evaluation The design of prompts is critical in evaluations utilizing automatic LLMs, significantly influencing performance outcomes. Even minor changes in phrasing, such as adjustments in word selection or punctuation, can cause notable variations in results, referred to as *prompt sensitivity* [Anw+24; Wan+24]. This variability is evident not only in complex prompts but also in simple paraphrasing, where seemingly minor modifications, such as substituting a word or inserting punctuation, can

trigger significant shifts in the model's accuracy [Miz+24]. For instance, a single term change in an instruction can alter performance by up to 28% in models like Flan-T5-large [Miz+24]. This inconsistency underscores the challenge of establishing a universally effective prompt across different models and tasks [Anw+24].

Prompt sensitivity complicates the evaluation process, and several inherent biases in LLMs further challenge the accurate assessment of their capabilities. A notable example is the *self-enhancement bias*, where models exhibit a preference for outputs they have generated themselves or those from similar models. This bias, particularly observed in commercial systems like GPT-4, tends to inflate the perceived quality of self-generated outputs [LPD24; Liu+23]. For example, GPT-4 shows a preference for outputs from GPT-3.5 over those from different sources [LPD24].

Another significant bias is the *position bias*, which affects how models evaluate responses based on their presentation order. The position of a response can influence the model's preference, with variations noted across different systems. For instance, GPT-4 often favors responses that are presented first in a pairwise comparison [Wan+24]. Strategies such as altering the order of comparisons, employing explanation-based methods to validate model choices, or demanding consistent results irrespective of sequence by evaluating self-consistency have been proposed to mitigate this bias [Wan+24; Zhe+23; Min+24b].

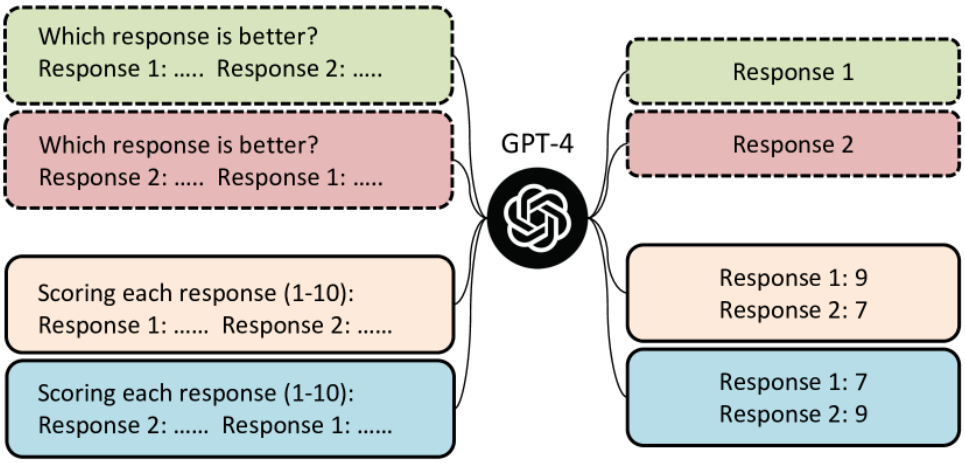

Figure 6: Positional Bias in Evaluation of LLMs from [Wan+24]

In fig. 6, the figure demonstrates how response order can impact model preferences. The model consistently scores the first-presented response higher, regardless of its quality.

The *verbosity bias* also influences model preferences, favoring more detailed responses. Systems such as GPT-4 and other fine-tuned models typically prefer lengthier, more elaborate responses, even if more succinct responses might be of superior quality [Hua+24; Zhe+23]. This tendency stems from training objectives that emphasize generating fluent and coherent text, which often results in an overemphasis on length [Hua+24].

Addressing these biases and the challenge of prompt sensitivity is essential for ensuring reliable and consistent evaluations of LLMs. Without careful consideration of these factors, evaluations may not accurately reflect a model's true capabilities, influencing both research outcomes and practical implementations.

Strategies for Reducing These Biases

- **Automatic Prompt Testing:** Employing automatic prompt testing is a primary method for reducing prompt sensitivity. This strategy involves testing multiple prompt variations systematically to find those that yield consistent results across different models [Kha+23; SC21].
- **Random Arrangement of Samples:** To address positional bias, samples should be arranged randomly within prompts. This method ensures no sample gains an advantage due to its position, thereby diminishing the likelihood of biased comparisons [Wan+24; Zhe+23]. Repeated randomization is advisable to neutralize any lingering effects of positional bias throughout the evaluations [Wan+24; Zhe+23].
- **Anti-Verbosity Prompting:** Implementing anti-verbosity prompting techniques can help minimize verbosity bias. These techniques encourage the model to prioritize concise, substance-rich outputs over longer, more detailed but potentially superficial responses [Anw+24].
- **Different Models for Generating vs. Evaluating:** Using different models for generating competence profiles and for their evaluation helps counter self-enhancement bias. This separation ensures that the evaluating model is unbiased by its own prior outputs. Employing a panel of models for the evaluation process, rather than a single system, further reduces bias potential [LPD24; Liu+23].

2.7 Fine-tuning of LLMs

This subsection explores the fine-tuning processes applied to pre-trained LLMs. It highlights the transition from broad, general pre-training phases to more targeted fine-tuning steps that refine these models for specific tasks. The focus is on enhancing the model's precision and task-specific performance through strategic training on smaller, specialized datasets. Discussions include various techniques such as Supervised Fine-Tuning (SFT), Reinforcement Learning from Human Feedback (RLHF), and Direct Preference Optimization (DPO), each tailored to optimize LLMs in unique ways that balance efficiency with computational demands.

2.7.1 Basics of Fine-tuning

Fine-tuning denotes the adaptation of a pre-trained LLM for a specific task by additional training on a smaller, task-specific dataset [Bro+20]. The process of pre-training incorporates extensive general data in a self-supervised manner, while fine-tuning focuses on task-specific learning to tailor the model's outputs to particular objectives. Such alignment enhances the model's ability to deliver more precise outcomes for specialized applications [Bro+20].

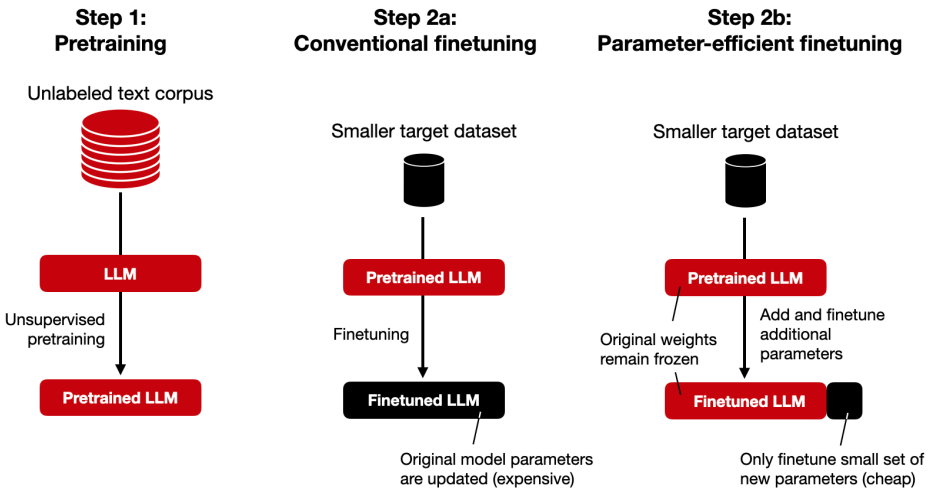

Figure 7: Fine-tuning Process for LLMs from [Ras23a]

During the pre-training phase (Step 1 in fig. 7), the model employs self-supervised learning to develop representations from extensive volumes of unlabeled data [Bro+20]. This stage allows the system to establish a comprehensive grasp of general language patterns, word associations, and syntactic structures. Nevertheless, without further refinement, pre-training does not achieve peak performance for specialized tasks, such as named entity recognition or sentiment analysis, which require specific knowledge.

Fine-tuning (Step 2a in fig. 7) addresses this necessity by enabling the system to assimilate task-specific nuances, typically through SFT [Bro+20]. In this phase, the pre-trained model's weights are refined as the system is trained on a labeled dataset designed for the specific application. This procedure focuses the model's capabilities, significantly boosting its accuracy and enabling effective generalization within the designated task domain.

The contrast between pre-training and fine-tuning underscores the adaptability of LLMs. While pre-training furnishes a broad comprehension of language, fine-tuning refines this foundation to fulfill the precise requirements of particular tasks, thereby enhancing both precision and relevance in task outcomes [Bro+20].

2.7.2 Reinforcement Learning from Human Feedback (RLHF)

RLHF, as presented in the study by Ouyang et al. [Ouy+22], aligns LLMs with human preferences by incorporating human feedback into the training regimen of the model. Unlike traditional supervised learning that relies on direct labeling, this method prioritizes human qualitative feedback to enhance model performance in generating useful and safe responses. The fundamental concept of RLHF is that distinguishing between adequate and inadequate responses is significantly simpler than creating the former. Consequently, a model requires considerably fewer data to discern effectively between these responses. Such discernment then serves to automatically direct the training process towards generating high-quality responses. Essentially, the model is trained to produce beneficial responses with minimal direct examples, relying instead on limited feedback indicating the quality of each response.

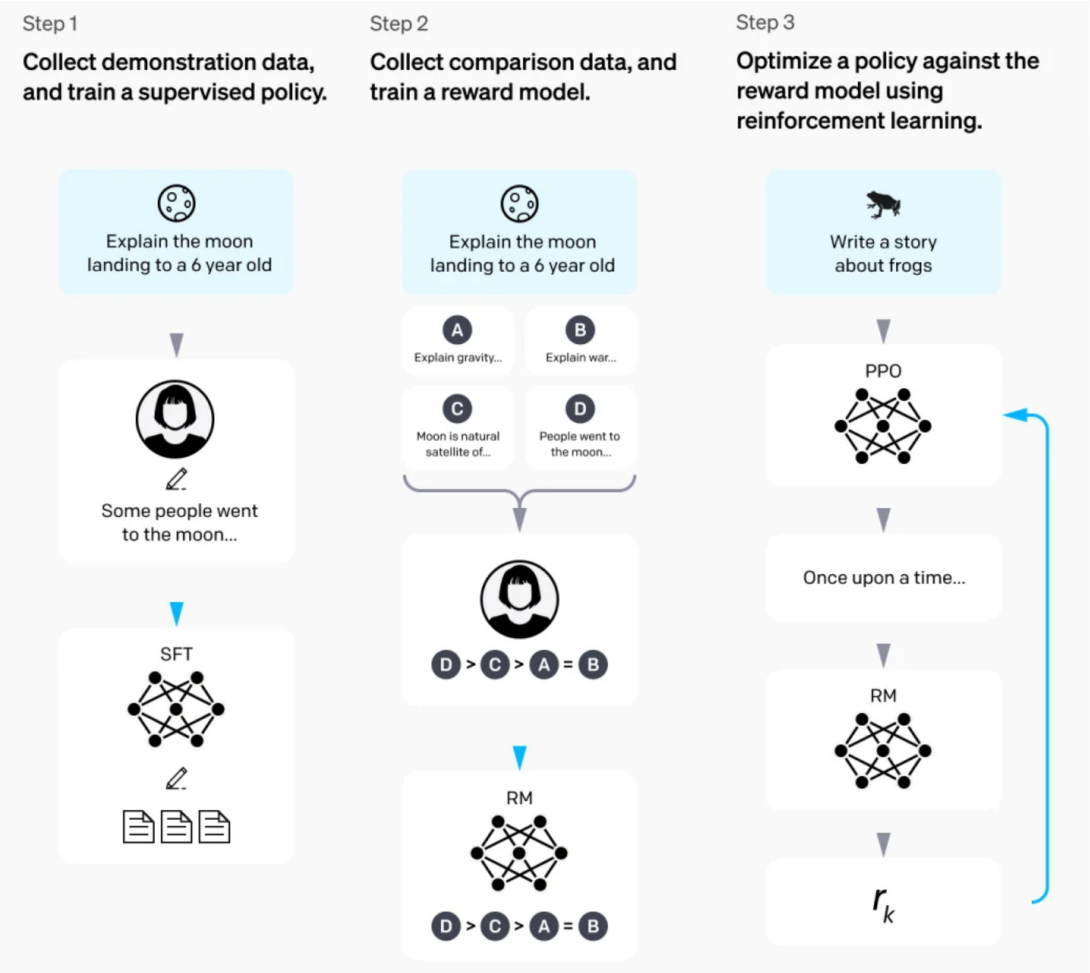

Figure 8: Reinforcement Learning from Human Feedback (reworked) from [Ouy+22]

Human Feedback and Reward Model Training The process to train RLHF models begins with the pre-training of a LLM based on general text data or specifically labled

data (depicted as Step 1 in fig. 8), which is afterwards fine-tuned using RL methods. The next step is to train the reward model. This is trained using human feedback such as pairwise comparisons of model outputs, to score responses based on their consistency with human expectations (shown in Step 2 in fig. 8). It is critical to manage this feedback meticulously to prevent the introduction of bias or errors into the training process. The reward model plays a pivotal role in furnishing a dependable signal that steers the LLM toward producing responses that meet human preferences, ensuring careful handling to avoid noise and bias. In the last phase, the LLM is refined to optimize the reward signal generated by a reward model (illustrated in Step 3 in fig. 8), no longer requiring human annotations, rather relying on accurate rewards from the trained reward model.

Training Challenges and Hyperparameter Sensitivity The training dynamics are notably influenced by hyperparameters, such as the learning rate and the reward signal, where the relationship between the reward model and the LLM greatly affects the outcome and presents substantial optimization challenges. It is crucial to ensure that this interaction remains stable and effective to achieve successful RLHF training.

Efficiency and Data Requirements RLHF tends to be more data-efficient compared to traditional supervised learning, as it utilizes the reward model to incorporate feedback effectively. This method is especially advantageous for tasks that demand precise, context-sensitive responses. Although generating human feedback is resource-intensive, RLHF minimizes the need for extensive labeled datasets, thereby reducing costs. This reduction in human labor for data collection significantly lessens the overall effort required to equip the model with necessary data, enhancing cost-efficiency. By only requiring feedback on the quality of responses, rather than labeled data, RLHF can be more cost-effective than traditional methods.

2.7.3 Direct Preference Optimization (DPO)

This section presents DPO from the publication [Raf+24], a fine-tuning methodology that aligns a pre-trained LLM with preference data through direct optimization, providing a simpler and more efficient alternative to RLHF. The foundation of this section is the research conducted by the authors.

Concept and Functionality DPO is a method for fine-tuning a pre-trained LLM that aligns it with human preferences through direct optimization, rather than through the indirect methods utilized in RLHF. It employs a distinct reward model parameterization, enabling the extraction of the optimal policy in a closed form and thereby eliminating the

necessity for a RL training loop. The core principle of DPO involves directly optimizing a classification objective to maximize the probability of the preferred completion (y_w) and minimize that of the non-preferred completion (y_l).

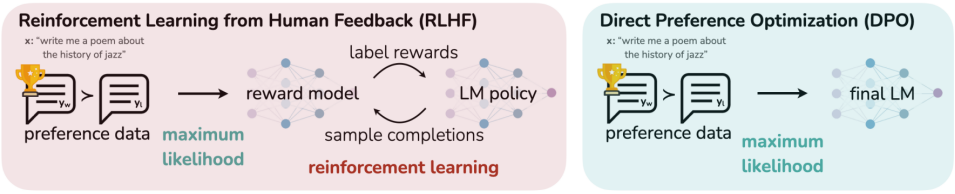

Figure 9: Comparison of RLHF and DPO from [Raf+24]

This demonstrates the simplicity and effectiveness of DPO, addressing the challenges associated with more complex methods like RLHF, which necessitate extensive hyperparameter adjustments and can result in instability. DPO reduces errors and facilitates manageable optimization, proving advantageous in resource-constrained environments. Existing studies indicate that models based on DPO can attain performance on par with or superior to models tuned using RLHF in NLP applications, such as sentiment control and dialogue quality.

Dataset Format and Model Adaptation DPO utilizes a dataset comprising triplets: an input prompt (x), a preferred completion (y_w), and a non-preferred completion (y_l). The fine-tuning involves establishing a reference model (π_{ref}) with static weights to act as a baseline, while the secondary model (π_θ) undergoes updates. This strategy ensures the updated model retains proximity to the original pre-trained behaviors.

The reward function in DPO is:

$$r_\theta = \beta \cdot \log \left(\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} \right)$$

This framework simplifies the alignment process by directly evaluating completions according to their likelihood of reflecting user preferences.

The reward is integrated with a Kullback-Leibler Divergence (KL) term to establish the DPO loss function:

$$L_{DPO}(\pi_\theta, \pi_{ref}) = -\mathbb{E}_{x,y_w,y_l} \left[\log \sigma \left(\beta \cdot \log \left(\frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} \right) - \beta \cdot \log \left(\frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)} \right) \right) \right]$$

where β is a non-negative constant hyperparameter that governs the deviation from the base reference policy (π_{ref}).

Significance and Implications DPO provides several advantages over RLHF:

1. **Simplicity**: DPO obviates the need for a separate reward model, thus reducing the complexity and resource demands of the fine-tuning process.
2. **Stability**: By avoiding potential misalignments or overfitting associated with reward models, DPO fosters more stable training outcomes.
3. **Efficiency**: This approach enhances efficiency, making it well-suited for fine-tuning using consumer-grade hardware.
4. **Dataset Creation**: The process for creating datasets is streamlined, requiring only two distinct completions per prompt, which are straightforward to evaluate using pairwise comparisons as opposed to the scoring required by RLHF (see section 2.5.1).

Comparison with Other Fine-Tuning Techniques DPO stands as a primary alternative to RLHF and Proximal Policy Optimization (PPO), both of which necessitate a reward model to ensure the model output aligns with human preferences. Both RLHF and PPO are characterized by their complexity, instability, and high computational demands. PPO faces challenges in maintaining reward model consistency with the pre-trained model, resulting in issues such as mode collapse and sample inefficiency.

DPO circumvents these challenges by simplifying the preference optimization process, which enables faster convergence and more stable performance. It forgoes the implicit dependency on a reward model, as seen in RLHF and PPO, and instead directly optimizes the model based on rewards. In practice DPO has demonstrated almost identical performance to RLHF while offering a more straightforward and efficient training process. This method is particularly effective given the dataset format utilized in this work, which relies on pairwise comparisons rather than pointwise reward estimations, thereby providing the DPO dataset format instead of the scores required for the reward model in RLHF.

2.7.4 Parameter-efficient Fine-tuning

Introduction to LoRA and Other Parameter-efficient Fine-tuning Techniques Parameter Efficient Fine-Tuning (PEFT), as outlined by [LMM23] and [Hu+21], minimizes the computational burden necessary to tailor large pre-trained models for specific tasks while preserving performance levels. Techniques such as Low-Rank Adaptation (LoRA) employ adapters that update only a subset of the model parameters during task adaptation, thereby reducing memory, storage, and computational demands (refer to Step 2b in fig. 7). These strategies are advantageous in resource-constrained settings, such as

edge devices or configurations with a single Graphics Processing Unit (GPU), where extensive fine-tuning is unfeasible due to substantial memory and processing requirements.

The principle is that minor modifications to the flow of information through the model are sufficient to adjust it for a new task. This is because the model already possesses a robust grasp of the language and requires only minimal adjustments. To this end, low rank matrix adapters are implemented to subtly direct the flow of information, while keeping computational costs low.

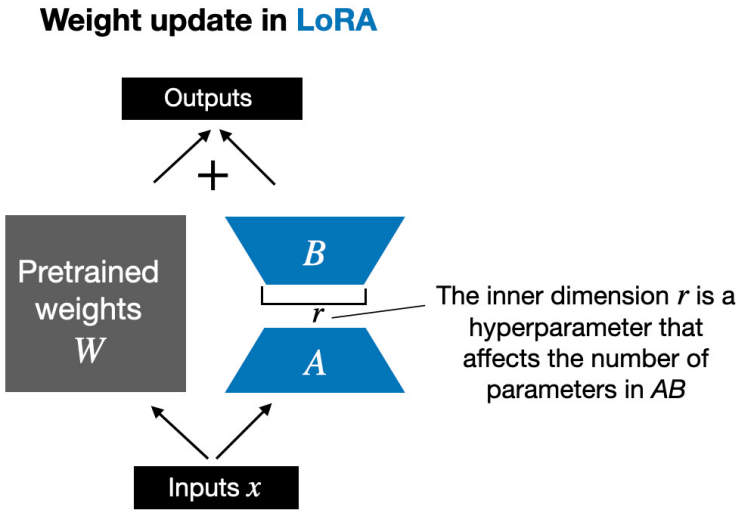

Figure 10: Adapters for PEFT from [Ras23b]

The blue fully connected layers depicted in fig. 10 comprise two matrices that map from $d_{hidden} \times r$ to $r \times d_{hidden}$, where d_{hidden} denotes the hidden dimension of the model and r indicates the rank of the adapter. Typically, the rank r is selected to be relatively low (16-64), representing the adjustable addition to the information flow through the model. This adjustment is achieved by augmenting the standard output of the model's pre-trained weights.

Quantized Low-Rank Adaptation (QLoRA) ([Det+23]) further reduces memory usage by implementing quantization techniques to decrease the model's size, notably through 4-bit quantization. By maintaining most model parameters static and executing fine-tuning via LoRA, QLoRA facilitates the fine-tuning of large models with significantly diminished memory requirements while maintaining performance.

Relevance of Parameter-efficient Fine-tuning for Resource-constrained Scenarios In scenarios with limited resources, PEFTs prove indispensable. Comprehensive fine-tuning of large models is unmanageable due to their substantial memory and pro-

cessing demands. For comparison, consult table 1[9] below, which contrasts the memory requirements for full fine-tuning with those for PEFT using LoRA and QLoRA:

Model Size	Full Fine-tuning	LoRA	Q-LoRA
8B	60 GB	16 GB	6 GB
70B	500 GB	160 GB	48 GB
405B	3.25 TB	950 GB	250 GB

Table 1: Training Memory Requirements for Different Fine-tuning Techniques for the Llama 3.1 Model Family

[9]Source: (visited on 10/01/2024) huggingface.co/blog/llama31

3 Current State of Research

This chapter provides an overview of recent advancements in the field of competency extraction and associated technologies. It concentrates on techniques for deriving competencies from job postings and scholarly documents.

3.1 Introduction to the Current State of Competence Extraction

Competence extraction has emerged as a pivotal process within both research domains and professional settings, driven by the escalating need for specialized skill sets across diverse sectors, such as academia and industry [GEA19a]. In academic environments, the extraction and profiling of competencies facilitate the precise identification of expertise, crucial for establishing research collaborations, making informed hiring decisions, and fostering academic advancement [Bay+18]. Methods including profile-centric techniques and the application of NLP are utilized to create these profiles through the analysis of textual data derived from publications and other scholarly communications [Bay+18]. These methods provide insightful observations yet encounter obstacles in managing the extensive data volume while ensuring accuracy.

Similarly, in professional contexts, competence extraction plays a vital role in aligning educational achievements with the demands of the job market [Bö+18]. Strategies for incorporating regional competence profiles into educational frameworks are devised to ensure that the capabilities of graduates align with industry requisites [Bö+18]. This alignment is especially significant in fields such as Human Resources (HR) development and organizational enhancement, where the definition and monitoring of employee competencies are directly linked to productivity and innovative outcomes [AS15].

The utility of competence extraction spans various applications, including academic profiling for scholars and organizational profiling for industries [Bö+18]. By developing precise competency profiles, both academic and industrial sectors realize benefits such as improved efficiency in skill identification, recruitment processes, and planning for developmental initiatives [AS15]. Systems like ScholarLens exemplify this by automating the creation of user profiles through the analysis of research publications, thereby facilitating the accurate pairing of researchers with projects or collaborative opportunities [Sat+17].

3.2 The AIFB Competence Pool

The AIFB Competence Pool[10] at the KIT functions as a platform to facilitate connections among researchers by rendering their expertise and competencies easily discoverable. The

[10](visited on 10/01/2024) bis.aifb.kit.edu/317_389.php

competence pool primarily aims to assist KIT staff, guests, and external partners in identifying relevant experts for project collaborations, interdisciplinary research, and various initiatives. In a secure environment, researchers create individual profiles where they define their competencies either using keywords from a predefined catalog or by inputting custom terms. These competencies are then associated with KIT centers and research topics, allowing researchers to integrate their publications via the KITopen library service.

The search functionality of the competence pool enables users to locate experts based on the competencies defined and their connections with KIT centers. This feature promotes efficient collaboration by facilitating targeted searches for specific areas of expertise, thereby increasing the visibility and accessibility of the research community both within and outside KIT.

At present, competencies in the AIFB Competence Pool are predominantly manually extracted and defined by the researchers. Nonetheless, the system includes a pilot feature that automates this process by extracting competencies directly from academic publications using NLPs and machine learning techniques. Although still in its nascent stages, this feature marks an important progression towards minimizing manual effort and enhancing the scalability of the competence pool.

The current limitations of this experimental system underscore the necessity for more refined methods, especially the inclusion of LLMs. These models hold considerable promise for advancing the precision and efficiency of competence extraction through their ability to manage extensive datasets and intricate text relationships. The research presented in this thesis explores the deployment of LLMs for competence extraction. The findings are expected to be incorporated into the backend of the AIFB Competence Pool, potentially augmenting the system's ability to automatically profile researchers and thus, improving the usability and accessibility of the pool for future users.

3.3 Skills Extraction in Other Domains

Methods of Skills Extraction in Job Postings The extraction of skills from job postings represents a significant application of NLP and machine learning [Zha+22]. A key technique involves multi-class classification, in which models such as BERT are trained on synthetic datasets to determine job postings' required skills [Zha+22]. This method improves model efficacy, especially in scenarios where actual training data are limited, as illustrated by the deployment of LLMs like text-davinci-003 and Falcon 7b to produce synthetic job postings [Rah+23].

Additionally, the application of weak supervision methods helps to mitigate the intensive

nature of annotating job postings [Mas+23]. These methods utilize frameworks like the European Skills, Competences, Qualifications, and Occupations (ESCO)[11] for skill extraction, achieving enhanced precision [Mas+23]. By employing latent representations instead of token-level matching, these systems exhibit substantial improvement over traditional supervised models [Vuk+21].

The sector faces challenges including the diverse formats of job postings, the frequent lack of explicit skill labels, and the need to process unstructured text [Ngu+24]. Additionally, job postings often contain implicit skills—those not explicitly stated but implied through context, such as industry-specific knowledge. Models employing semantic similarity techniques like Doc2Vec have been developed to identify these implicit skills [LY18].

In contrast, extraction in scientific domains demands structured data retrieval from precise and highly specialized research papers, posing increased challenges due to the necessity for accuracy and domain-specific expertise in competency extraction [GEA19a].

Other Relevant Areas of Application Beyond job postings, skill extraction is crucial in HR management and organizational development. In HR management, machine learning models are used to construct competency databases that support talent management and deployment [Mau+18]. Techniques such as Optical Character Recognition (OCR) and NLP are utilized to develop competency matrices from employee records, aiding managers in making decisions about skill development and resource allocation [Tia+23; JT21].

Furthermore, in organizational development, the extraction of both hard and soft skills from job advertisements aids in the understanding of labor market trends. Research indicates that tailoring NLPs models to specific domains like job postings greatly enhances skill identification accuracy, which subsequently influences decisions in organizational training and development initiatives [JT21; AS15].

The strategies employed in HR management and job postings, such as NLP-based skill extraction and competency matrix creation, lay the groundwork for expanding competency extraction methods across various fields. These techniques foster scalable, automated competence identification, increasingly pertinent in both academic and corporate environments [Mau+18; AS15].

In summary, the extraction of skills and competencies across diverse domains, from academic settings to job postings, underscores the critical role of advanced techniques such as NLP and machine learning. The challenges identified, ranging from data volume management to the precision of skill labeling, highlight the necessity for innovative, adaptable solutions.

For this thesis, these insights reinforce the decision to employ State-of-the-Art (sota)

[11] (visited on 10/01/2024) esco.ec.europa.eu/en

LLMs. By leveraging their sophisticated capabilities in handling extensive datasets and complex text structures, the aim is to significantly enhance the accuracy and efficiency of competence profiling. This approach promises to advance the effectiveness of competence extraction systems, enabling more dynamic and valuable applications in both academic and professional environments.

4 Methodology

This chapter delineates the approach for developing and assessing methods of competency extraction. It details techniques for deriving competencies from extensive scientific texts, ranging from abstracts to entire documents. The chapter introduces methods for LLMs fine-tuning with a focus on DPO and domain adaptation strategies such as RAG. It also covers data sources, preprocessing steps, and evaluation metrics utilized in determining the performance of the system.

4.1 Overview of the Methodology

The primary goal of this system is to develop the most effective competency extraction framework possible for analyzing and summarizing competencies from scientific documents and other extensive texts.

While current LLMs can process individual long-form documents within a single prompt, our challenge lies in extracting competencies from multiple documents to construct a more comprehensive competency profile. To address this, various context reduction techniques designed to manage the integration of information from multiple texts were implemented. These techniques form the core of our methodology for enabling the extraction of competencies across numerous documents.

Before initiating the context reduction methods, the selection criteria and data sources for the documents to be processed were determined (see section 4.2). Different context reduction methods are then defined and implemented (see section 4.3), each serving as a distinct approach to competency extraction. To enhance the prompts, retrieval augmented generation methods are incorporated, utilizing examples to clarify the extraction tasks (see section 4.4). Additionally, the competencies are parsed into a structured format, facilitating further processing and application in varied contexts (see section 4.5). An automated pre-evaluation system then assesses these methods alongside their hyperparameters to highlight the most effective models, balancing model size for feasible fine-tuning against performance efficacy (see section 4.7).

Once a suitable model is pinpointed, it undergoes fine-tuning specifically for the competency extraction task (see section 4.8).

This process is complemented by a detailed expert evaluation phase, where competency profiles generated from individual researchers' works are assessed by the researchers themselves. This ensures that the extracted competencies accurately reflect the expertise demonstrated in their publications (see section 6.1).

Ultimately, the insights gathered through this rigorous methodology aim to establish

recommendations for the best models, methods, and settings for competency extraction (see section 7.2). The design of the system is such that it not only excels in specific domains but is also capable of handling various types of documents, facilitating domain-agnostic applications (see section 4.9). The robustness of this approach will be further validated through extensive testing on a diverse set of company data provided by CAS Software AG (CAS), ensuring the system's effectiveness across different domains and document formats (see section 4.10).

This section sets forth a comprehensive blueprint for each phase of the methodology, collectively aimed at developing and validating an advanced, versatile competency extraction system capable of navigating the complexities of modern data-rich environments.

4.2 Data Sources and Selection Criteria

Data Sources The sources of data for competency extraction initially include scientific publications, specifically articles and conference papers (section 4.10 discusses applications in other domains). These sources are chosen for their broad coverage of research contributions and the expertise of scientific personnel. Each document offers an in-depth presentation of the author's research and is appropriate for the creation of competency profiles derived from the content.

Criteria for Document Selection To maintain a suitable balance between relevance and scope, this system selects the five most-cited papers of each individual as the primary data source. This selection is predicated on the premise that a paper's citation frequency is indicative of its significance and the author's area of expertise. By restricting the analysis to five papers, it ensures the total input for LLMs remains within manageable limits, accommodating all paper abstracts or summaries within the prompt. Longer contexts would necessitate additional merging steps, potentially introducing noise in the compilation of profiles from numerous documents [Yan+24] (refer to section 4.3 (*extract from full texts*) for additional details on the merging process).

Preprocessing of the Data The preprocessing of selected documents is recommended to enhance the accuracy and efficiency of competency extraction [Yan+24; PLW19]. The text extraction process from PDF documents may incorporate non-essential sections such as references, attachments, and supplementary information. Removing these elements is crucial as they may interfere with the extraction process. A range of methods, including manual curation, algorithmic parsing, and the utilization of LLMs for preprocessing, are utilized to optimize this step and ensure the input is clean and suitable for the extraction model [PLW19; Yan+24].

4.3 Methods for Competency Extraction

The competency extraction system utilizes various techniques to extract competencies from scientific texts efficiently. These techniques are tailored to handle the complexity and extensive nature of scientific literature while maintaining the accuracy and relevance of the extracted competencies. The system employs LLMs and context reduction strategies to optimize the extraction process and produce detailed competency profiles.

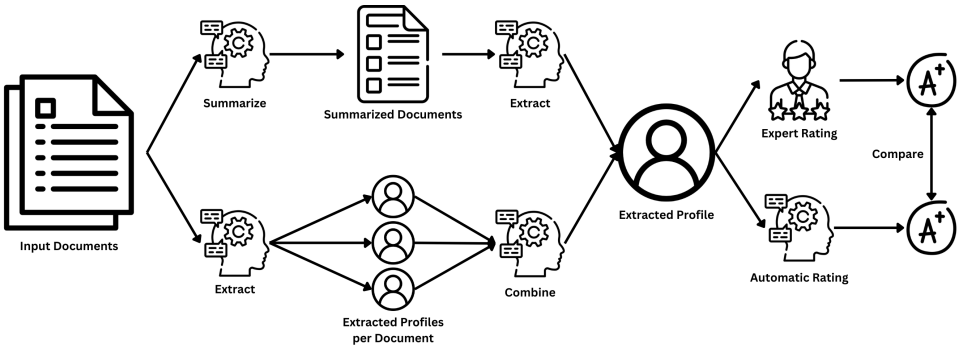

Figure 11: Data Flow Diagram of the Competency Extraction System

Extract from Summaries The initial method for competency extraction utilizes LLMs to generate summaries of entire documents (see fig. 11 upper extraction method), which facilitate the extraction of competencies. This method effectively condenses critical information from lengthy documents into a succinct format, thus addressing the challenge posed by model input size constraints. Thereby only one document has to be processed at a time, limiting the required context length.

Summarization techniques, extensively researched within the NLP domain, show sota capabilities of LLMs in producing coherent and informative summaries. These capabilities enable the system to generate precise and succinct summaries of scientific documents [Wid+20].

The summaries produced by LLMs encapsulate the principal points, methodologies, and outcomes of the original documents, providing a compressed overview of the content. Competencies are subsequently extracted from these summaries during another LLM session where the context length is considerably reduced, allowing for precise and efficient competency extraction.

Nonetheless, summarization may inadvertently miss subtle or less prominent details in the full text. This oversight could result in the loss of critical information or the exclusion of important competencies.

Extract from Abstracts Competency extraction from abstracts employs a similar approach to that from summaries, capitalizing on the fact that paper authors have already distilled the most significant information into the abstracts. The abstracts from all documents are collectively processed through the model to extract competencies. Abstracts succinctly capture the core content of papers, thus serving as an effective focus for extracting pertinent competencies without the need to analyze the entire document. This method is computationally efficient and applicable when abstracts are accessible and contain essential information for competency extraction.

Research indicates that abstracts, despite their brevity, retain a high level of informativeness, which is instrumental in extracting key information [Har04].

This approach is particularly advantageous in scenarios requiring large-scale competency extraction, such as compiling profiles from numerous documents or researchers. Processing abstracts achieves high throughput while maintaining accuracy.

Another significant benefit of this method is that it obviates the need for the full text of the document, which might not always be available and may exceed the model's context window if presented in its entirety, necessitating additional steps to segment the document and subsequently integrate the results or omit information to accommodate in *extract from full texts* and *extract from summaries*. This approach avoids the challenges of splitting and merging document sections, thereby streamlining the extraction process.

However, the effectiveness of this method is contingent on the quality of the abstracts. Abstracts that lack comprehensive summaries or omit essential competencies may result in incomplete profiles. Brief or overly generalized abstracts might not fully convey the range of contributions, potentially necessitating supplementary methods to ensure more comprehensive outcomes. Moreover, not all document types feature abstracts, such as in other domains like business or medicine, which limits the applicability of this method.

Extract from Full Texts Competency extraction from the complete texts of documents provides an exhaustive perspective of the subject matter (see fig. 11 lower extraction). This method capitalizes on the full breadth of information available, facilitating the development of an elaborate competency profile.

Given the extensive contexts typical in scientific papers, only one document is processed at a time. From each document, the model extracts a complete competency profile encompassing all relevant competencies identified in the text. These competencies must then be amalgamated into a unified competency profile, which reflects a comprehensive portrayal of all the document's contents. This merging process is vital for ensuring the accuracy and representativeness of the extracted competencies but is also the most challenging aspect of the extraction process. It requires the model to discern the most pertinent

competencies from a set of profiles, selecting and aggregating competencies into a final profile, whereby only a fraction of the competencies may be retained. In this process, the model relies solely on the information within the profiles, which could lead to the omission of vital details that cannot be directly compared to the original document. This challenge is exacerbated when the profiles differ significantly, as the model cannot merely amalgamate multiple competencies into a single logical competency but must judiciously determine which competencies are most relevant without any weighting on them.

4.4 Retrieval Augmented Generation (RAG)

This subsection explores the RAG system (refer to section 2.4 for the background on RAG), detailing its integration with LLMs to enhance the accuracy of competency profiles by utilizing context-specific example pools. The discussion spans the methodology of example retrieval and the strategic use of both manually and automatically generated examples to maintain the balance between precision and operational efficiency. Various applications of RAG, including profile extraction, text summarization, and profile merging, are discussed to demonstrate its utility in minimizing errors and refining output relevance.

4.4.1 Application of RAG for Example Retrieval

The RAG system enhances the capabilities of LLMs by integrating external, context-specific information from an example pool (see fig. 5). This strategy mitigates issues inherent in LLMs, such as the production of hallucinations, irrelevant outputs, or malformed structured responses, by steering the model with indexed, pertinent examples.

In the context of competency extraction, the RAG system ensures the precision and contextual relevance of generated profiles. It supports dynamic example retrieval during the model's generation phase, thereby enhancing the accuracy of competency profiles through exposure to relevant domain-specific examples.

The employed methodology involves the storage of example representations in a vector database through the creation of embeddings that encapsulate semantic significance. Upon request for a competency profile, the model identifies and retrieves the most applicable examples for the given context, utilizing similarity metrics. These examples then serve as supplementary inputs for the LLM, facilitating the generation of more contextually suitable profiles.

Example Scenario: Competency Extraction from Abstracts Referencing fig. 5, the RAG system employs a vector database containing sets of abstracts linked with op-

timal competency profiles, which exemplify complete and optimal competency identification. When initiating a competency extraction task, the system first queries this database using the new abstracts to locate semantically related existing examples. These optimal profiles are then employed as part of the prompt structure, which includes:

- The competency extraction task description
- Semantically related example abstracts from the database
- Simulated LLM response showcasing the optimal profile of those example abstracts
- The abstracts to extract from

This setup guides the LLM in generating a new, accurate competency profile for the input abstracts, thereby producing more relevant and precise outputs than without an appropriate example.

4.4.2 Manually or Automatically Generated Examples

Examples for RAG may be produced either manually by domain experts or automatically by fine-tuned LLMs. The manual approach guarantees precision and expert involvement, whereas the automatic method provides scalability and operational efficiency. It is essential for examples to be representative and conform to the required output format to prevent distortions of profiles. Thus, to achieve a balance between precision and scalability, both manual and automatic generation of examples are employed. Engaging a highly skilled LLM for example generation, followed by manual review and correction by domain experts, is a widely adopted strategy to maintain the quality and relevance of the examples [Zhe+23].

4.4.3 Different Example Types

This thesis employs RAG across various task types, retrieving distinct example sets based on the task at hand. This includes profile extraction (in *extract from summaries*, *extract from abstracts* and *extract from full texts*), text summarization (in *extract from summaries*), and profile merging (in *extract from full texts*). The examples are instrumental in guiding the generation process and in minimizing hallucinations while ensuring the adherence to the desired output format by anchoring the results to factual information.

4.5 Output Parsing and Formatting

This subsection explores the transition from JSON to a custom format in the context of output handling for data interchange. Initially favored for its ubiquity, JSON's limitations in token efficiency and reliability during testing prompted the need for alternatives. A custom format, structured similarly to YAML and thus accessible to LLMs, is introduced as a solution that reduces token consumption and enhances output validity. Comparisons between the two formats, along with specific examples, highlight the benefits and efficiencies gained with the custom format, addressing the challenges previously encountered with JSON.

JSON was the initial format for output handling, recognized for its widespread use in data interchange. Nevertheless, during testing phases, it became apparent that JSON posed several constraints, notably in token consumption and output dependability.

Issues were identified in the consistent generation of valid JSON by some models. Malformed or incomplete JSON complicates concurrent processing as manual corrections are necessary to parse errors. Furthermore, JSON generally utilizes approximately 20% more tokens than alternative formats, which introduces inefficiencies. These issues necessitated the exploration of other options.

The development of a custom format aimed to resolve these difficulties. Structured similarly to YAML, which is accessible to LLMs due to their training, this format offers enhanced flexibility. It has shown a reduction in the incidence of invalid outputs and a decrease in token consumption. Initial testing indicated that the rigid syntax of JSON resulted in a significant portion of outputs being invalid—a problem more pronounced in certain models than others, potentially disqualifying them from further application.

Short Example of the Output Formats:

Listing 2: JSON format example

```
{
 ..."domain": "Expert in developing web communities through competence...",
 ..."competencies": {
 ......"Competence Identification": "Utilizes scientific publications to...",
 ......"Community Building": "Develops web-based scientific communities...",
 .........
 ...}
}
```

Listing 3: Custom format example

```
Domain: Expert in developing web communities through competence analysis.
Competencies:
– Competence Identification: Utilizes scientific publications to map out...
– Community Building: Develops web–based scientific communities by...
...
```

The custom format provides enhanced handling of errors. A parsing system was devised to address potential errors in LLMs outputs, permitting the processing of malformed results with limited intervention. Unlike JSON, minor formatting errors such as unescaped characters in the custom format do not disrupt data management, thus supporting more efficient downstream processing. The custom format also uses fewer tokens, offering marginal improvements in efficiency for extensive processing, where the tokens highlighted in red in the JSON example listing 2 are now redundant.

The transition from JSON to a custom format stemmed from the necessity for improved reliability and efficiency. Although JSON serves well in various scenarios, its use in LLMs outputs uncovered specific challenges that were effectively addressed by the custom format. It is crucial to acknowledge that with the recent advancements in LLM technology, mechanisms are now available that can enforce outputs to adhere to specific formats, such as JSON, which might render the custom format unnecessary (refer to section 2.3.4). However, at the time of authorship, these advancements were not yet implemented.

4.6 Development of the Evaluation Framework

This subsection explores the comprehensive evaluation framework developed to assess competency profiles, highlighting its reliance on expert judgments facilitated through a structured visual tournament. It details the methodology of employing pairwise comparisons to determine the most accurate profile, addressing the absence of pre-existing datasets and the inefficacies of traditional scoring methods. Further, it introduces an automatic evaluation system that complements expert assessments by leveraging larger LLMs, offering a dual approach to enhance accuracy and reliability in profile evaluation especially during development. The subsection also discusses strategies to mitigate biases in the evaluations, ensuring the consistency and fairness of the results.

4.6.1 Principles of Evaluation

The selection of expert evaluation as the primary method for assessing competency profiles stemmed from the lack of pre-existing labeled datasets. The utilization of pairwise comparisons as the central evaluation technique was chosen for its robustness and efficiency over traditional scoring methods in qualitative analysis (see section 2.5.1). The evaluators

engaged in direct comparisons between two profiles, choosing the more accurate of the pair. This procedure helps mitigate the inconsistencies often found when scoring profiles on a linear scale.

The development of a comprehensive evaluation framework was critical to examine a broad spectrum of parameters and extraction methods. The possibility of providing detailed error descriptions for each profile was considered but deferred to future research due to current project limitations. The framework prioritized the identification of the most accurate profile at each evaluation stage, disregarding those not selected.

The design of the evaluation cycle allowed for a duration of approximately five minutes per cycle, enabling expert evaluators to review multiple profiles efficiently while ensuring precision. This method facilitated the examination of a vast array of parameters without placing excessive demands on the evaluators. The profile deemed most accurate at the conclusion of the evaluations was recognized as the superior result from the compared profiles, with emphasis on the parameters of this winning profile.

Employing expert evaluation through pairwise comparisons proved effective in determining the superior system from a variety of extraction methods. The evaluations were conducted as blind tests, where the experts were unaware of the specific method or model employed in the profile extraction.

4.6.2 Structure of Expert Evaluation

The competency extraction expert evaluation employed a visual tournament structure, selected to enable a rapid and thorough assessment process. In this methodology, competency profiles are evaluated pairwise, with experts choosing the superior profile in each round until a single profile emerges as the definitive winner.

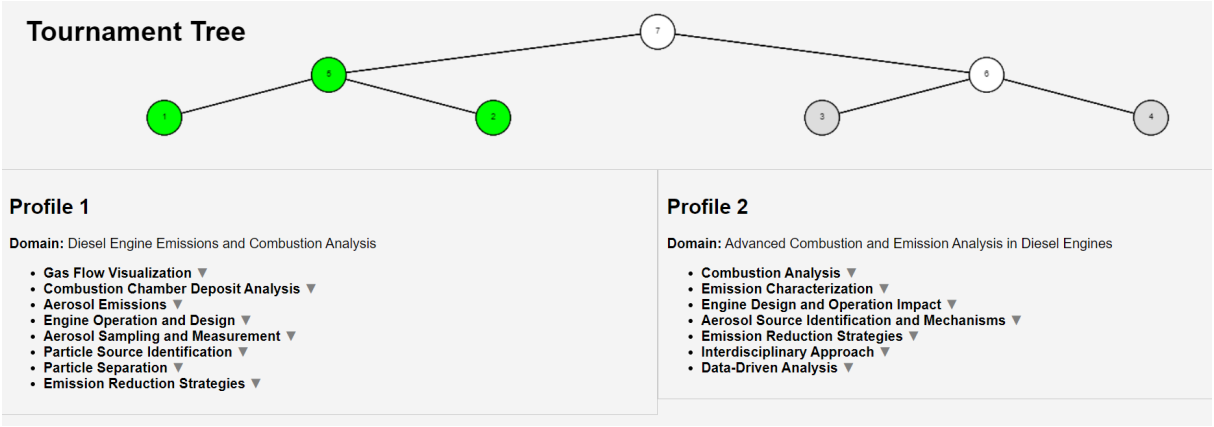

Figure 12: Visual Representation of the Tournament Structure as used in the Expert Evaluation

Direct Preferences In the tournament, the number of games (comparisons) required to determine the winner corresponds to direct preferences. Each game eliminates one profile, resulting in $D(n) = n - 1$ eliminations, thus defining $n - 1$ direct preferences in a tournament comprising n profiles.

Implicit Preferences Implicit preferences arise as a consequence of the tournament's design. A profile's victory in a game at any given round r implies superiority over all competitors previously defeated by the eliminated profile in earlier rounds. The number of such defeated competitors by round r is $2^{r-1} - 1$, with r varying from 1 to $\log_2(n)$.

Calculation of Implicit Preferences The quantity of implicit preferences in any round r is determined by the number of games played during that round and the implicit preferences accrued per game:

$$\text{Implicit Preferences in Round } r = 2^{\log_2(n)-r} \times (2^{r-1} - 1)$$

Aggregating the implicit preferences from all rounds results in the total implicit preferences $I(n)$:

$$I(n) = \sum_{r=1}^{\log_2(n)} 2^{\log_2(n)-r} \times (2^{r-1} - 1)$$

Total Number of Preferences The combined tally of preferences $P(n)$, incorporating both direct and implicit types, is computed as follows:

$$P(n) = D(n) + I(n) = (n - 1) + \sum_{r=1}^{\log_2(n)} 2^{\log_2(n)-r} \times (2^{r-1} - 1)$$

Example Calculation Table 2 illustrates the calculations for direct, implicit, and total preferences for varying n values:

n	$D(n)$	$I(n)$	$P(n)$
2	1	0.0	1.0
4	3	1.0	4.0
8	7	5.0	12.0
16	15	17.0	32.0
32	31	49.0	80.0

Table 2: Example Calculations for Direct, Implicit, and Total Preferences

Evaluation Tournament Example Assuming the winner of the tournament's round 7 in fig. 12 is either profile 3 or 4, then the winner prefers all the green profiles indirectly, as the green profiles were all defeated by the loser of the final game. This method allows for a comprehensive evaluation of all profiles, even those that were not directly compared, providing a larger preference pool for analysis and fine-tuning.

With the 4 profiles in the tournament, we'll recieve 3 direct preferences in match 5, 6 and 7.

Assuming 1 wins (match 5), and 3 wins (match 6), the next match (7) is 1 against 3. The winner of match 7 is the winner of the tournament. Assuming profile 1 wins match 7, then an additional indirect preference for profile 1 over profile 4 can be inferred, as profile 1 won against profile 3, which won against profile 4. This way additional preferences for profiles that were not directly compared can be inferred, under the assumption of transitivity of preferences. Therefore this evaluation tournament results in 3 direct preferences in the evaluated matches and 1 indirect preference from the tournament structure. Refer to section 2.5.3 for more information on the tournament structure.

4.6.3 Automatic Evaluation

An automatic evaluation system has been established to facilitate the assessment of extensive parameter spaces and numerous profiles, minimizing the reliance on expert judgment. This system employs LLMs for evaluating the profiles and offers an assessment that complements that provided by experts. It is particularly advantageous for model development and statistical analysis because it can assess a vast array of profiles efficiently.

The system depends on more extensive LLMs to ensure both accuracy and reliability in evaluations. To evaluate the profiles, models such as GPT-4 or Llama3-70B are utilized because they surpass the capability of models ranging from 3B to 8B parameters, which were examined for profile extraction in this thesis. The system performs comparisons between two profiles derived from the same abstracts to determine which more accurately represents the documents. It achieves this by initially requesting the LLM to conduct a detailed comparison and provide reasoning behind the evaluations of the profiles, followed by a final determination of which profile is superior. This method is adopted because detailed reasoning can yield significant insights into the quality of the profiles and the LLM's comprehension of them, thereby markedly enhancing the quality of LLM evaluations [Koj+23] (see section 2.3.3). The objective is to mimic expert evaluations using LLMs, thus enabling assessments on a significantly larger scale.

The system capitalizes on the premise that evaluation tasks are inherently less complex than generation tasks. Determining the superiority of one profile over another is more straightforward than creating a high-quality profile. By employing more capable LLMs

for this less demanding task, the system can deliver results that are both accurate and consistent with expert evaluations.

4.6.4 Bias Management

Biases in LLM-based evaluations significantly impact the correlation between automated and expert assessments. Notably, position bias may cause discrepancies in competence rankings, with the model favoring profiles that do not align with expert evaluations due to their position in the evaluation sequence. It is crucial to address these biases to ensure the reliability of LLM-based evaluations, which should consistently reflect human assessments. To minimize these biases, steps such as shuffling profiles and maintaining an even distribution of example preferences were implemented.

The strategies applied to manage biases in the evaluation process are detailed in section 2.6.2.

Shuffling Profiles The initial step involves shuffling the order of profiles to prevent any influence of their sequence on the evaluation. This randomization avoids the consistent matching of parameters in a fixed order, which could disadvantage strong profiles in early elimination rounds. By shuffling, the occurrence of two strong profiles competing in the initial round is reduced to a probability of $1/n$, where n is the total number of profiles. Additionally, this approach helps mitigate the LLM's inherent positional bias (refer to section 2.6.2), ensuring that the order of profiles does not exploit this bias.

Even Distribution of Example Preferences To counteract the self-enhancement bias, where the LLM may show a preference for profiles indexed similarly to examples it generated (see section 2.6.2), it is essential to balance the example preferences. By ensuring an equal number of preferences for both profiles in the prompts, the LLM's tendency to favor a particular profile based on its index is neutralized. This strategic distribution of example preferences follows the guidance provided in [Wan+24].

Prompt Engineering Addressing prompt sensitivity and verbosity bias involves meticulous prompt engineering. The designed prompts focus on assessing profile quality by requiring the LLM to provide detailed justifications for its evaluations, thus enhancing response quality (see section 2.3.3). Additionally, the prompts direct the model to verify the format and structure of the profiles, ensuring they do not exceed the maximum of eight competencies and do not exploit verbosity bias by generating more competencies than necessary. Through multiple manual iterations, biases and issues were identified and corrected, ensuring the prompts effectively reduce biases in the evaluation process.

Implementing the above strategies helps diminish the influence of biases on competence extraction, leading to evaluations that are both more accurate and equitable. Ongoing investigations into the extent of biases in model evaluations, through correlation analyses with expert assessments [CL23], are vital. These analyses provide insights into discrepancies between automated systems and human judgment, identifying areas for ongoing improvement (see section 6.2 for results of these analyses).

4.7 Model Selection

This section describes the methodology for selecting appropriate models for expert evaluation in competency extraction tasks. The evaluation aimed at pinpointing models that exhibit optimal performance in accuracy, efficiency, and resource utilization. The selection procedure entailed automated testing of various models and extraction methods under different parameter configurations to evaluate their efficacy and appropriateness for the designated task. The objective was to identify a parameter space sufficiently compact to enable the completion of the expert evaluation within approximately 5 minutes, while accessing the most advantageous parameters for optimal performance.

4.7.1 Conducting the Automatic Evaluation

The selection of models for expert evaluation is based on a comprehensive automatic evaluation process. This process tests multiple models with varying parameter settings to determine the most promising candidates for further refinement and expert analysis. It is assumed that the results of the automatic evaluation correlate significantly with those of the expert evaluation (see section 6.2). The sota models evaluated include:

- **A large language model family developed by french mistral.ai (Mixtral)-8x7B**: A large mixture of experts model (see section 2.3.2) noted for its strong performance and resource efficiency.
- **A large language model family developed by Google (Gemma2)-27B** and **Gemma2-9B**: A sota open weights models developed by Google.
- **Llama3-70B** and **Llama3-8B**: A sota open weights models developed by Meta.
- **Hermes-2-Pro-8B-Instruct**[12]: A sota community trained model, surpassing the base Llama3-8B model in numerous benchmarks.
- **Phi3-14B** and **Phi3-3.8B**: sota open weights models developed by Microsoft.

[12](visited on 10/01/2024) huggingface.co/NousResearch/Hermes-2-Pro-Llama-3-8B

These models, which represent the sota in open weight LLMs, are anticipated to exhibit strong performance in the competency extraction task. The evaluation process aims to pinpoint the models best suited for additional fine-tuning and expert evaluation.

4.7.2 Preliminary Results

Performance Evaluation An extensive pre-evaluation utilizing automatic LLMs evaluation was performed in three trials with roughly 180 profiles in each. The profiles are extracted from different authors and for each author with all combinations of parameters such as extraction method, extraction model and number of examples. Therefore, for each author, each model was tasked with extracting a profile with each of the extraction methods, once with no examples, once with one and once with two examples per extraction. The findings are depicted in fig. 13.

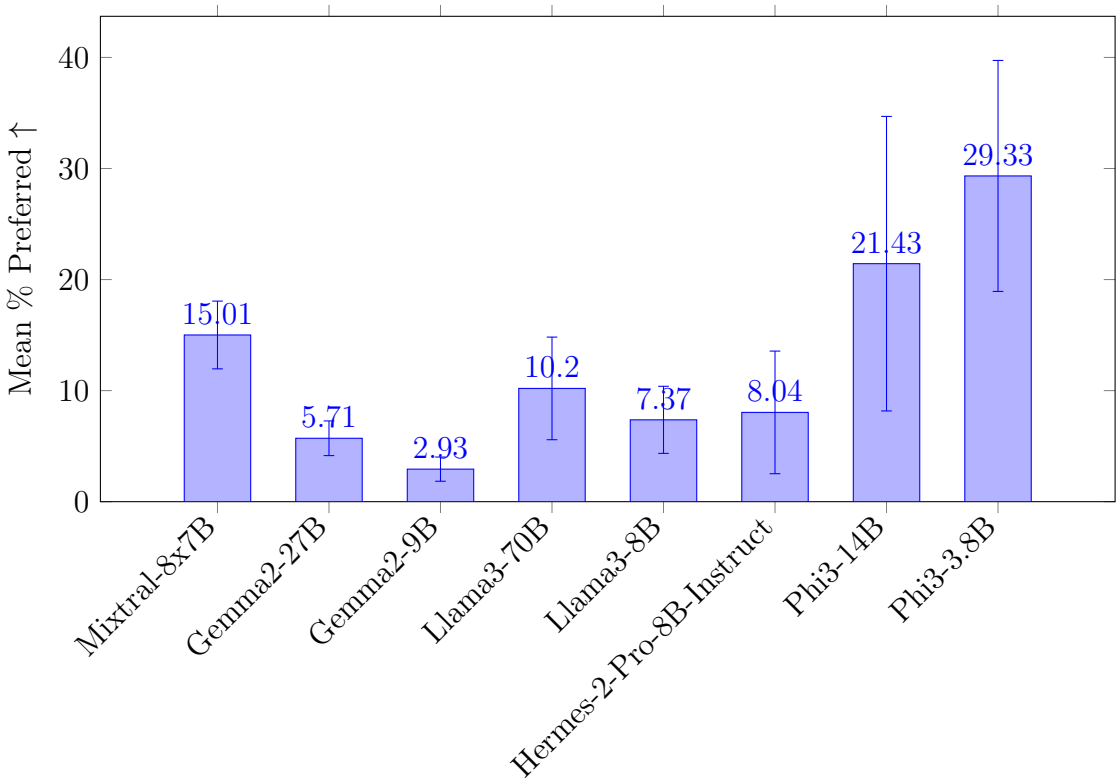

Figure 13: Model performance based on mean percentage preferred (↑: Higher is better) and standard deviation (↓: Lower is better) over three runs.

It was observed that the Hermes-2-Pro-8B-Instruct model marginally outperformed the base Llama3-8B model. The Phi3-3.8B and Phi3-14B models demonstrated superior performance in terms of mean preference rate. The Mixtral-8x7B model was also chosen for further evaluation to investigate the performance of larger models and to diversify the model families utilized in the evaluation. The higher standard deviation of the Phi3-3.8B

and Phi3-14B models indicates more variability and less consistency in the evaluations, which could impact the comparison with the expert evaluations.

The Phi3-3.8B model's robust performance renders it an ideal candidate for further fine-tuning, as the resources required to finetune a 3.8B model are considerably lower than those required for an 8B+ model. Hence, the Phi3-3.8B model is selected for additional fine-tuning.

Competence Extraction Methods Three distinct methods for competence extraction were assessed in the automatic evaluation: *Extract from Abstracts*, *extract from full texts*, and *Extract from Summaries*. The outcomes are shown in fig. 14.

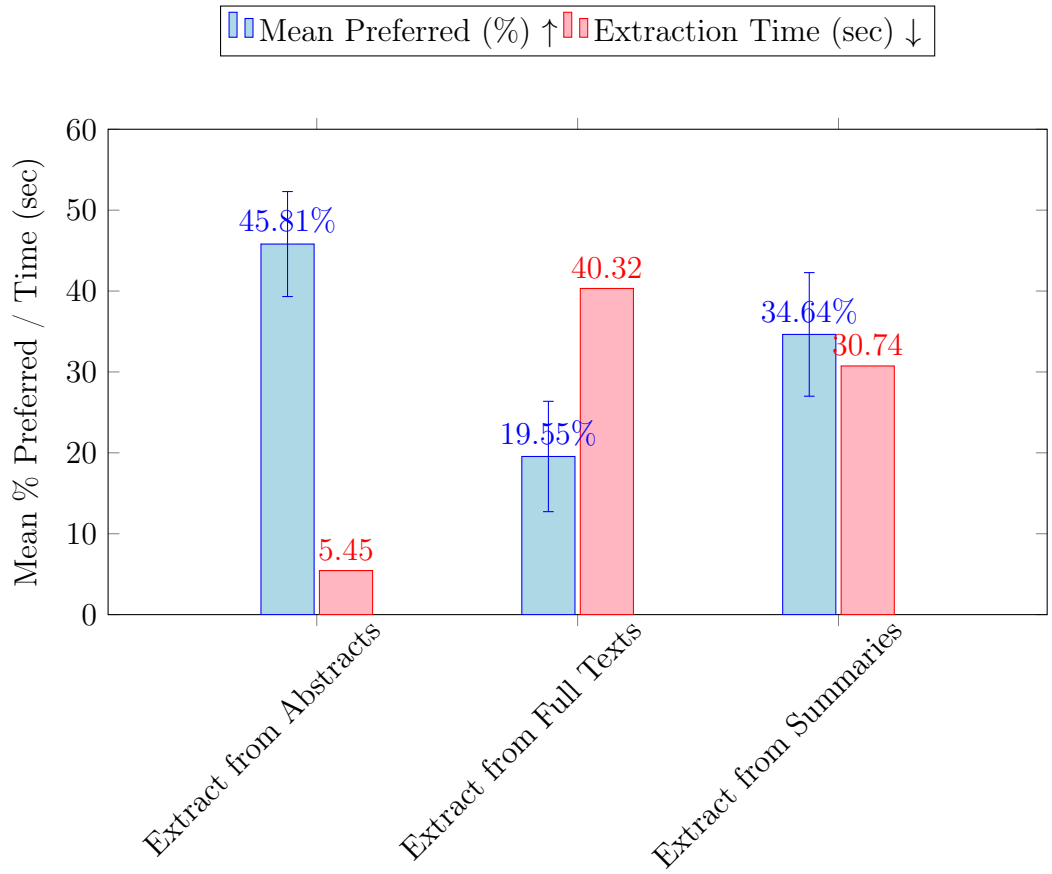

Figure 14: Extraction method performance based on mean percentage preferred by the automatic evaluation and standard deviation, and mean extraction time.

The initial findings indicate that the *Extract from Abstracts* method is the most precise, achieving a mean preference rate of 45.81%, and it is also the quickest. The *extract from full texts* method is the least precise, recording a mean preference rate of 19.55%, and is the slowest. The *Extract from Summaries* method is positioned intermediately with a mean preference rate of 34.64%. Refer to section 6.2 for results on the expert evaluation and the correlation of the evaluation methods.

Impact of Example Count on Performance The inclusion of examples (0, 1, or 2) in the prompt significantly influenced both preference rates and extraction times. The presence of an example in the prompt resulted in a preference rate of 73.17%, markedly higher than the rate of 26.83% for profiles without an example in the prompt.

Example Count	Mean Extraction Time
0 Examples	4.92sec
1 Example	5.45sec
2 Examples	6.36sec

Table 3: Mean extraction time based on the number of examples in the prompt for the *Extract from Abstracts* method with GPT-4o-mini.

The extraction time exhibited a linear increase with the number of examples in the prompt. Introducing one example per prompt provided a balance between performance and resource efficiency. The findings align with prior research that demonstrates the enhancement of LLMs performance across various tasks when examples are included in the prompt [Osw+23] (refer to section 2.3.3 for more on in-context learning).

Output Format No significant performance differences were identified between the JSON and custom formats. The inclination of JSON to generate occasional invalid entries and the associated parsing challenges led to the decision to adopt the custom format. This format proved more resource-efficient, reducing token usage by approximately 20%, and it facilitated easier processing, resulting in more consistent extraction performance. Given the absence of notable performance variances and the goal to avoid doubling the amount profiles that experts have to review, the custom format was selected as the exclusive output format for expert evaluation.

4.7.3 Final Model Selection

The selected models for expert evaluation based on the automatic evaluation results are:

- Phi3-3.8B
- Phi3-14B
- Mixtral-8x7B
- GPT-4o-mini
- Fine-tuned Phi3-3.8B

The fine-tuning process is applied to the Phi3-3.8B model to enhance its effectiveness. The GPT-4o-mini model serves as a benchmark for comparison with larger models accessed via Application Programming Interface (API). The Phi3-3.8B and Phi3-14B models are chosen based on their superior performance in automatic evaluations. The inclusion of the Mixtral-8x7B model allows for the examination of larger models and ensures that the evaluation does not depend solely on one model family. This selection strategy aims to present a varied group of models that demonstrated optimal results in preliminary evaluations, thus identifying the most suitable model for the specific task.

The evaluation of all three extraction methods will utilize a single example per prompt with a uniform output format. This approach stems from a pronounced preference for using examples and the negligible differences between the output formats. Considering the time constraint of approximately five minutes for the expert evaluation, it is essential to limit the number of profiles assessed. With 12 profiles, plus one fine-tuning profile, totaling 13, this number is close to the maximum feasible within the five-minute timeframe. Consequently, the parameter space for further evaluation cannot be expanded in this context.

4.8 Fine-Tuning of LLMs

This subsection explores the fine-tuning of LLMs, specifically the Phi3-3.8B model, for competency extraction. The procedure involved the development of a novel dataset and the implementation of a methodical approach to enhance model performance efficiently. Key components such as the selection of the base model, dataset creation, and iterative fine-tuning strategies are outlined, emphasizing their roles in optimizing the extraction capabilities of LLMs within constrained computational resources.

4.8.1 Selection of the Base Model for Fine-Tuning

Phi3-3.8B was selected for its balance of performance and resource demands. Models with higher capacities, such as 70B+ parameters, surpass the project's limits regarding computational power and time. The Phi3-3.8B model yielded robust results in an automatic evaluation (see section 4.7.2), affirming its appropriateness for subsequent fine-tuning. Techniques like LoRA have been employed to mitigate the resources required for fine-tuning.

Selection of Extraction Method The *Extract-from-Abstracts* method was employed for extraction, as delineated in section 4.7.2, due to its performance and computational efficiency. Abstracts provide a succinct encapsulation of a document's content, facilitating

a less resource-intensive extraction process than alternative methods such as full-text processing or summaries. This approach restricts the number of LLM queries to one per author, precluding iterative summarization and profile merging. Such avoidance is crucial for efficient large dataset generation and the fine-tuning process, enabling both to proceed effectively on consumer-grade hardware.

4.8.2 Creation of a Synthetic Dataset

A synthetic dataset was developed for this thesis due to the absence of pre-existing datasets suitable for the specialized task of competence extraction. The DPO fine-tuning method necessitates a dataset formatted with a *prompt*, a *chosen* (preferred) response, and a *rejected* response. This format is precisely what the tournament structure with automatic evaluation is designed to generate. For this purpose, 5 abstracts were selected from an English-speaking author from the US, and with the same prompt, 8 different profiles were generated. These profiles are devised by the model to be fine-tuned using top-k sampling (see section 2.3.4). The number 8 is determined as a hyperparameter, which might be increased to generate more preferences through the tournament ranking. Nonetheless, memory constraints limited the profile count to 8, as this number represents the maximum that could be generated without incurring GPU memory overflow. The profiles are required to be diverse enough to encapsulate different preferences and quality extractions. The success of the DPO fine-tuning process hinges on the diversity of the profiles, ensuring that the model learns to consistently generate the preferred profile.

An expert LLM (Llama3-70B) was employed to evaluate the automatic tournament, generating all preferences. The efficacy of the expert model is paramount for the dataset's quality, as it consistently identifies the superior profile. The presumption is that a large, robust expert model will yield the best preferences, vital for directing the fine-tuned model's training. The Llama3-70B model was selected due to its status as the most substantial and capable model available for self-hosting. Access to larger models through paid APIs exists, yet this was not feasible within the financial constraints of this project (see section 6.2 for an evaluation of a more potent expert model). From the tournament, 12 preferences ($P(8) = D(8) + I(8) = 7 + 5 = 12$) were derived, which adhere to the DPO format (see section 4.6.2 for a derivation of the formulas).

It is critical to acknowledge that the fine-tuning process was confined to papers from the US to prevent the contamination of the training data with evaluation samples from KIT professors, from which the expert evaluation was done. This ensures, that the model is not trained on the same data it is evaluated on, which would lead to contamination and biased results.

4.8.3 Fine-Tuning Procedure

The fine-tuning of the Phi3-3.8B model utilized the DPO strategy. This strategy was implemented to optimize the system based on the direct preference data collected from the proprietary dataset, as detailed in section 2.7.3. The DPO strategy is particularly apt for scenarios requiring preference optimization between two outputs, such as in the competency extraction task. The selection of the DPO strategy over alternatives such as RLHF and PPO was based on its superior simplicity and reduction of hyperparameter tuning requirements. The DPO strategy is more straightforward to implement and requires fewer hyperparameters, making it more suitable for the project's constraints. The DPO strategy is also more computationally efficient than RLHF and PPO, which is crucial for training on consumer-grade hardware.

The method involved using the model targeted for fine-tuning to create a dataset, as outlined in section 4.8.2. This dataset creation employed an expert LLM to generate high-quality, representative preferences. Subsequently, this dataset facilitated the fine-tuning of the model using the DPO method. The QLoRA technique was applied during fine-tuning, which supports parameter-efficient adjustments. This technique substantially reduces the resources required for training by focusing on small adapters rather than the entire model, thereby enabling feasible training on consumer-grade hardware. The application of QLoRA led to good results, making the training approach effective. Further details on the implementation can be found in section 5.6, and the outcomes of the fine-tuning process are discussed in section 6.3.2.

4.8.4 Iterative Approach

The concept involved either creating a large initial dataset from the base model to fine-tune in a single iteration or conducting the entire fine-tuning process iteratively. The iterative fine-tuning involves enhancing a model by refining it through multiple training cycles on progressively better datasets. Each cycle yields a more refined model, which then produces higher quality synthetic datasets for subsequent fine-tuning rounds (refer to section 4.8.2) (see fig. 15). This approach facilitates consistent improvements in model performance without the need for an excessively large dataset or extensive computational resources. Moreover, it tends to lead to quicker convergence and superior outcomes, as data quality significantly influences the success of fine-tuning [Hsi+23].

Model Training The iterative process commences with an initial base model (Phi3-3.8B from huggingface) trained on a synthetic dataset generated by the same base LLM. Post the first fine-tuning round, this refined model typically generates higher quality profiles than the base model, thereby improving the quality of both the selected and rejected

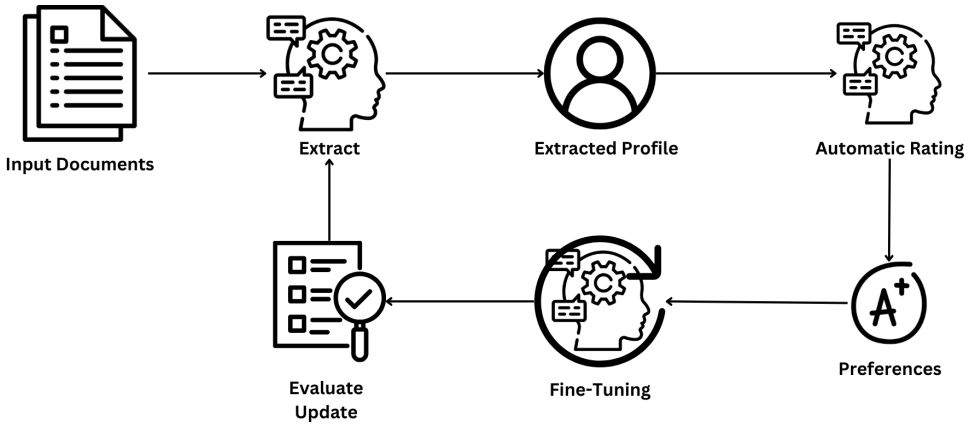

Figure 15: Data Flow of the Iterative Fine-Tuning Process

profiles in the dataset. Each subsequent iteration aims to enhance profile generation beyond the quality of profiles rejected in prior rounds. This process repeats until the model's output meets the necessary quality standards for the intended application. The iterative method builds upon previous versions, enabling quicker convergence and reducing computational demands.

Iteration Evaluation To verify ongoing improvements, the performance of the model is assessed after each iteration against both the previous version and the base model. These evaluations occur after each cycle to ensure progress before proceeding to the next round. The expert LLM (Llama3-70B) assesses the generated profiles, which consistently originate from the same authors not included in the training dataset. It is critical to use the same authors for evaluations to maintain comparability across iterations.

4.9 General System Design

This section describes the architecture of the developed general system that operates independently of domain constraints and adapts to various types of documents. The adaptability of the system across various domains leverages domain-specific examples. For instance, when applied to the corporate sector, the examples utilized are derived from patents or annual reports pertinent to that sector. The system incorporates multiple extraction methodologies to accommodate various input types, selecting the optimal extraction method for each type of document to ensure maximum performance and accuracy.

The extraction process, as depicted in fig. 16, initiates by accepting as inputs abstracts and full-text documents. To generate one succinct competency profile based on all the input data, the system starts by processing abstracts in groups of five to construct $\lceil \frac{\text{num abstracts}}{5} \rceil$ competency profiles through the *extract from abstracts* method. Concurrently, it extracts

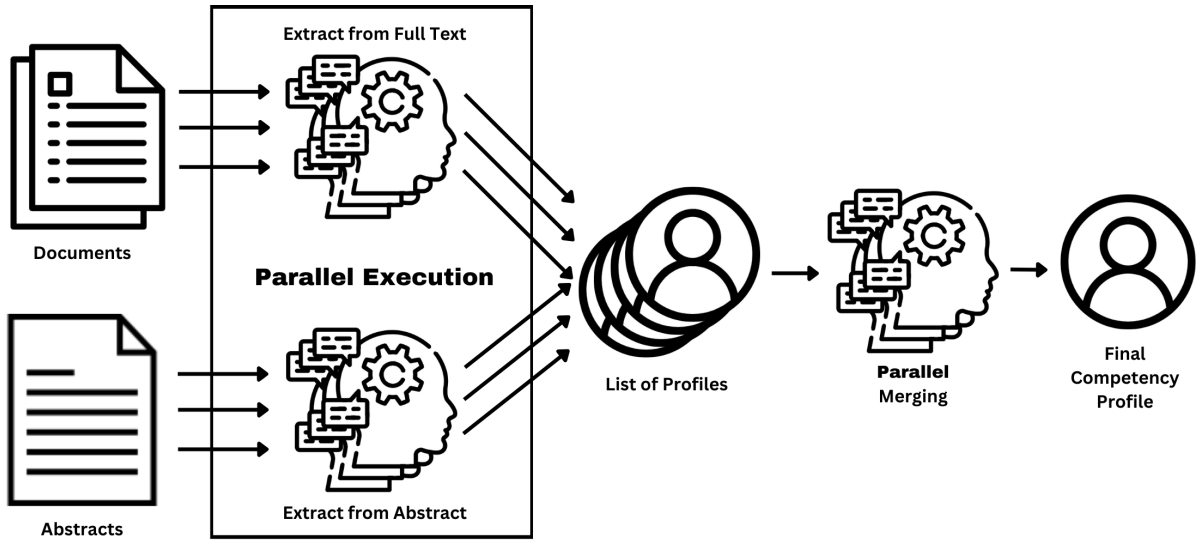

Figure 16: Data flow during extraction through the general system

individual competency profiles from full texts using the *extract from full text* method. These ($\lceil \frac{\text{num abstracts}}{5} \rceil$ + num full texts) profiles are combined in a recursive manner, in batches of five, employing the merge algorithm from the *extract from full text* method. The merging strategy ensures the final competency profile embodies all crucial data from both abstracts and full texts.

This methodology facilitates parallel processing of extraction tasks, enhancing efficiency for extensive datasets. It is premised on the capability to process an indeterminate number of documents concurrently, as the extraction operation is not contingent on document quantity but rather on the computational resources available. To achieve optimal performance, the system necessitates a minimum of: ($\frac{\text{num abstracts}}{5}$ + num full texts), accessible GPUs to process all documents concurrently. The duration to finalize the merging operation is computed using the formula:

$$T_{\text{complete merge}} = T_{\text{merge one}} \times \log_2(\lceil \frac{\text{num abstracts}}{5} \rceil + \text{num full texts})$$

whereas the overall extraction duration T_{extract} is established by:

$$T_{\text{extract}} = \max(T_{\text{abstract extraction}}, T_{\text{full text extraction}}) + T_{\text{complete merge}}$$

Under optimal circumstances, this results in an extraction duration of less than a minute, even for datasets nearing 100 documents.

The modular and flexible design of the system renders it appropriate for a wide array of applications. By incorporating new RAG examples pertinent to the desired domain, the system can quickly adjust to varied fields, such as academic research, business profiling, or medical competency extraction. This configuration affirms the system's generalizability

to multiple use cases without substantial modifications, highlighting its robustness and scalability.

4.10 Domain-Agnostic Examination

The developed competency extraction system has undergone testing across various academic domains, including humanities, computer science, and physics. It demonstrates potential for application in sectors like medicine and business, where it could facilitate the extraction of competency profiles from doctors or employees to enhance task allocation and collaboration efforts.

General Applicability Across Domains By altering the input data and examples used within the prompts, the system demonstrates the capability to perform competency extraction efficiently across multiple domains. The incorporation of examples in RAG increases the model's output relevance by retrieving appropriate data points from a vector database. For domain-specific tasks, a meticulously curated set of examples is crucial to ensure that the competencies extracted are pertinent to the domain's specific requirements.

The application of LLMs, enhanced by retrieval mechanisms, allows adaptation to the specific demands of different domains without the necessity for additional fine-tuning [Lew+21; Ram+23]. Studies concerning retrieval-augmented models illustrate significant performance improvements in scenarios that require specialized domain knowledge [Lew+21; Ram+23].

4.10.1 Application to Corporate Data

The system designed for extracting competency profiles from scientific documents was assessed for its application in the corporate sector. The evaluation examined the system's capability to adapt by extracting competency profiles from corporate documents such as patents and annual reports, and then comparing these profiles with those manually created for companies.

A dataset from CAS consisting of patents and annual reports from various corporations was utilized. These documents, abundant in technical and business-specific details, were employed to evaluate the precision and pertinence of the competency profiles extracted. Summaries provided by the companies outlining their principal activities were used as benchmarks for comparison.

System Adjustments for Corporate Data Despite the system's architecture remaining as specified in section 4.9, modifications were introduced to the input data and RAG examples to suit the corporate environment. The system utilized patents and annual reports to extract competency profiles, following the method outlined in section 4.9. The examples in the prompts were customized to emphasize competencies pertinent to business operations and technological innovations.

4.10.2 Evaluation of Results

An automated LLM approach served as the evaluation framework to gauge the alignment between the profiles generated by the system and the short profiles provided by companies. The primary metric involved assessing the correspondence of key competencies with the companies' descriptions and the overall structure of the profile. Scores were allocated based on the extent to which the extracted competencies corresponded with the terminology, scope, and content of the companies' summaries. Refer to section 6.4.1 for the outcomes of this evaluation.

5 Development

This chapter addresses the technical construction of the competency extraction system. It describes the architecture and components of the system, from backend infrastructure to LLMs integration. The processes for developing extraction methods—deriving competencies from abstracts, complete texts, and summaries—are elucidated. It further explores the challenges posed by diverse document types and formats and the iterative process of optimizing LLMs for superior performance.

5.1 Data Acquisition and Processing

This subsection details the structured approach to data acquisition and text extraction utilized in this thesis.

5.1.1 Data Acquisition via the OpenAlex API

The data acquisition system employed the API provided by OpenAlex, which contains metadata for over 250 million scientific publications and 90 million disambiguated authors. OpenAlex offers precise filtering options, such as restricting results by language (f.e. English), author-specific criteria, and ensuring accessibility based on the number of publications (f.e. with a minimum threshold of five). These capabilities enabled the targeted collection of relevant publications necessary for the research.

For efficient interaction with the API, the Python binding "PyAlex" was utilized, facilitating programmatic queries that adhered to the thesis's filtering criteria. This automation streamlined the data collection process by minimizing manual intervention and enhancing reproducibility, critical for the creation of a large fine-tuning dataset.

Despite challenges in data standardization and occasional manual data cleaning, the system facilitated rapid scaling of data acquisition processes and provided a robust foundation for subsequent analyses.

5.1.2 Extraction of Full Texts with PyPDF

The extraction of full-text documents is crucial for the analysis that follows in the methods of extraction. PyPDF, a Python library designed for handling PDF documents, facilitated the extraction of full texts from the collected scientific publications. For these scientific papers, a subsequent postprocessing step was necessary to eliminate the references and appendix sections, thus ensuring extraction of solely the main content.

Challenges in Text Extraction A prevalent issue was the inconsistent quality of the extracted text, which typically manifested as incomplete or poorly structured text. This inconsistency potentially degraded the accuracy of competency extraction. The extracted text often contained superfluous artifacts such as metadata, headers, and footers; additional preprocessing steps to cleanse and standardize the data would be beneficial.

Moreover, PDFs, especially those containing scientific content, pose substantial challenges due to their intricate formatting, which includes multi-column layouts, figures, tables, and mathematical expressions. Such elements frequently lead to erroneous interpretations by PyPDF, resulting in garbled or incomplete text extractions.

Future Work and Improvements The less-than-optimal quality of text extraction impacts the accuracy and dependability of the competency profiles extracted. Future initiatives should investigate alternative tools or develop bespoke solutions that are better suited to the formats of scientific documents. The application of advanced NLP techniques and LLM-based approaches could enhance the quality of extractions by more effectively managing complex document structures.

5.2 Development of the System

This subsection elaborates on the implementation specifics of the competency extraction system, which was developed using Python. The entire system is available on GitHub[13].

5.2.1 Completely Self-Developed System

This competence extraction system was meticulously constructed from scratch, avoiding the use of pre-existing frameworks to tailor it specifically for the intended tasks. Emphasis was placed on integrating caching mechanisms that minimize the number of LLMs queries, and implementing both asynchronous and batch processing of LLM requests.

The development included the creation of automated tools for analyzing evaluation results, tracking progress, visualizing findings, and supervising LLMs interactions. For debugging purposes, HTML-based tools were developed, providing comprehensive analyses of LLMs dialogues and outputs. These tools pinpointed several issues such as formatting discrepancies and the "4k token problem", which led to performance deterioration when the token count exceeded 4096 (refer to section 6.3.3 for additional details).

Moreover, tools were devised for conducting tournament-style evaluations to graphically display preferences determined by automated evaluation models, assess the LLMs's per-

[13]https://github.com/BertilBraun/Master-Thesis

formance, and scrutinize the evaluation timings along with the parsing of inputs and outputs.

The debugging tools uncovered significant challenges, such as incorrect inputs, misformatted outputs, and token capacity constraints. The strategic decision to develop bespoke tools was crucial in identifying and resolving these issues promptly, thereby enhancing the system's reliability. These tools not only facilitated feedback during the development phase but also improved the monitoring and optimization capabilities of the competence extraction system. The early development and integration of these tools were instrumental in bolstering the overall stability of the system.

5.2.2 Backend Integration and LLM Calls

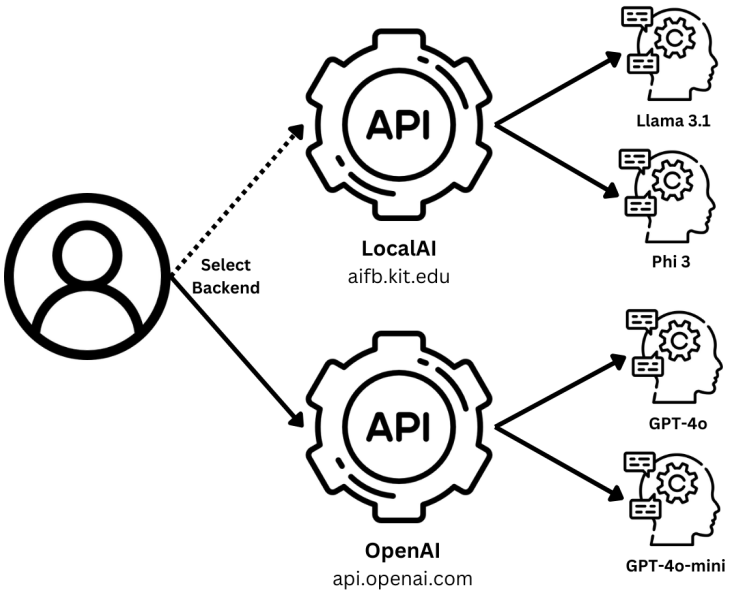

Figure 17: System Architecture: Backend Integration for LLM Calls

The backend system is configured to facilitate smooth integration with both the OpenAI API and a local LLM server designated as Local-AI via a REST API. This configuration allows the system to alternate between different LLMs and providers by modifying the base URL and model ID, thereby minimizing the need for extensive code modifications. Such capability proves advantageous in scenarios where dependency on cloud services raises privacy issues, permitting the adoption of local models in compliance with strict data protection regulations.

5.2.3 Strong Typing, Extendability, and Code Reusability

The development of the system emphasized strong typing to guarantee that the codebase is highly extendable and easy to maintain. Type annotations were consistently imple-

mented to enhance the robustness of the system by utilizing generics for increasing code reusability while ensuring strict type safety. This approach conforms with well-established software engineering principles, where the application of generics facilitates type-safe code reusability without logic duplication. Generics enabled the creation of reusable modules that can be customized for specific requirements without sacrificing the flexibility or extendability of the system. Such flexibility in typing ensures that subsequent developers can extend the system with minimal need for major refactoring. Moreover, strong typing aids in early error detection, diminishing runtime errors and boosting the overall reliability and maintainability of the system.

Impact on Code Reusability and Successor Development By integrating strong typing and generics throughout the system, subsequent developers are equipped to construct new components or enhance existing ones with reduced effort. Type annotations further act as in-code documentation, aiding in the comprehension and navigation of unfamiliar code. This practice is consistent with the best practices in modular and reusable code design.

The emphasis on clear type definitions and the use of generics are crucial for ensuring effective reuse of the system, particularly by future developers.

5.3 Output-Format-Investigation and Implementation

This section addresses the challenges and methodologies involved in organizing and processing outputs produced by the LLMs for competency extraction tasks. The focus is on determining appropriate formats that enhance performance while ensuring high parsing accuracy.

In the evaluation phase, the custom format was marginally favored in preliminary tests due to its reduced processing demands and its capacity to parse with fewer errors, although the accuracy of the evaluations remained comparable between both formats, indicating that the informational content of the formats was largely identical.

5.3.1 Methods for Enforcing the Output Format

Various strategies have been employed to maintain a consistent output format from LLMs, as detailed in section 2.3.4. A notable strategy involves the use of LMQL, which permits users to define output constraints directly within prompts. This technique facilitates enhanced control over the output structure, such as JSON or other custom formats, essential for applications that necessitate structured output parsing. LMQL functions by directing the generation of the model's output, adjusting logits prior to sampling to

ensure adherence to the specified format.

Investigated Methods for Formatting The implementation of LMQL, however, results in a significant increase in performance overhead, with observed reductions ranging from 200% to 300%. It is speculated that this decline is linked to memory management and cache issues on the GPU, though the exact reasons are not definitively known. This reduction in system efficiency has rendered LMQL unsuitable for the intended applications. Moreover, the application of LMQL precluded the possibility of utilizing API driven LLM interactions, which were necessary for efficient communication between the backend LLM server and model.

Robustness of Prompts and Hyperparameter Considerations Given these constraints, it was decided to leverage the native formatting capabilities of the LLMs, depending on their standard output structures instead of imposing stringent formats. The custom format devised for this initiative managed to accommodate most discrepancies in output. Although this method did not impose the strictures of LMQL, it enabled quicker processing times and maintained high output quality. Future will be able to build on the structured output enforcing features introduced by OpenAI in August 2024 (see section 2.3.4).

5.3.2 Robustness of Prompts and Hyperparameter Considerations

Extensive experimentation was conducted on the extraction process to ascertain the robustness of the prompts applied within the LLMs. These experiments were crucial in formulating prompts that consistently produced reliable outputs. The robustness of these prompts, when coupled with the meticulous selection of output formats, was essential for the high quality of the extraction results. The development of the prompts was refined through numerous iterations, culminating in stable extraction profiles that strictly conformed to the designated output formats.

In the extraction process, the hyperparameters were meticulously chosen to enhance the performance of the LLMs. The selection of these hyperparameters drew upon both literature and empirical insights from experts, aiming for optimal accuracy and performance. It was noted that the chosen hyperparameters markedly impacted the quality of successive training sessions, thereby influencing the quality of the extracted profiles. The adjustment of hyperparameters presents potential avenues for future work, as further enhancements could yield better extraction outcomes. Techniques such as Bayesian optimization or grid search might be employed to refine the hyperparameters for superior performance [VM21].

5.4 Retrieval Augmented Generation (RAG)

The implementation of RAG aims to enhance extraction accuracy by integrating a vector database of examples. RAG utilizes retrieval-based methods alongside generative LLM systems to improve both the quality and relevance of the outputs generated (refer to section 2.4 for additional information).

ChromaDB served as the vector database for storing and retrieving examples, playing a pivotal role in enhancing the performance of the RAG system. ChromaDB enabled efficient storage and retrieval of context-sensitive data during model inference, which was essential for including relevant examples in the model's prompts. This integration was critical for enhancing extraction quality, especially with large volumes of data processed.

The retrieval process utilized Cosine-Similarity as the primary distance metric to measure the proximity between stored examples and queries. Although this method was generally effective, it occasionally presented challenges concerning the quality of retrieved examples. These examples were not always the ones that would have been manually selected as the most relevant. Considering alternative metrics or retrieval methods and embedding models, such as Euclidean distance[14] and OpenAI's text-embedding-3-large as a stronger embedding model, could potentially yield superior results.

Postprocessing ensured that retrieved examples maintained a minimum distance from one another to promote diversity and prevent excessive similarity to the original sample. This step was crucial for eliminating redundancy and sustaining the variety and accuracy of the generated outputs, and for ensuring that the answer is not included in the prompt as an example. For expert evaluation, a single example sufficed, but broader applications required maintaining diversity among retrieved examples.

5.5 Development of the Evaluation Framework

The evaluation framework supports assessments of competency profiles by both experts and automated systems. This section describes the technical development of the framework, emphasizing its design, implementation, and importance in the evaluation process.

5.5.1 Development of the Visual Tournament

The *Visual Tournament* system was constructed using HTML and JavaScript to craft an intuitive and efficient interface for the expert evaluation of competency profiles, as demonstrated in fig. 12. This system presents pairs of competency profiles to the expert, who

[14] medium.com/@stepkurniawan/comparing-similarity-searches-distance-metrics-in-vector-stores-rag-model-f0b3f7532d6f

then selects the more appropriate profile through successive comparisons in a tournament format. The process is repeated until a definitive winner emerges.

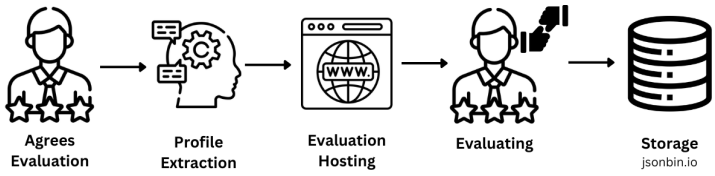

Figure 18: Flow of the Expert Evaluation Process

The evaluation process was initiated by contacting a list of experts via email, inviting them to participate in the evaluation. Each expert who has agreed to participate was provided a tailor-made evaluation, with the system generating a distinct HTML page for each participant. These pages were hosted on a server, with personalized email links distributed to each expert to facilitate access and execution of the tournament-based evaluation. This method enhanced the efficiency of the evaluation process and promoted expert engagement.

The evaluations were recorded in JSON format on an external server[15], facilitating convenient access for subsequent processing and analysis. The utilization of a static web page reduced system complexity and enhanced portability, eliminating the need for dynamic server-side processing.

The system effectively transformed expert evaluations into measurable outcomes. The tournament format lessened cognitive burden by offering only two choices at any given time, thereby streamlining the decision-making process. By graphically illustrating the comparison, the system ensured that evaluations were both dependable and replicable.

Potential challenges involve biases in the selection process due to subjective preferences of the experts. These could be addressed by broadening the diversity of the evaluator pool in future implementations.

5.5.2 Integration of the Automatic Evaluation System

Overview The automatic evaluation system was implemented to enhance the assessment process. It functions on identical principles to the visual tournament of processing the profiles in pairwise preference matches to determine preferences but is facilitated by a LLM instead of a human expert. Large LLMs, specifically the Mixtral-8x7B and Llama3-70B models, are utilized as automatic expert evaluators for assessing competency profiles. The choice of larger LLMs is predicated on their superior performance outcomes. The automatic evaluation system is employed during pre-evaluation (refer to section 4.7), for

[15]jsonbin.io

creating the fine-tuning dataset (refer to section 4.8) and alongside expert evaluations to ascertain the correlation between these two assessment approaches. Furthermore, the system generates preferences for the fine-tuning dataset (refer to section 4.8.2).

The automatic evaluation system's capability to conduct rapid pre-evaluations with minimal human involvement highlights its importance, especially in extensive model evaluations involving hundreds of samples which is not possible on demand with human experts.

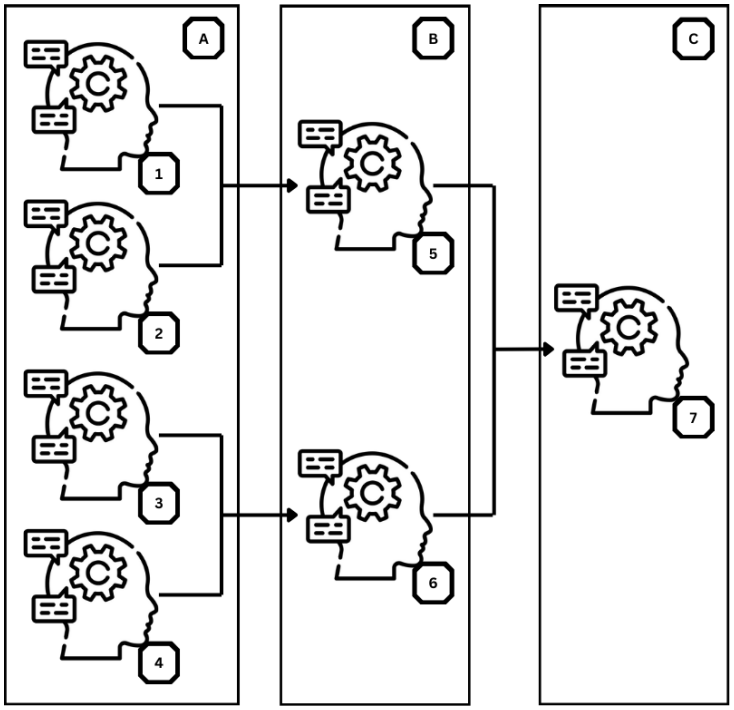

Figure 19: System Architecture: Parallelized LLM Calls for Tournament Evaluation

Performance Optimization A complete tournament can be evaluated in less than two minutes of computational time through sequential LLM calls. To enhance this process, the parallelization of LLM calls at each tournament level diminishes the total computation time from:

$$T_{totalunoptimized} = (n-1) \cdot T_{compare}$$

to:

$$T_{totaloptimized} = O(\log_2(n)) \cdot T_{compare}$$

by evaluating all pairs at each level simultaneously, thus only necessitating $\log_2(n)$ levels of comparisons (A, B, and C in fig. 19) as opposed to the $n-1$ comparisons in sequential evaluation (1-7 in fig. 19). This advancement significantly boosts performance,

enabling quicker feedback loops and the assessment of substantially larger parameter spaces, provided sufficient computational resources are available, requiring $\frac{n}{2}$ GPUs for the parallelization of the initial level and reduced portions for subsequent levels.

5.6 Fine-Tuning of the LLMs

The fine-tuning of LLMs occurred at the Baden-Württemberg University Cluster 2.0 (BW-Uni-Cluster)[16], which provided access to NVIDIA A100 80GB Tensor Core GPUs (A100s) for training optimization and efficient resource utilization. Emphasis was on maximizing the use of available resources to reduce both training and generation times. Each compute node within the cluster was equipped with up to four A100s, enabling task parallelization that significantly shortened the time necessary for synthetic dataset creation. This streamlined process facilitated the production of datasets in hours instead of days, expediting the fine-tuning procedure.

5.6.1 Parallelization and Multithreading

Python's multithreading capabilities were employed to execute multiple tasks simultaneously. The workload was evenly distributed across GPUs using intelligent batch processing strategies to optimize throughput. Through the application of multiprocessing techniques, each GPU was fully utilized, enhancing the efficiency of all four A100s in a node. Enhanced batch parallelization in the LLM inference process contributed to a reduction in the time required for dataset generation.

5.6.2 Use of Transformers and TRL

The fine-tuning of the LLM utilized the Transformers[17] library from Hugging Face[18] and the Transformers Reinforcement Learning (TRL)[19] framework. These resources provided straightforward and adaptable capabilities for implementing the DPO fine-tuning strategy.

Transformers and TRL Framework The Transformers library by Hugging Face offers a comprehensive collection of pre-trained models and tools that expedite the fine-tuning process for various NLP tasks. Its cohesive API supports efficient model utilization among researchers and practitioners. The TRL framework supports RL methodologies

[16](visited on 10/01/2024) wiki.bwhpc.de/e/BwUniCluster2.0
[17](visited on 10/01/2024) huggingface.co/docs/transformers
[18](visited on 10/01/2024) huggingface.co
[19](visited on 10/01/2024) huggingface.co/docs/trl

such as PPO and the pertinent DPO for fine-tuning based on human and synthetic feedback.

Controlled Training Process The fine-tuning process was executed in incremental, controlled stages to confirm accuracy before advancing. This step-wise method enabled meticulous oversight and modification, minimizing the likelihood of accumulating errors. This technique is advantageous in environments with limited resources, such as the BW-Uni-Cluster. Enhancing GPU efficiency through concurrent processing and multithreading approaches streamlined the fine-tuning process, diminishing the duration necessary for dataset production and model adjustments.

5.6.3 Challenge of Job Queues

The prolonged wait times within the job queues posed a significant challenge during the fine-tuning of the LLMs. Managed by the BW-Uni-Cluster, these queue management systems are integral to handling the workload. Extensive delays in job processing are a well-known issue in high-performance computing environments, where job execution is frequently postponed pending the availability of adequate resources [Par19].

To counter the challenges of extended delays, the fine-tuning was segmented into smaller, manageable jobs. This segmented approach facilitated more accurate monitoring of system progress and diminished the likelihood of disruptions causing failures.

Iterative Approach to Monitoring Progress This iterative method allowed for preemptive adjustments prior to initiating longer tasks, thereby enhancing job robustness. Despite introducing additional scheduling challenges, this strategy was successful in lowering the incidence of failures from unexpected complications.

Performance Trade-offs and Scheduling Complexity The strategic segmentation of larger tasks into smaller subtasks was effective in curtailing wait times, thus promoting fairness and enhancing system performance.

Job Handling Method	Performance Impact
Single large jobs	Higher risk of failure
Smaller iterative jobs	Reduced risk, increased scheduling complexity

Table 4: Performance trade-offs between different job scheduling methods.

Despite the inherent trade-offs, the division of larger jobs into smaller subjobs has enhanced overall system resilience and adaptability within a resource-limited setting.

5.7 Domain-Agnostic Investigation

This subsection briefly details additional implementation specifics for the domain-agnostic investigation (see section 4.10).

Data Preparation and Cleaning Annual reports contain extensive tabular data irrelevant to the extraction of competencies. These tables were selectively excluded to refine the dataset, thus enhancing both efficiency and the quality of the output.

Otherwise, data from patents and annual reports were incorporated into the competence extraction system with minimal preprocessing.

The domain-agnostic nature of this investigation underscores the system's capability to process data from diverse sectors without significant modifications.

6 Evaluation

The chapter on evaluation details findings from expert assessments, contrasting various competency extraction methods and their effectiveness. Quantitative results such as enhancements in preference scores, along with qualitative feedback from domain specialists, are discussed to evaluate the accuracy and dependability of the competency profiles. The benefits of DPO in refining the extraction process are examined. Additionally, the system's efficiency and scalability are assessed.

6.1 Evaluation of the Extraction Methods

This section presents the outcomes of the expert assessment. For expert recruitment, a request for volunteers was made at a university meeting, receiving 18 responses from individuals ranging from students to professors across disciplines such as material sciences, physics, chemistry, and computer science. Unfortunately, some volunteers did not fulfill the criterion of having published papers themselves, necessary for the extraction process, since they need to be able to reliably judge the amalgamated competencies represented in all papers, thus were excluded from the evaluation. From the others, not all responded within a month that submissions were awaited. Ultimately, nine evaluations were obtained and utilized for assessing the extraction methods. The evaluation framework is delineated in section 4.6.1.

The outcomes indicate a noticeable correlation between the automatic and expert evaluations. The methods *extract from abstracts* and *extract from summaries* exhibit closely aligned preference rates. The *extract from full texts* method registers a higher preference in the expert evaluation, comparable to the *extract from summaries* method, suggesting both methods deliver comparable results but fall behind the *extract from abstracts* method substantially. The potential reason is cumulative errors beginning with problematic PDF extraction as noted in section 5.1.2, which adversely affects the summaries and initial profiles for full text extraction, thereby diminishing the quality at each step. The fine-tuning method, despite registering the lowest preference rate in the expert evaluation, necessitates adjustment since the other extraction methods utilized four different models, whereas the fine-tuning was conducted solely with the fine-tuned model. Theoretically, the preference rate for fine-tuning should approximate four times its current rate, suggesting a rate of 36.36% in the expert evaluation, positioning it as the second most effective method after *extract from abstracts*, which benefits from significantly larger and more robust base models (refer to section 6.3.2 for further details). The notable discrepancy in evaluation between the automatic and expert assessments of the fine-tuning method, which was specifically trained to optimize the preference rate of the evaluation

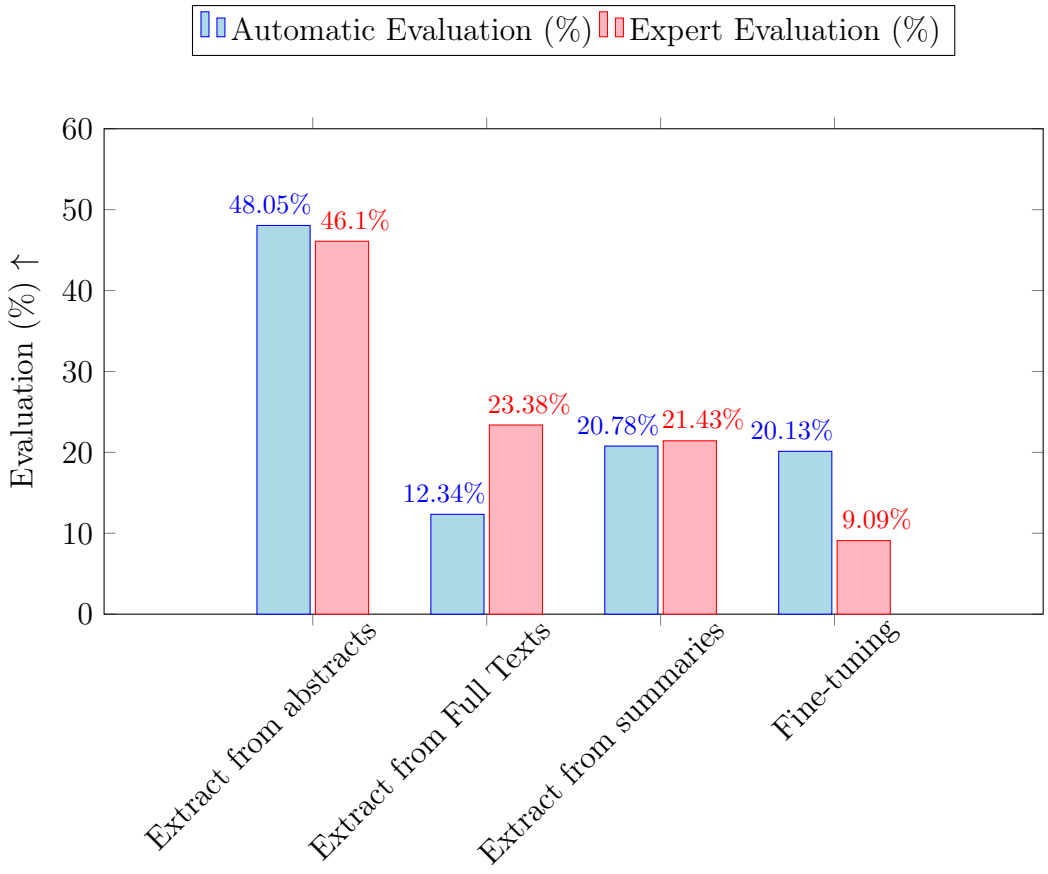

Figure 20: Preference rates of the different extraction methods in the automatic and expert evaluations (↑: Higher is better).

model used for automatic assessments, suggests limited generalization of the fine-tuned model, so that the fine-tuned model overfits to the preferences of the evaluation model, which no longer represents the preferences of the expert evaluators. Future work should concentrate on enhancing the generalization capabilities of the fine-tuned model, possibly by incorporating a more diverse set of different evaluation models or by employing a more extensive preference building method (see section 6.2).

6.1.1 Timings of the Evaluations

Timing Results for Each Model The table below summarizes the timings (in seconds) for extraction as presented in Table table 5:

Notably, the time required to process summaries and full texts is significantly longer. Each full text and the subsequent generation of competency profiles necessitate multiple LLM passes. Specifically, the system processes each full text separately, followed by competency profile generation, resulting in a sixfold increase in processing time for abstracts (1 prompt vs 6 prompts). Moreover, full-text prompts, which are longer than abstracts,

Model	Abstracts	Full Texts	Summaries
GPT-4o-mini	5 sec	40 sec	30 sec
Phi3-14B	15 sec	180 sec	160 sec
Phi3-3.8B	12 sec	120 sec	90 sec
Mixtral-8x7B	11 sec	120 sec	80 sec

Table 5: Single-threaded timings of the different extraction methods for each LLM.

require the system to manage more data. Consequently, the computational time escalates quadratically with the length of the input as outlined in section 2.3.5. Although the number of prompts is only six times greater, the computational time can be up to twelve times longer.

The GPT-4o-mini model achieves faster processing times due to throughput optimizations and direct access via the OpenAI API, unlike models that rely on local hardware such as the A100 GPU in use at the LocalAI instance.

Parallelization Potential and Computational Optimization As indicated in Table table 5, the timings represent single-threaded operations on extensive text inputs. However, tasks such as full-text extraction can be efficiently parallelized. Whether using five GPUs for local processing or making five simultaneous API calls, the time required to process an author can be reduced to less than 15 seconds for GPT-4o-mini. This approach significantly enhances efficiency, providing flexible deployment options to suit different needs, including considerations for data privacy.

6.2 Evaluation of the Automatic Evaluation

The correlation of model evaluations is less distinct than anticipated.

The fine-tuned model displays a preference rate of 20.13% in the automatic evaluation, markedly higher than the 9.09% observed in the expert evaluation. This discrepancy likely mirrors the earlier noted disparities in evaluation consistency related to the fine-tuning approach, as outlined in fig. 20. The finetuned model was trained to optimize the preference rate of the exact evaluation model, that was used for automatic evaluation. Therefore the evaluation LLM seems to prefer the fine-tuning results more than the experts actually do, which would indicate limited generalization capabilities of the finetuned model. Future research should aim at enhancing these capabilities, ensuring both automatic and expert evaluations yield similar preferences. It is important to note that the fine-tuned model comprises only has one-third of the profiles of the other models in the evaluation, as each of the other models submitted 3 profiles, one for each of the extraction methods, while the fine-tuned model only submitted one profile. Therefore the

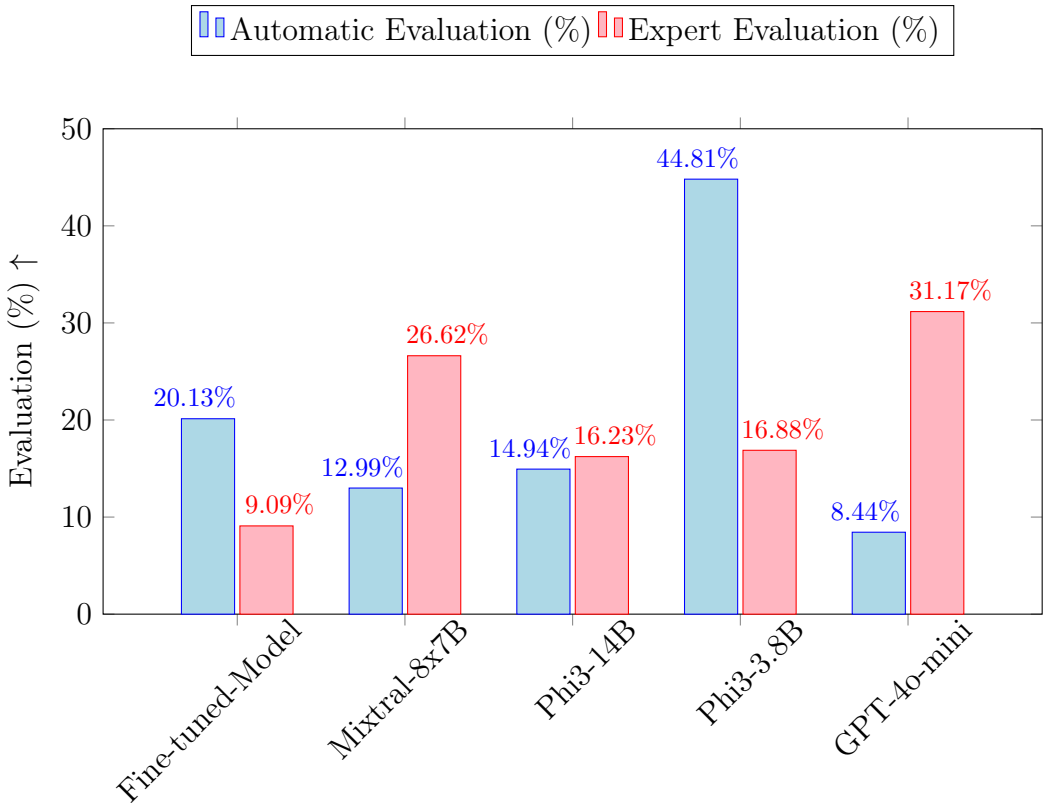

Figure 21: Preference rates (↑: Higher is better) of the different models in the automatic and expert evaluations.

preference rate of the fine-tuned model is not directly comparable to the other models (see section 6.3.2 for more on that).

The Phi3-3.8B model exhibits a significantly higher preference rate in the automatic evaluation than in the expert evaluation, as predicted in section 4.7.2. Here, the variability in the Phi3 models was greater than that in other models.

In contrast, the expert evaluation confirms the hypothesis that larger and stronger base models perform better in the extraction task. The GPT-4o-mini and Mixtral-8x7B models achieve expert evaluation preference rates of 31.17% and 26.62% respectively, aligning better with expectations compared to their lower rates in the automatic evaluation. This evaluation discrepancy underscores the need for further investigation into the causes of these differences. While the automatic evaluation offers an initial gauge of model performance, it should not serve as the sole evaluation criterion. Expert evaluation is recommended as the more dependable method for future assessments.

The recommendation for implementing the competency extraction system includes using the GPT-4o(-mini) model, or, if API call costs or data privacy concerns arise, the Mixtral-8x7B model. Alternatively, exploring the fine-tuning of a more robust base model could potentially enhance outcomes, as evidenced by the significant improvements demonstrated

by the fine-tuned model (see section 6.3.2).

Results of Bias Management The reliability of automatic evaluations is not as high as that of expert evaluations due to biases inherent in the training data of LLMs. This section examines the limitations of automatic evaluations and hypothesizes that such evaluations are biased.

The bias management capabilities of LLMs, specifically models Mixtral-8x7B, Llama3-70B, and GPT-4o, were assessed using a dataset comprising over 70 profiles. The findings indicated a positional bias towards the first profile presented in the evaluation prompts. Although this bias was somewhat reduced through systematic shuffling and balanced presentation of examples, it remained evident across different models. Moreover, the models exhibited inconsistencies; they assessed the same profiles differently depending on their presentation order.

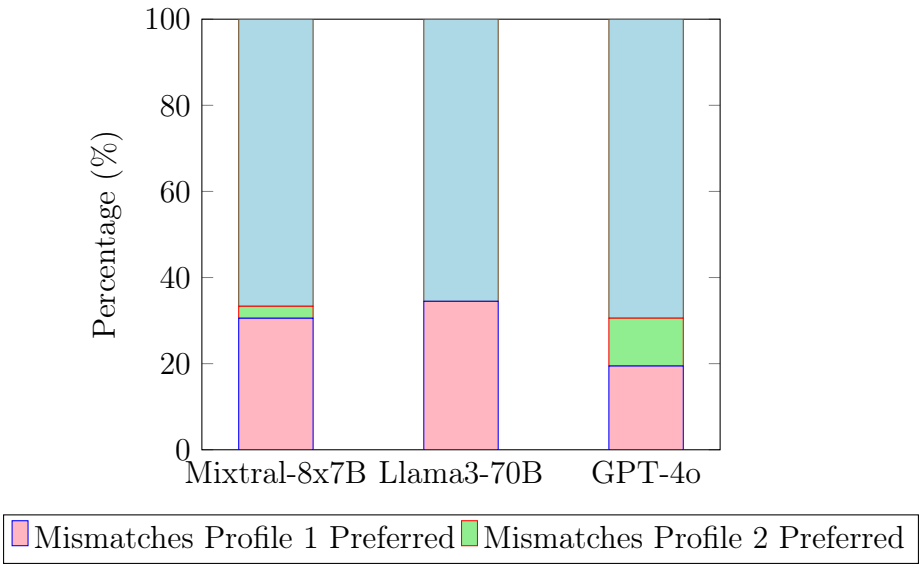

Figure 22: Mismatches (everything that is not blue) with which profile was preferred in the evaluation.

Evaluation of Biases Figure 22 reveals significant mismatches (everything not in blue, so 30%) and fig. 23 positional bias in all models, particularly the high number of mismatches is concerning. Notably in fig. 22, most mismatches resulted from the model favoring the first profile repeatedly (red), while only rarely favoring the second profile repeatedly (green), indicating that a lot of the mismatches are a result of a strong positional bias. The positional bias can be seen in fig. 23 where an optimal preference rate of 50% for either position is clearly biased towards the first profile. This positional bias is challenging to address due to its deep roots in the model's training data.

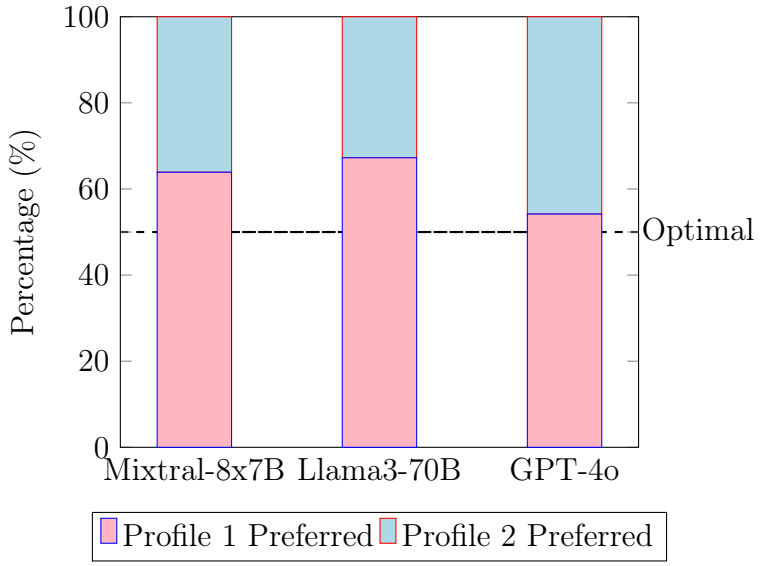

Figure 23: Profile Preferences (Profile 1 vs Profile 2)

Calculations The calculations for estimating errors due to positional bias and inconsistencies are as follows:

The error from positional bias E_{pb} is computed by: $E_{pb} = (P_1 - 50) \times 2$, where P_1 is the frequency of preference for Profile 1.

The inconsistency I is then calculated by: $I = M - E_{pb}$, where M represents the frequency of mismatches.

The consistency C is subsequently derived as: $C = 100 - I$.

Metric	P_1 (%)	M (%)	E_{pb} (%) ↓	I (%) ↓	C (%) ↑
Mixtral-8x7B	63.89	33.33	27.78	5.55	94.45
Llama3-70B	67.24	34.48	34.48	0	100
GPT-4o	54.17	30.56	8.34	22.22	77.78

Table 6: Calculation of the error due to positional bias (↓: Lower is better), inconsistency (↓: Lower is better), and consistency (↑: Higher is better) for the different models.

The appendix appendix B contains detailed calculations, which demonstrate that while some models like Llama3-70B exhibit high consistency, others do not. This discrepancy aligns with literature findings, suggesting a typical consistency range between 55-81% for sota models in more complex coding tasks [Min+24b; Zhe+23]. Given the reduced complexity of the evaluation task, the observed consistency rates should be higher. Therefore the consistency, while not optimal, is to be expected based on current research findings.

Although measures were taken, residual biases indicate that further enhancements in bias mitigation are necessary. Larger models like GPT-4o, while exhibiting less positional bias, also show more pronounced issues with self-consistency. This suggests that even

larger models may not be infallible and could benefit from additional research into bias management techniques to improve evaluation consistency and reduce biases. Further investigation into the use of even larger models is recommended, despite their higher operational costs. Future work can also explore other techniques to improve bias management and consistency, such as using self-consistency algorithms [Min+24b] and Elo ratings (see section 2.5.4) for more reliable evaluations.

Calculation of Costs The cost analysis for employing GPT-4o in the dataset creation for fine-tuning is detailed below:

The dataset comprises approximately 7000 samples generated using the tournament method, which involves 7 pairwise comparisons per 8 profiles, leading to 12 preferences. The costs are calculated as follows:

- For each pairwise comparison, approximately 4500 tokens are required for the input, and 400 tokens for the output.
- For 7000 samples, utilizing the tournament method:
  - Input tokens: $7000 \times \frac{7}{12} \times 4500 = 18.375M$
  - Output tokens: $7000 \times \frac{7}{12} \times 400 = 1.6M$
- The cost per million tokens is 5€ for input and 15€ for output:
  - Input: $18.375M \times 5\frac{€}{M} = 91.875€$
  - Output: $1.6M \times 15\frac{€}{M} = 24€$
- The total cost for evaluating 7000 samples is approximately 115€.

This cost is considered prohibitive for this project due to budget constraints. However, the findings suggest that employing a more expert model could slightly improve bias reduction and consistency, warranting further investigation into cost-effective methods.

Self consistency methods as described in [Min+24b] would additionally at least double the costs, making them even more prohibitive. However, the results of this thesis suggest that the costs might be justified, as the self-consistency methods could significantly improve the consistency of the evaluations, which is a key factor in the reliability of the evaluations and therefore the quality of the fine-tuning dataset.

6.3 Investigation of Fine-Tuning Results

This subsection presents a systematic exploration of the fine-tuning process. The analysis includes a series of iterative training runs aimed at refining the system's performance

through various adjustments in training parameters and strategies. Key findings highlight the impacts of these adjustments on the model's preference rates, both in automatic and expert evaluations, offering insights into the potential and limitations of successive fine-tuning iterations. Comparative data and diagnostic issues encountered during the process are also discussed, providing a comprehensive overview of the fine-tuning landscape.

6.3.1 Analysis of the Fine-Tuning Process

This section outlines the training procedure and examines various strategies implemented to evaluate and enhance the system's performance. The objective was to determine the effectiveness of the initial training setup and whether alternative approaches could provide superior results. Several configurations were tested and their outcomes were compared to ensure optimal fine-tuning of the system.

1. **First training run:** The system used the Phi3-3.8B base model available online and generated 2400 new samples for fine-tuning. Training occurred at a learning rate of 2e-5 across three epochs, totaling approximately 6.5 hours on an A100 with 80 GB of VRAM. Subsequent to this training, the system exhibited a 74% preference rate over the online base model, indicating that fine-tuning resulted in a notable enhancement, at least in automatic evaluation.

2. **Second training run:** Utilizing 2100 samples generated by the fine-tuned system from the initial iteration, the training maintained a learning rate of 2e-5, lasting about six hours. The preference rate over the online base model decreased to 71%, and the system from the second run was preferred in only 42% of comparisons against the system from the first run, which was preferred 58

3. **Third training run:** Employing the same samples as the second run, this iteration extended the training for six hours with a reduced learning rate, doubling the number of epochs. This strategy improved the performance, achieving an 80% preference rate over the online base model and a 52% preference rate compared to the system from the initial run. These results suggest that extending the number of training epochs beyond initial expectations (six epochs appeared to still improve) with a significantly reduced learning rate leads to enhanced outcomes.

4. **Fourth training run:** A fourth iteration was conducted using 1750 new samples generated by the system after the third run, fine-tuned over six epochs at a learning rate of 5e-6 with an effective batch size of 32. The preference rate was 77% over the online base model and 48% over the system from the first run. Although these results were marginally less favorable than those of the third run, the considerably reduced learning rate provided more stable training. Loss and accuracy plots had

not yet plateaued, suggesting that further training could yield further improvements. Future work could examine the potential for extending the iterative approach further, though it appears there is a point where additional iterations cease to yield further enhancements. The results are already significantly better than the online base model, with an 80% preference rate following the third run, demonstrating the high efficacy of the system.

5. **Fifth training run:** For verification, a fifth session utilized the entire dataset from iterations one and two and yielding nearly 4500 samples, with a slightly larger effective batch size and a reduced learning rate from the outset. This system achieved a 77% preference rate over the online base model, surpassing the system from the first iteration but not reaching the performance level of the best iteratively trained system. These results indicate that while the quality of the dataset significantly influences performance enhancements, the iterative approach, where only high-quality samples are retained, results in superior outcomes.

6. **Sixth training run:** Lastly, the standard SFT, as proposed by the DPO paper, was explored. The DPO paper suggests that the system should undergo SFT prior to applying the DPO method. The intent was to assess whether this process would improve the preference rate compared to both the original model and the system from the first run. For this purpose, an SFT dataset was created using the 4500 samples from the first two runs, converted from DPO format to SFT format by concatenating the *prompt* and the *chosen* answer of each sample. The system trained on this dataset demonstrated an 80% preference rate over the online base model but only a 35% preference rate compared to the system from the first run.

6.3.2 Comparison of Fine-Tuning Models with Expert Evaluations

The performance of the fine-tuned model is evaluated by comparing its preference rate to that of other models in the *Extract from Abstracts* task, as depicted in table 7.

Model	**Model Size**	**Preferred by Experts ↑**
Fine-tuned	3.8B	16.47%
Mixtral	8x7B	34.12%
GPT-4o-mini	Unknown	31.76%
Phi3	14B	11.76%
Phi3	3.8B	5.88%

Table 7: Expert Preferences (↑: Higher is better) by Model Size

In 16.47% of evaluations, experts preferred the final fine-tuned model, which represents a threefold improvement compared to the base model. This rate corresponds well with

the outcomes from automatic evaluations following the fine-tuning iterations where the fine-tuned model achieved 80% preference rate over the base model. Despite these improvements, the model remains less preferred compared to the larger models such as Mixtral-8x7B (34.12%) and GPT-4o-mini (31.76%).

A discernible correlation exists between model size and expert preference, wherein larger models exhibit preferred extraction performace rates. For demanding performance criteria, particularly in intricate extraction tasks, the larger models like Mixtral-8x7B and GPT-4o-mini demonstrate greater capability, assuming sufficient computational or financial resources are available.

The analysis concludes that although the fine-tuned model shows marked enhancements over the base model, it continues to be outperformed by larger models in terms of expert preference. It is implied that refining the fine-tuning process and employing a larger base model to fine-tune may further improve the model's performance in competency extraction tasks and achive highest preference rates in expert evaluations.

6.3.3 Problem Diagnosis and Improvement Suggestions

Model Inconsistencies and Faulty Results Inspecting the fine-tuning results revealed inconsistencies in the evaluation model, leading to frequent crashes in 15-20% of cases. These erroneous results were incorporated into the dataset, creating distortions that persisted beyond the evaluation phase and potentially affecting the comprehensive assessment.

Termination of Generation at 4k Tokens The implementations of the Llama3-70B model and the Phi3-3.8B model encountered a significant limitation: both models ceased token generation at 4k tokens, despite a designed maximum context length of 8k and 128k tokens, respectively. This limitation was linked to the RoPE implementation and Key-Value (KV) cache scaling issues (refer to section 2.3.5 for additional details). Both models operated effectively below the 4k token threshold and also performed well when the prompt initially contained more than 4k tokens. However, crossing the 4k token limit resulted in either termination or the generation of nonsensical outputs. The problem was, that different scaling factors were used below 4k tokens, where the model was initially trained on, and above 4k tokens, where RoPE extrapolation was used and subsequently fine-tuned to handle longer contexts when provided with other scaling factors. Only did the implementations of the models not account for a discrepancy where the old computed values with the old scaling values were still contained and used from the KV cache, leading to a mix of short scaling factors and long scaling factors being used in the same context, which caused the model to generate nonsensical outputs. This problem was reported to

Hugging Face, and a temporary fix was implemented by adjusting the maximum position embedding to 8k tokens. This adjustment ensured functionality below the 4k token limit but slightly altered positional embeddings from the initial training data. A subsequent pull request in the Hugging Face library has addressed this issue permanently. Nevertheless, the solution was implemented after the issue had already caused a significant amount of evaluations to crash, as previously noted.

For an illustration of the impact of the 4k token limit on model outputs, see [20] and [21].

Consistency Problems The bias evaluation (section 6.2) indicated a consistency rate of approximately 70%, which led to discrepancies in pairwise preference comparisons. The presence of inconsistent outputs across evaluations compromised reliability and distorted the dataset.

Hardware Limitations Hardware constraints impacted the fine-tuning process, characterized by extended waiting times for fine-tuning and dataset generation at the BW-Uni-Cluster, compounded by frequent system crashes on the AIFB LocalAI server. These issues not only delayed the fine-tuning process but also prolonged the period required for model optimization.

6.3.4 Publication of the Fine-Tuning Dataset

The dataset used for fine-tuning the competency extraction model is publicly accessible to support additional research and facilitate the replication or expansion of the findings. The dataset is available on Hugging Face in the repository competency-extraction-dpo[22] and holds an Apache 2.0 license.

Opportunities for Further Research The public availability of this dataset enables the exploration of scalability in fine-tuning methods and the refinement of bias management techniques, both identified as potential enhancements during the evaluation of the model. Furthermore, it provides opportunities to establish benchmarks with newer models.

[20] gitlab.kit.edu/kit/aifb/BIS/abschlussarbeiten/2024/ma-braun-competence-extraction/public-data/-/raw/main/Llama3_4k_Token_Problem.png

[21] gitlab.kit.edu/kit/aifb/BIS/abschlussarbeiten/2024/ma-braun-competence-extraction/public-data/-/raw/main/Phi3_4k_Token_Problem.png

[22] huggingface.co/datasets/BertilBraun/competency-extraction-dpo

6.4 Evaluation of the Domain-Agnostic System

This section presents a comprehensive evaluation of the domain-agnostic system, using a LLM as the evaluator to assess competency profiles across various data scenarios. The analysis spans four data subsets to explore the impact of data availability on the accuracy and consistency of competency evaluations. Results from scoring 51 companies demonstrate how data quality influences the representativeness of competency profiles. Further discussions highlight the operational costs and usability of the system in diverse application contexts, emphasizing scalability and adaptability.

6.4.1 Results of the Scoring System

Investigations were conducted to determine the optimal functionality of the system, utilizing a LLM as the scorer. The system assigned scores ranging from 1 to 100 across a dataset of 51 companies, with the outcomes summarized in table 8.

The system showcased its adaptability by producing detailed competency profiles from patents and annual reports, aligning well with manually crafted summaries. Its ability to adjust to various domains was demonstrated, with slight alterations to the input data and examples facilitating precise competency extraction in the corporate realm.

The impact of various data types and their availability on the scoring system was assessed through analysis of the following subsets:

1. Subset with 5 or More Patents
2. Subset with 5 or More Patents but No Annual Reports
3. All Companies with All Annual Reports
4. All Companies with No Annual Reports

The analysis demonstrates a definitive trend where an increase in the number of abstracts and reports correlates with more representative and accurate competency profiles. This correlation is illustrated by the data plotted in fig. 24.

The results indicate that a greater availability of abstracts reduces the likelihood of inadequate competency extraction. The increased volume of good-quality data facilitates the creation of profiles that more accurately represent the companies' competencies. This is further supported by the reduced standard deviation, which signifies a decrease in the likelihood of outlier profiles as the data volume increases.

Metric	Subset 1	Subset 2	Subset 3	Subset 4
Mean Score ↑	78.64	71.36	60.10	47.65
Standard Deviation ↓	21.46	23.14	27.69	28.82
Median Score ↑	85.0	85.0	70.0	60.0
Minimum Score ↑	20.0	30.0	0.0	0.0
Maximum Score ↑	95.0	95.0	95.0	85.0
Mean Number of Abstracts	12.36	12.36	4.24	4.24
Mean Number of Annual Reports	10.27	0.0	10.39	0.0
Number of Scores	11	11	51	51

Table 8: Results of the Scoring System (↑: Higher is better, ↓: Lower is better)

It's important to note that a single patent abstract might not be directly representative of the entire company. A greater volume of data provides a more comprehensive overview of the company's competencies, making the profiles more representative.

The results affirm that the scoring system is effective for evaluating the quality of competency profiles extracted. There is a consistent correlation between the number of abstracts and the scores, suggesting that the system scales effectively with data availability. The performance of the system remains consistent across different subsets, with the mean score and standard deviation both decreasing as data diminishes. Notably, the scoring system tends to avoid 50% scores and displays a tendency toward scores rounded to the nearest 5 or 10, which may suggest a systematic bias that does not, however, lead to a non-representative assessment of competencies. Future research should explore these potential biases and further assess the correlation between the LLM evaluations and expert assessments on a broader and more varied dataset.

Challenges and Critical Observations Encountered challenges included the absence of domain-specific nuances in corporate texts. While patents and technical reports present structured information, annual reports predominantly address financial and strategic aspects of business, thus influencing the competency extraction towards management and leadership skills. This underscored the necessity for more specialized examples and potentially customized prompts to enhance the system's accuracy and applicability in the corporate setting.

6.4.2 Cost Analysis of the Extractions

Token Costs per Operation This section provides an analysis of the estimated token consumption for various operations. These are calculated as the average over all respective

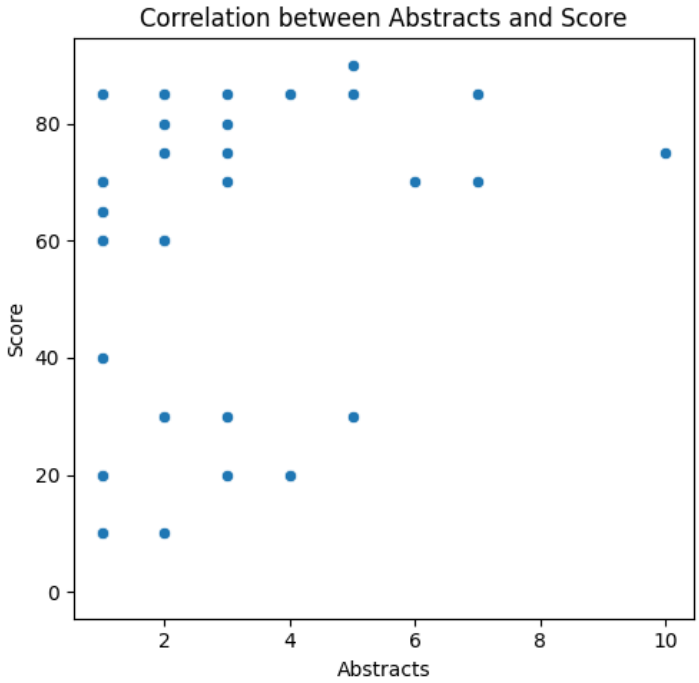

Figure 24: Correlation between abstracts and scores on the full dataset of 51 companies with all annual reports included (Subset 3) with the exclusion of 3 companies with 19, 31 and 50 abstracts and scores of 85, 95 and 85, respectively.

data in the CAS evaluation dataset using the OpenAI tokenizer[23].

- **Profile Merge:** The process of merging tokens from different documents to create a comprehensive competency profile typically requires 2.2k tokens for 5 profiles to be merged into one.
- **Extraction from Abstracts:** Generating profiles based on abstracts necessitates around 2.7k tokens, with the assumption that approximately five abstracts are analyzed.
- **Extraction from Full Texts:** The extraction from full texts is notably more token-intensive, demanding between 1.5k and 8k tokens per document, based on the document's length.

Token Cost Calculation Formula The formula to compute the overall token expenditure is expressed as follows:

$$T_{cost} = \left\lceil \frac{N_{abstracts}}{5} \right\rceil \times 2.7k + N_{documents} \times T_{tokens/doc} + \log_2 \left(\left\lceil \frac{N_{abstracts}}{5} \right\rceil + N_{documents} \right) \times 2.2k$$

[23](visited 10/01/2024) platform.openai.com/tokenizer

Here, $N_{abstracts}$ and $N_{documents}$ signify the counts of abstracts and complete documents, respectively, and $T_{tokens/doc}$ represents the token allocation per document, which varies from 1.5k to 8k tokens.

Example Calculation Consider an example where there are 50 abstracts and 12 lengthy (approx. 8k tokens or 13 full pages) documents:

$$T_{cost} = 10 \times 2.7k + 12 \times 8k + \log_2(10 + 12) \times 2.2k = 133k \text{ tokens.}$$

Cost Comparison of Models Here, the token costs per million tokens differ according to the model utilized. By applying the above token calculations, one can ascertain the costs associated with various GPT models:

- **GPT-4o:** With a rate of €5 per 1 million tokens, the expense is approximately €0.66 for each profile.
- **GPT-4o-mini:** With a rate of €0.15 per 1 million tokens, the expense is approximately €0.002 for each profile.

6.4.3 Usability of the System

The evaluation showed that the system designed for competency extraction is highly flexible and adaptable across domain-agnostic contexts. Its architecture supports integration with multiple data sources and facilitates domain adaptation with minimal reconfiguration.

A significant advantage of the system lies in its ability to process varied types of input data, such as abstracts and full-text documents, to create competency profiles. The system is designed for immediate use, indicating that it requires no extensive customization for new applications.

Additionally, the quality of the competency profiles generated by the system enhances with the increase in data volume. Utilizing both abstracts and full texts contributes to a more detailed and precise depiction of the competencies extracted. Although not yet validated on extremely large datasets, the system's design is capable of scaling effectively to handle up to 100 documents.

This implies that the system can be implemented in a diverse array of application areas, ranging from corporate evaluations to academic research, with little need for domain-specific adjustments.

Nevertheless, there are opportunities for further refinement. Sectors with limited or unstructured data might not achieve the same accuracy level. Ongoing enhancements in processing increasingly complex datasets could further improve the system's usability and extend its applicability to a wider variety of domains.

7 Summary and Outlook

7.1 Summary and Recommendations

This thesis explored the utilization of LLMs for competency extraction from scientific documents. The research focused on three different context reduction strategies, alongside a fine-tuning method, culminating in the development of a system capable of swiftly extracting precise competency profiles from a voluminous document set. Furthermore, a domain-agnostic system was established, functioning as a black box across various application areas, provided there is access to sufficient and high-quality data. The investigation revealed that the quality of extraction is significantly influenced by the volume and clarity of the data and is particularly dependent on the LLM employed. The analysis compared various LLMs and provided recommendations for their optimal deployment in competency extraction tasks. It was determined that fine-tuning a robust LLM using the developed iterative fine-tuning technique is likely to achieve the best outcomes, albeit it demands substantial effort to implement. In contrast, the system demonstrated optimal performance using the *Extract-from-Abstracts* method with a robust LLM as the foundational model, specifically GPT-4o(-mini) in this instance. The cost associated with this method is 1.4 cents per profile for GPT-4o and 0.04 cents per profile for GPT-4o-mini, respectively. The recommendation is to evaluate the largest available LLMs for superior results, as they tend to outperform smaller models, while considering the potential necessity of local LLM hosting of open weight LLMs to address data privacy concerns.

7.2 Usability and Recommendations for the Competency Network

The *Extract from Abstracts* method emerges as the most suited method for the extraction of competencies, given its lower computational demands and absence of setup costs associated with fine-tuning. Despite a possible slight underperformance compared to an optimized finetuned model, its efficacy makes it a preferred choice for most applications. Conversely, the *Extract from Summaries* method is advisable when abstracts are not accessible, offering a favorable compromise between performance and computational load. The performance disparity between summaries and full texts is negligible, yet the *Extract from Summaries* method exhibits a 10-30% reduction in computational effort.

Future research should explore the efficacy of a larger or more powerful base model when finetuned and should aim to diminish the cumulative errors present in the summary and full text methodologies, beginning with enhanced PDF text extraction techniques. Such improvements might elevate the performance to levels more closely aligned with the

Extract from Abstracts method. Additionally, it is necessary to conduct evaluations on a broader scale, as the current dataset of nine evaluations with 13 profiles each—totaling 117 evaluations—may not suffice to yield representative results.

Feedback from expert evaluators indicates that selecting more representative papers for processing is an essential direction for future research. The papers chosen for evaluation did not accurately reflect the experts' actual competencies. The Competence Pool requires users to manually select papers that are emblematic of an author's competencies, which is feasible for extracting a single profile in under a minute. However, preparing the entire evaluation tournament required nearly one hour, including manual interventions to incorporate fine-tuning results and verify the profiles, thereby rendering real-time manual selection and tournament creation impractical in this instance, which is why the most cited papers were selected instead.

Cost Analysis of the Extractions Generating profiles based on abstracts necessitates around 2.7k tokens, with the assumption that five abstracts are analyzed. The cost for extracting a profile based on five abstracts is calculated as follows:

GPT-4o: With a rate of €5 per 1 million tokens, the expense is approximately:

$$2.7k \text{ tokens} \times \frac{5 \text{ €}}{1.000.000 \text{ tokens}} = 0.014 \text{ €}$$

GPT-4o-mini: With a rate of €0.15 per 1 million tokens, the expense is approximately:

$$2.7k \text{ tokens} \times \frac{0.15 \text{ €}}{1.000.000 \text{ tokens}} = 0.0004 \text{ €}$$

7.3 Future Work

This subsection outlines the planned advancements in iterative fine-tuning strategies, the integration of new language models, and the enhancement of domain-agnostic systems.

7.3.1 Enhancing the Iterative Fine-tuning Strategy

Biases:

The examination of biases and the strategies to mitigate them are essential for the enhancement of LLMs. The technique of *self-consistency testing* has been employed to diminish positional and ranking biases through the aggregation of multiple outputs [Min+24b]. This approach has improved consistency and reduced bias in listwise ranking tasks by 7-18% in models such as GPT-3.5 and Llama3 [Tan+24]. Despite its benefits, this

method is computationally demanding, as it requires generating each preference at least twice, thus more than doubling the time and cost involved. In instances of inconsistent preferences, it may be necessary to generate outputs three or more times to implement majority voting.

Another method involves the Elo Rating System (Elo) system, which may offer a balance between consistency and accuracy in a self-correcting framework. The efficacy of the Elo system in this context warrants further examination.

Exploration of Error Sources The iterative fine-tuning method has identified several significant error sources that affect LLMs, including inaccurate outputs, backend failures, and crashes at token lengths exceeding 4k. It is imperative to address these issues to ensure system reliability. These problems were only discovered post-evaluation, indicating that the fine-tuning process was compromised by these errors, potentially affecting 15-20% of the generated profiles.

Enhanced Expert Models The effectiveness of the fine-tuning process relies heavily on the quality of the expert models used for evaluating the generated profiles. Research indicates that employing larger LLMs as expert evaluators fosters more consistent and superior outputs [CL23; Liu+23]. This evidence, alongside the findings of this thesis, underscores the pivotal role of expert model robustness in the outcomes of fine-tuning.

Stronger Base Model It is posited that larger, more robust and stronger base models are likely to yield better results due to their enhanced learning and generalization capabilities. Nonetheless, the adoption of these models incurs significant increases in computational costs, particularly with regard to training duration and resource allocation. Future research should determine whether the performance enhancements justify the escalated resource expenditures.

Optimization of Hyperparameters The methodical exploration of hyperparameters, including learning rate, batch size, and the number of learnable epochs, is critical for the enhancement of fine-tuned LLMs performance. Investigating various configurations may identify the most effective setups.

7.3.2 Updating and Integrating New LLMs

Drop-In Replacements With the advent of stronger LLMs, such as GPT-4 a year ago, enhancements in language comprehension and generation have been observed. The

integration of these stronger LLMs into established systems necessitates a thorough assessment to confirm their superiority over earlier or specialized models.

Implementing new LLMs as "drop-in" replacements might improve performance; however, rigorous testing is essential to verify whether these new models surpass the capabilities of existing ones within specific applications.

Evaluation of Specialized Models Models developed for specific tasks, such as text analysis in sectors like healthcare or finance, often outperform general-purpose models in competency extraction. This is due to their enhanced ability to process highly structured data and detailed domain-specific knowledge [Leh+23].

7.3.3 Enhancement of the Domain-Agnostic System

Refinement of Prompt Application The exploration of the impact of specifically tailored prompts on the system's performance may result in considerable enhancements. The sensitivity to prompts is pivotal in LLMs. Within a domain-agnostic framework, the optimization of prompts ensures that competency extraction is effectively adapted across various application domains. Systems such as AutoPrompt[24] from [Kha+23] could be utilized to automatically refine task instructions, facilitating the creation of more accurate competency profiles.

Enhancement of the Example Database The development of a robust example database is crucial for the optimization of the RAG system. Employing a system for the automated replenishment of the vector database can enhance the relevance and quality of the examples retrieved. Utilizing top-tier LLMs to produce high-quality examples, which are subsequently manually reviewed and integrated into the vector database, ensures that the system progressively improves its capability to retrieve the most pertinent examples.

Broadening to Additional Domains Extending the domain-agnostic system to diverse fields such as medicine, law, or engineering represents a significant prospective development. Tailoring prompts and system settings may be essential to attain optimal competency extraction in these fields. The expansion of the competency extraction system into various areas enhances its adaptability, though further examination is required to ascertain the necessary modifications for each domain.

Summarization and Competency Profiles A potential enhancement involves employing summarization techniques to develop competency profiles based on document

[24](visited on 10/01/2024) github.com/Eladlev/AutoPrompt

summaries rather than entire texts. Expert assessments have demonstrated that extractions based on abstracts surpass those from full texts in clarity and relevance. Investigating summarization approaches to condense full texts into abstracts, and then utilizing the *Extract-from-Abstracts* method with the actual abstracts combined with the generated summaries could elevate the system's overall efficacy. This advancement necessitates additional research.

7.3.4 Expansion of the Evaluation to New Use Cases

Integration into Existing Corporate Software The integration of the competency extraction system into corporate HR or Knowledge Management Systems (KMSs) provides substantial benefits in functionality and efficiency. Such systems, utilizing LLMs for competency extraction, could improve the automation of skill identification and competency profile creation, thereby reducing manual labor.

Adapting the system for use within existing corporate software necessitates modifications to meet the unique requirements of the corporate environment, which include complex organizational structures and domain-specific terminologies. The application of frameworks based on pre-trained language models, specifically developed for corporate data domains, would be advantageous.

However, incorporating these systems into HR or KMSs presents challenges, notably in terms of scalability and data privacy. Implementations on a large scale would demand strategies to manage extensive data volumes with scalable performance while preserving accuracy. Addressing privacy concerns is crucial, especially when handling sensitive employee information. Employing local LLM models represents a viable approach to these challenges.

Key Considerations

- **Scalability:** The system is required to manage large-scale data efficiently while ensuring accurate competency extraction.
- **Data privacy:** Ensuring the protection of sensitive employee data through the use of local LLM models and secure data management practices is essential.
- **Customization:** It is crucial to tailor the system to align with domain-specific terminologies and organizational structures to achieve optimal performance.

References

[Anw+24] Usman Anwar et al. *Foundational Challenges in Assuring Alignment and Safety of Large Language Models*. 2024. arXiv: 2404.09932 [cs.LG].

[Aro+23] Simran Arora et al. "Language Models Enable Simple Systems for Generating Structured Views of Heterogeneous Data Lakes". In: *ArXiv* (2023). DOI: 10.48550/arXiv.2304.09433.

[AS15] Meera Alagaraja and Brad Shuck. "Exploring Organizational Alignment-Employee Engagement Linkages and Impact on Individual Performance." In: *Human Resource Development Review* (2015). DOI: 10.1177/1534484314549455.

[AV01] Paul C.H. Albers and Han de Vries. "Elo-rating as a tool in the sequential estimation of dominance strengths". In: *Animal Behaviour* (2001). DOI: 10.1006/anbe.2000.1571.

[Bay+18] Julie Elizabeth Bayley, David Phipps, Monica Batac, and Ed Stevens. "Development of a framework for knowledge mobilisation and impact competencies". In: *Evidence and Policy* (2018). DOI: 10.1332/174426417x14945838375124.

[Bro+20] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *Neural Information Processing Systems* (2020). arXiv: 2005.14165 [cs.CL].

[Bö+18] Katy Börner et al. "Skill discrepancies between research, education, and jobs reveal the critical need to supply soft skills for the data economy." In: *Proceedings of the National Academy of Sciences* (2018). DOI: 10.1073/pnas.1804247115.

[Cam+10] Dr. Craig Campbell, Ivan Silver, Jonathan Sherbino, Olle ten Cate, and Eric Holmboe. "Competency-based continuing professional development". In: *Medical teacher* 32 (2010), pp. 657–62. DOI: 10.3109/0142159X.2010.500708.

[CL23] Cheng-Han Chiang and Hung yi Lee. "Can Large Language Models Be an Alternative to Human Evaluations?" In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2023). DOI: 10.18653/v1/2023.acl-long.870.

[DAG15] Philipp Doebler, Mohsen Alavash, and Carsten Giessing. "Adaptive experiments with a multivariate Elo-type algorithm". In: *Behavior Research Methods* (2015). DOI: 10.3758/s13428-014-0478-7.

[Det+23] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. "QLoRA: Efficient Finetuning of Quantized LLMs". In: *Neural Information Processing Systems* (2023). arXiv: 2305.14314 [cs.LG].

[Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *NAACL-HLT* (2019). arXiv: 1810.04805 [cs.CL].

[Dia+24] Shizhe Diao et al. "Active Prompting with Chain-of-Thought for Large Language Models". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2024). DOI: 10.18653/v1/2024.acl-long.73.

[Dou+24] Matthijs Douze et al. *The Faiss library*. 2024. arXiv: 2401.08281 [cs.LG].

[Dun+22] Alex Dunn et al. "Structured information extraction from complex scientific text with fine-tuned large language models". In: *ArXiv* (2022). DOI: 10.48550/arXiv.2212.05238.

[Elm20] Jeffrey L. Elman. "Finding structure in time". In: *Connectionist psychology: A text with readings* (2020). DOI: 10.4324/9781315784779-11.

[FAO17] Markus Freitag and Yaser Al-Onaizan. "Beam Search Strategies for Neural Machine Translation". In: *NMT@ACL* (2017). DOI: 10.18653/v1/w17-3207. URL: http://dx.doi.org/10.18653/v1/W17-3207.

[FLD18] Angela Fan, Mike Lewis, and Yann Dauphin. "Hierarchical Neural Story Generation". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2018). DOI: 10.18653/v1/p18-1082.

[Fre+21] Markus Freitag et al. "Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation". In: *Transactions of the Association for Computational Linguistics* (2021). DOI: 10.1162/tacl_a_00437.

[FZS22] William Fedus, Barret Zoph, and Noam Shazeer. "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity". In: *Journal of machine learning research* (2022). arXiv: 2101.03961 [cs.LG].

[Gas98] S. I. Gass. "Tournaments and transitivity and pairwise comparison matrices". In: *The Journal of the Operational Research Society* (1998). DOI: 10.2307/3010670.

[GEA19a] Alexandra González-Eras and Jose Aguilar. "Determination of Professional Competencies Using an Alignment Algorithm of Academic Profiles and Job Advertisements and Based on Competence Thesauri and Similarity Measures." In: *International Journal of Artificial Intelligence in Education* (2019). DOI: 10.1007/s40593-019-00185-z.

[GEA19b] Alexandra González-Eras and Jose Aguilar. "Determination of Professional Competencies Using an Alignment Algorithm of Academic Profiles and Job Advertisements, Based on Competence Thesauri and Similarity Measures". In: *International Journal of Artificial Intelligence in Education* (2019). DOI: 10.1007/s40593-019-00185-z.

[Gen+23] Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. "Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (2023). DOI: 10.18653/v1/2023.emnlp-main.674.

[Hah20] Michael Hahn. "Theoretical Limitations of Self-Attention in Neural Sequence Models". In: *Transactions of the Association for Computational Linguistics* (2020). DOI: 10.1162/tacl_a_00306.

[Har04] James Hartley. "Current findings from research on structured abstracts." In: *Journal of the Medical Library Association : JMLA* (2004). DOI: 10.3163/1536-5050.102.3.002.

[HGS21] Anwar Haque, Sayeed Ghani, and Muhammad Saeed. "Image Captioning With Positional and Geometrical Semantics". In: *IEEE Access* (2021). DOI: 10.1109/ACCESS.2021.3131343.

[Hol+20] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. "The Curious Case of Neural Text Degeneration". In: *ICLR 2020 Conference* (2020). arXiv: 1904.09751 [cs.CL].

[Hon+21] Zhi Hong, Logan Ward, Kyle Chard, Ben Blaiszik, and Ian Foster. "Challenges and Advances in Information Extraction from Scientific Literature: a Review". In: *JOM* (2021). DOI: 10.1007/s11837-021-04902-9.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* (1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[Hsi+23] Cheng-Yu Hsieh et al. "Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes". In: *Findings of the Association for Computational Linguistics: ACL 2023* (2023). DOI: 10.18653/v1/2023.findings-acl.507.

[Hu+21] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations* (2021). arXiv: 2106.09685 [cs.CL].

[Hua+24] Hui Huang et al. *On the Limitations of Fine-tuned Judge Models for LLM Evaluation*. 2024. arXiv: 2403.02839 [cs.CL].

[Ing21] Martin Ingram. "How to extend Elo: a Bayesian perspective". In: *Journal of Quantitative Analysis in Sports* (2021). DOI: 10.1515/jqas-2020-0066.

[JT21] Kameni Florentin Flambeau Jiechieu and Norbert Tsopze. "Skills prediction based on multi-label resume classification using CNN with model predictions explanation". In: *Neural Computing and Applications* (2021). DOI: 10.1007/s00521-020-05302-x.

[Kha+23] Omar Khattab et al. *DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines*. 2023. arXiv: 2310.03714 [cs.CL].

[KHB09] Tobias Kollmann, Matthias Häsel, and Nicola Breugst. "Competence of IT Professionals in E-Business Venture Teams: The Effect of Experience and Expertise on Preference Structure". In: *Journal of Management Information Systems* (2009). DOI: 10.2753/mis0742-1222250402.

[Kir+24] Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. "General-Purpose In-Context Learning by Meta-Learning Transformers". In: *CoRR* (2024). arXiv: 2212.04458 [cs.LG].

[Koj+23] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. "Large Language Models are Zero-Shot Reasoners". In: *Neural Information Processing Systems* (2023). arXiv: 2205.11916 [cs.CL].

[Kum+22] Vivek Kumar, Diego Reforgiato Recupero, Rim Helaoui, and Daniele Riboni. "K-LM: Knowledge Augmenting in Language Models Within the Scholarly Domain". In: *IEEE Access* (2022). DOI: 10.1109/access.2022.3201542.

[Leh+23] Eric Lehman et al. "Do We Still Need Clinical Language Models?" In: *ACM Conference on Health, Inference, and Learning* (2023). arXiv: 2302.08091 [cs.CL].

[Lew+21] Patrick Lewis et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Neural Information Processing Systems* (2021). arXiv: 2005.11401 [cs.CL].

[Li+20] Wen Li et al. "Approximate Nearest Neighbor Search on High Dimensional Data — Experiments, Analyses, and Improvement (v1.0)". In: *IEEE Transactions on Knowledge and Data Engineering* (2020). DOI: 10.1109/tkde.2019.2909204.

[Liu+23] Yang Liu et al. "G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (2023). DOI: 10.18653/v1/2023.emnlp-main.153.

[Liu+24] Xiaoran Liu et al. “Scaling Laws of RoPE-based Extrapolation”. In: (2024). arXiv: 2310.05209 [cs.CL].

[LMM23] Baohao Liao, Yan Meng, and Christof Monz. “Parameter-Efficient Fine-Tuning without Introducing New Latency”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2023). DOI: 10.18653/v1/2023.acl-long.233.

[LPD24] Ruosen Li, Teerth Patel, and Xinya Du. “PRD: Peer Rank and Discussion Improve Large Language Model based Evaluations”. In: *Transactions on Machine Learning Research* (2024). arXiv: 2307.02762 [cs.CL].

[LY18] Hana Lee and Young Yoon. “Engineering doc2vec for automatic classification of product descriptions on O2O applications”. In: *Electronic Commerce Research* (2018). DOI: 10.1007/s10660-017-9268-5.

[Mar+22] Asta Margienė, Simona Ramanauskaitė, Justas Nugaras, Pavel Stefanovič, and Antanas Čenys. “Competency-Based E-Learning Systems: Automated Integration of User Competency Portfolio”. In: *Sustainability* (2022). DOI: 10.3390/su142416544.

[Mas+23] Claire M. Mason, Haohui Chen, David Evans, and Gavin Walker. “Illustrating the application of a skills taxonomy and machine learning and online data to inform career and training decisions”. In: *The International Journal of Information and Learning Technology* (2023). DOI: 10.1108/ijilt-05-2022-0106.

[Mau+18] Andrea De Mauro, Marco Greco, Michele Grimaldi, and Paavo Ritala. “Human resources for Big Data professions: A systematic classification of job roles and required skill sets”. In: *Information Processing and Management* (2018). DOI: 10.1016/j.ipm.2017.05.004.

[Mik+13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. “Efficient Estimation of Word Representations in Vector Space”. In: *International Conference on Learning Representations* (2013). arXiv: 1301.3781 [cs.CL].

[Min+24a] Bonan Min et al. “Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey”. In: *ACM Computing Surveys* (2024). DOI: 10.1145/3605943.

[Min+24b] Marcus J. Min et al. “Beyond Accuracy: Evaluating Self-Consistency of Code Large Language Models with IdentityChain”. In: *International Conference on Learning Representations* (2024). arXiv: 2310.14053 [cs.LG].

[Miz+24] Moran Mizrahi et al. “State of What Art? A Call for Multi-Prompt LLM Evaluation”. In: *Transactions of the Association for Computational Linguistics* (2024). DOI: 10.1162/tacl_a_00681.

[Mé05] Sylvie-Anne Mériot. "One or several models for competence descriptions: Does it matter?" In: *Human Resource Development Quarterly* (2005). DOI: 10.1002/hrdq.1138.

[Mü+16] Oliver Müller, Theresa Schmiedel, Elena Gorbacheva, and Jan vom Brocke. "Towards a typology of business process management professionals: identifying patterns of competences through latent semantic analysis". In: *Enterprise Information Systems* (2016). DOI: 10.1080/17517575.2014.923514.

[Ngu+24] Khanh Cao Nguyen, Mike Zhang, Syrielle Montariol, and Antoine Bosselut. "Rethinking Skill Extraction in the Job Market Domain using Large Language Models". In: *NLP4HR* (2024). arXiv: 2402.03832 [cs.CL].

[NOS17] Sahand Negahban, Sewoong Oh, and Devavrat Shah. "Rank Centrality: Ranking from Pairwise Comparisons". In: *Operations Research* (2017). DOI: 10.1287/opre.2016.1534.

[OMK21] Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. "A Survey of the Usages of Deep Learning in Natural Language Processing". In: *IEEE Transactions on Neural Networks and Learning Systems* (2021). DOI: 10.1109/tnnls.2020.2979670.

[Ope+23] OpenAI et al. "GPT-4 Technical Report". In: (2023). arXiv: 2303.08774 [cs.CL].

[Osw+23] Johannes von Oswald et al. "Transformers learn in-context by gradient descent". In: *International Conference on Machine Learning* (2023). arXiv: 2212.07677 [cs.LG].

[Ouy+22] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Neural Information Processing Systems* (2022). arXiv: 2203.02155 [cs.CL].

[Par19] Ju-Won Park. "Queue Waiting Time Prediction for Large-scale High-performance Computing System". In: (2019), pp. 850–855. DOI: 10.1109/HPCS48598.2019.9188119.

[PLW19] Rasmus Berg Palm, Florian Laws, and Ole Winther. "Attend, Copy, Parse End-to-end Information Extraction from Documents". In: *2019 International Conference on Document Analysis and Recognition (ICDAR)* (2019). DOI: 10.1109/icdar.2019.00060.

[Raf+24] Rafael Rafailov et al. "Direct Preference Optimization: Your Language Model is Secretly a Reward Model". In: *Neural Information Processing Systems* (2024). arXiv: 2305.18290 [cs.LG].

[Rah+23] Ibrahim Rahhal, Kathleen M. Carley, Ismail Kassou, and Mounir Ghogho. "Two Stage Job Title Identification System for Online Job Advertisements". In: *IEEE Access* (2023). DOI: 10.1109/access.2023.3247866.

[Ram+23] Ori Ram et al. "In-Context Retrieval-Augmented Language Models". In: *Transactions of the Association for Computational Linguistics* (2023). DOI: 10.1162/tacl_a_00605.

[Ras23a] Sebastian Raschka. "Finetuning LLMs Efficiently with Adapters". In: (2023). URL: https://magazine.sebastianraschka.com/p/finetuning-llms-with-adapters (visited on 09/26/2024).

[Ras23b] Sebastian Raschka. "Practical Tips for Finetuning LLMs Using LoRA (Low-Rank Adaptation)". In: (2023). URL: https://substack.com/@rasbt/p-138081202 (visited on 09/26/2024).

[RG19] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019). DOI: 10.18653/v1/d19-1410.

[RHU01] A. Richei, U. Hauptmanns, and H. Unger. "The human error rate assessment and optimizing system HEROS". In: *Reliability Engineering and System Safety* (2001). DOI: 10.1016/s0951-8320(01)00005-9.

[RK92] Johan Roos and Georg Von Krogh. "Figuring out your competence configuration". In: *European Management Journal* (1992). DOI: 10.1016/0263-2373(92)90006-p.

[Sat+17] Bahar Sateli, Felicitas Löffler, Birgitta König-Ries, and René Witte. "ScholarLens: extracting competences from research publications for the automatic generation of semantic user profiles". In: *PeerJ Computer Science* (2017). DOI: 10.7717/peerj-cs.121.

[SC21] Ramya Srinivasan and Ajay Chander. "Biases in AI Systems". In: *Communications of the ACM* (2021). DOI: 10.1145/3464903.

[SCF22] Tyler J. Skluzacek, Kyle Chard, and Ian Foster. "Automated metadata extraction: challenges and opportunities". In: *2022 IEEE 18th International Conference on e-Science (e-Science)* (2022). DOI: 10.1109/escience55777.2022.00088.

[Sch19] Käthe Schneider. "What Does Competence Mean?" In: *Psychology* (2019). DOI: 10.4236/psych.2019.1014125.

[Sha+17] Yuanlong Shao et al. "Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017). DOI: 10.18653/v1/d17-1235.

[SL22] Uri Shaham and Omer Levy. "What Do You Get When You Cross Beam Search with Nucleus Sampling?" In: *Proceedings of the Third Workshop on Insights from Negative Results in NLP* (2022). DOI: 10.18653/v1/2022.insights-1.5.

[Su+24] Jianlin Su et al. "RoFormer: Enhanced Transformer with Rotary Position Embedding". In: *Neurocomputing* (2024). DOI: 10.1016/j.neucom.2023.127063.

[Tan+24] Raphael Tang, Crystina Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. "Found in the Middle: Permutation Self-Consistency Improves Listwise Ranking in Large Language Models". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (2024). DOI: 10.18653/v1/2024.naacl-long.129.

[Tia+23] Xiaoguang Tian, Robert Pavur, Henry Han, and Lili Zhang. "A machine learning-based human resources recruitment system for business process management: using LSA and BERT and SVM". In: *Business Process Management Journal* (2023). DOI: 10.1108/bpmj-08-2022-0389.

[TWT23] Sameer Yadavrao Thakur, K. H. Walse, and V. M. Thakare. "A Systematic Review on Explicit and Implicit Aspect Based Sentiment Analysis". In: *Lecture Notes in Networks and Systems* (2023). DOI: 10.1007/978-981-19-6631-6_24.

[Vas+23] Ashish Vaswani et al. "Attention is All You Need". In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)* (2023). DOI: 10.48550/arXiv.1706.03762.

[Ver+08] Joris Vertommen, Frizo Janssens, Bart De Moor, and Joost R. Duflou. "Multiple-vector user profiles in support of knowledge sharing". In: *Information Sciences* (2008). DOI: 10.1016/j.ins.2008.05.001.

[VM21] A. Helen Victoria and G. Maragatham. "Automatic tuning of hyperparameters using Bayesian optimization". In: *Evolving Systems* (2021). DOI: 10.1007/s12530-020-09345-2.

[Vuk+21] Davor Vukadin, Adrian Satja Kurdija, Goran Delac, and Marin Silic. "Information Extraction From Free-Form CV Documents in Multiple Languages". In: *IEEE Access* (2021). DOI: 10.1109/access.2021.3087913.

[Wan+24] Peiyi Wang et al. "Large Language Models are not Fair Evaluators". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2024). DOI: 10.18653/v1/2024.acl-long.511.

[Wes01] Wim Westera. "Competences in education: A confusion of tongues". In: *Journal of Curriculum Studies* (2001). DOI: 10.1080/00220270120625.

[WHB14] Sebastian Weirich, Martin Hecht, and Katrin Böhme. "Modeling Item Position Effects Using Generalized Linear Mixed Models". In: *Applied Psychological Measurement* (2014). DOI: 10.1177/0146621614534955.

[Wid+20] Adhika Pramita Widyassari et al. "Review of automatic text summarization techniques and methods". In: *Journal of King Saud University - Computer and Information Sciences* (2020). DOI: 10.1016/j.jksuci.2020.05.006.

[Yan+24] Jingfeng Yang et al. "Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond". In: *ACM Transactions on Knowledge Discovery from Data* (2024). DOI: 10.1145/3649506.

[Ye+23] Xi Ye et al. "Complementary Explanations for Effective In-Context Learning". In: *Findings of the Association for Computational Linguistics: ACL 2023* (2023). DOI: 10.18653/v1/2023.findings-acl.273.

[Zha+22] Mike Zhang, Kristian Jensen, Sif Sonniks, and Barbara Plank. "SkillSpan: Hard and Soft Skill Extraction from English Job Postings". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2022). DOI: 10.18653/v1/2022.naacl-main.366.

[Zhe+23] Lianmin Zheng et al. "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena". In: *Neural Information Processing Systems* (2023). arXiv: 2306.05685 [cs.CL].

A Attribution

ChatGPT ChatGPT [25] served as a tool for rapid prototyping and testing within the development of the competency extraction system, alongside facilitating both prompt generation and iteration as well as example creation. Although the capabilities of ChatGPT proved insufficient for generating substantial components of the final system, it was instrumental in enabling swift ideation and iterations. Initial prototypes were crafted using ChatGPT, subsequently transferred to VS-Code for extensive revisions and enhancements, typically involving a complete overhaul of the prototype with the aid of GitHub Copilot.

Furthermore, ChatGPT assisted in drafting the initial version of this work. This draft was subsequently mostly discarded and extensively revised manually. In the final stages, ChatGPT contributed to refining the tone and style, aligning them with the scholarly standards expected in thesis writing.

Citations Zotero[26] was employed for the aggregation, organization, and citation of research papers. Additionally, Consensus[27] facilitated the identification of significant and highly cited research relevant to the thesis topics throughout the composition phase.

Images Images incorporated within this work were either designed using Canva[28] supplemented by icons from Flaticon[29], generated in LaTeX or directly attributed to their sources in the captions.

B Bias Evaluation Calculations

The subsequent calculations underpin the results delineated in Table table 6.

	Mixtral-8x7B	**Llama3-70B**	**GPT-4o**
Mismatches (M)	12/36 (33.33%)	10/29 (34.48%)	11/36 (30.56%)
Profile 1 Preferred (P_1)	46/72 (63.89%)	39/58 (67.24%)	39/72 (54.17%)
Mismatched & Profile 1 Preferred	11/12 (91.66%)	10/10 (100%)	7/11 (63.64%)
Mismatched & Profile 2 Preferred	1/12 (8.33%)	0/10 (0%)	4/11 (36.36%)

Table 9: Bias Evaluation Results

[25] www.chatgpt.com
[26] www.zotero.org
[27] www.consensus.app
[28] www.canva.com
[29] www.flaticon.com

The absence of outputs for Llama3-70B indicates a lack of production from the LLM for those evaluations, a problem extensively examined in section 6.3.3.

- The estimated error attributable to positional bias, E_{pb}, is calculated as:

$$E_{pb} = (P_1 - 50\%) \times 2$$

  Where P_1 denotes the proportion favoring Profile 1:

  - Mixtral-8x7B: $(63.89\% - 50\%) \times 2 = 27.78\%$
  - Llama3-70B: $(67.24\% - 50\%) \times 2 = 34.48\%$
  - GPT-4o: $(54.17\% - 50\%) \times 2 = 8.34\%$

- The inconsistency I is derived as:

$$I = M - E_{pb}$$

  Where M represents the mismatch rate:

  - Mixtral-8x7B: $33.33\% - 27.78\% = 5.55\%$
  - Llama3-70B: $34.48\% - 34.48\% = 0\%$
  - GPT-4o: $30.56\% - 8.34\% = 22.22\%$

- The consistency C is computed as:

$$C = 100\% - I$$

  - Mixtral-8x7B: $100\% - 5.55\% = 94.45\%$
  - Llama3-70B: $100\% - 0\% = 100\%$
  - GPT-4o: $100\% - 22.22\% = 77.78\%$

- The expected number of mismatches E_m is determined as: $E_m = I \times N$, with N being the total comparisons.

$$E_m = I \times N$$

  - Mixtral-8x7B: $5.55\% \times 36 = 1.998$
  - Llama3-70B: $0\% \times 29 = 0$
  - GPT-4o: $22.22\% \times 36 = 7.9992$

  These findings align closely with two times the observed mismatches where profile 2 was preferred, as expected.

Assertion

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, October 2, 2024 Bertil Braun